

U N I V E R Z A V L J U B L J A N I

Fakulteta za elektrotehniko

Melita Hajdinjak

**PREDSTAVITEV ZNANJA IN
VREDNOTENJE UČINKOVITOSTI
SODELUJOČIH SAMODEJNIH
SISTEMOV ZA DIALOG**

DOKTORSKA DISERTACIJA

Mentor: izr. prof. dr. France Mihelič

Somentor: doc. dr. Andrej Bauer

Ljubljana, 2006

Ključne besede: sistemi za dialog, sistemi za podajanje informacij, eksperiment Čarovnik iz Oza, modul za vodenje dialoga, tehnika polnjenja predalčkov, konverzacijska teorija iger, konverzacijske igre, konverzacijske poteze, rekurzivne mreže prehodov, dialog mIPC Kripkejevi modeli, intuicionistična modalna logika, vrednotenje učinkovitosti sistemov za dialog, ogrodje PARADISE, zadovoljstvo uporabnikov, cene dialoga, uspešnost naloge, funkcija učinkovitosti, multipla linearna regresija, koeficient determinacije, delna F statistika, podatkovne zbirke, podatkovni modeli, relacijske podatkovne zbirke, relacijski podatkovni modeli, relacijska algebra, deduktivne podatkovne zbirke, deduktivni podatkovni modeli, Datalog, ekstenzionalna podatkovna zbirka, intenzionalna podatkovna zbirka, Hornovi stavki, dejstva, pravila, integritetne omejitve, predpostavka zaprtega sveta, sodelujoči podatkovni sistemi, tehnike sodelujočega odgovarjanja, relaksacija poizvedb, gnezdeni relacijski podatkovni modeli, urejeni relacijski podatkovni modeli, objektno-relacijski podatkovni modeli, sodelujoči relacijski podatkovni modeli, teorija kategorij, kategorni podatkovni modeli, regularni podobjekti, zožki, kategorni model relacijske algebre, začetni objekt, končni objekt, kategorni produkt, slika, kategorna projekcija, kategorni presek, kategorna izbira, kategorna unija, kategorna razlika, kategorija podobnosti, kategorija urejenih množic, kategorija kompaktnih metričnih prostorov

*Tistim, ki vedno zrejo v tla,
se lahko zgodi, da prezrejo mavrico.*

Zahvala

Iskrena hvala izjemnemu mentorju prof. dr. Francetu Miheliču, ki je vame verjel in me bodril tudi takrat, ko mi je bilo najtežje. Prepričana sem, da brez njegovega včasih že skoraj brezmejnega razumevanja, neizmerne človeške in strokovne podpore ter pozitivne naravnosti te do sedaj najtežje življenjske preizkušnje ne bi zmogla.

Nepogrešljiv delež k disertaciji pa je prispeval somentor doc. dr. Andrej Bauer, čigar strokovna odličnost, osebnostna odprtost in dostopnost so mi bile v neprecenljivo pomoč in spodbudo pri premagovanju številnih ovir, s katerimi sem se srečevala v obdobju nastanka tega dela.

Hvala obema – mentorju, ki me je seznanil z zelo privlačnim raziskovalnim področjem govornih tehnologij in me usmeril vanj, ter somentorju, ki me je navdušil za (prej neprijetno) teorijo kategorij in me s svojimi idejami vedno znova presenečal in motiviral. Ne nazadnje bi se rada zahvalila tudi ostalim članom komisije, ki so s svojimi tehtnimi in skrbno preišljenimi pripombami pozitivno vplivali na kvaliteto doktorske disertacije.

Hvala sedanjim in nekdanjim sodelavcem Laboratorija za umetno zaznavanje, sisteme in kibernetiko, ki so me sprejeli tako, kot sem – odprto, neposredno, za marsikoga preveč uporniško in trmasto. Izpostaviti želim vodjo laboratorija prof. dr. Nikola Pavešića, ki mi je bil s svojim pozitivnim odnosom do ljudi in življenja vedno za zgled.

Prav (p)osebna hvala gre vsem mojim najdražjim, ki so mi vivali upanja in prepotrebne moči takrat, ko so me okoliščine in življenje zlomili. Hvala staršem, bratu, prijateljem, še posebej pa mojima dragima Borisu in Urški, ki sta mi vselej stala ob strani, tudi takrat, ko sem potrebovala le 'taksista' do bolnice in nazaj. Hvala za izjemen posluh, razumevanje, pomoč in potrpežljivost!

Povzetek

Avtomatizacija sporazumevanja z govorom je eden izmed največjih raziskovalnih izzivov današnjega časa. Računalniške sisteme, prek katerih z govorom dostopamo do določenih aplikacij, imenujemo samodejni sistemi za dialog ali govorni vmesniki. Disertacija se osredotoča na (samodejne) sisteme za podajanje informacij, ki so govorna različica splošnejših podatkovnih sistemov, in je usmerjena v razvoj dvojezičnega sistema za podajanje informacij o vremenu in vremenski napovedi.

Eksperimentalni del razvoja sistema za dialog običajno poteka v treh korakih. Najprej analiziramo dialoge človek–človek, ki potekajo v ciljni domeni sistema. Nato sledi konstrukcija vsaj enega sistema Čarovnik iz Oza, ki služi kot ogrodje za izvedbo istoimenskega eksperimenta, v katerem uporabnike pustimo v prepričanju, da se pogovarjajo z računalnikom, v resnici pa za računalnikom sedi človek čarovnik, ki vsaj delno uravnava delovanje sistema. Podatki, pridobljeni v eksperimentih Čarovnik iz Oza, odsevajo jezik komunikacije človek–stroj in služijo končni zasnovi sistema za dialog. Zadnji korak je vrednotenje, izboljševanje in dopolnjevanje že dograjenega sistema, ki temelji na podatkih, pridobljenih od dejanskih uporabnikov.

Doktorska disertacija se nanaša na drugi in tretji korak eksperimentalnega dela razvoja sistemov za dialog – zajema zbiranje podatkov z eksperimentom Čarovnik iz Oza, načrtovanje sistema za dialog ter vrednotenje učinkovitosti zgrajenih modulov.

Uvodno poglavje predstavi obstoječe sisteme za dialog, njihovo zgradbo in smernice razvoja.

Drugo poglavje obravnava eksperimente Čarovnik iz Oza, na katerih temeljijo raziskave, razvoj in sklepi disertacije. Eksperiment Čarovnik iz Oza smo v okviru razvoja dvojezičnega sistema za podajanje informacij o vremenu uporabili dvakrat. V prvem sistemu je čarovnik simuliral delovanje modulov za razpoznavanje govora, razumevanje naravnega jezika in vodenje dialoga. Na osnovi podatkov, pridobljenih v prvem eksperimentu Čarovnik iz Oza, smo razvili modul za (samodejno) vodenje dialoga in ga vključili v sistem, s katerim smo izvajali drugi eksperiment Čarovnik iz Oza. V drugem sistemu smo relacijsko zbirko vremenskih podatkov zamenjali s sodelujočo podatkovno zbirko, ki je temeljila na dialog mIPC Kripkejevih modelih, tj. podatkovnih modelih, razvitih v sklopu raziskav, predstavljenih v magistrskem delu. Sledi natančnejši opis modula za vodenje dialoga in sodelujoče podatkovne zbirke, torej komponent, glede na kateri sta se oba sistema Čarovnik iz Oza razlikovala. Dialog smo modelirali s tehniko polnjenja predalčkov, ki predstavlja trenutno najobičajnejši način modeliranja dialogov človek–stroj v govornih vmesnikih, strategijo vodenja dialoga pa s konverzijsko teorijo iger. Podatkovni model sodelujoče podatkovne zbirke, uporabljene v drugem eksperimentu Čarovnik iz Oza, je temeljil na intuicionistični modalni logiki.

Tretje poglavje opisuje potencialno splošno metodologijo vrednotenja učinkovitosti sistemov za dialog, imenovano ogrodje PARADISE (PARAdigm for Dialogue System Evaluation). Ogrodje PARADISE omogoča izpeljavo ocene učinkovitosti sistema kot uteženo linearno kombinacijo od domene odvisnih parametrov uspešnosti naloge in cen dialoga, zajema pa model učinkovitosti sistema, katerega osnovni cilj je maksimirati zadovoljstvo uporabnikov. Model učinkovitosti sistema, ki ga zajema ogrodje PARADISE, trdi, da lahko funkcijo učinkovitosti sistema določimo z multiplo linearno regresijo z zadovoljstvom uporabnikov kot odvisno spremenljivko ter parametri uspešnosti naloge in cen dialoga kot neodvisnimi spremenljivkami. Funkcija učinkovitosti tedaj omogoča napovedovanje zadovoljstva uporabnikov, vrednotenje učinkovitosti in izboljševanje sistema, primerjavo sistemov z istimi ali različnimi domenami, samodejno iskanje problematičnih dialogov in spreminjanje strategije vodenja dialoga že med samo interakcijo. Poglavje se osredotoča na nekatere pomanjkljivosti, težave in nerešena vprašanja ogrodja PARADISE – opiše vpliv normalizacije na natančnost napovedovanja zadovoljstva uporabnikov, navede regresijske predpostavke in poudari pomembnost dobre izbire regresijskih parametrov. Nakaže tudi morebitne učinke razpoznavanja govora na rezultate vrednotenja in analizira vprašalnik, na podlagi katerega ogrodje PARADISE določa zadovoljstvo uporabnikov (tj. odvisno spremenljivko funkcije učinkovitosti). V dosedanjih raziskavah so bile nekatere izmed naštetih točk premalo poudarjene, zanemarjene ali celo napačno interpretirane.

Četrto poglavje dokumentira vrednotenje učinkovitosti obeh sistemov Čarovnik iz Oza z ogrodjem PARADISE. Opišemo izbrane parametre modela učinkovitosti, ki jih določimo samodejno, če je mogoče, ali jih ročno označimo. Vpeljemo t. i. parametre podatkovne zbirke, ki se v literaturi o vrednotenju učinkovitosti sistemov za dialog ne pojavljajo. Odvisno spremenljivko modela učinkovitosti, torej zadovoljstvo uporabnikov, pridobimo z vprašalnikom, ki ga predlaga ogrodje PARADISE in v katerem uporabniki podajo stopnjo strinjanja z izjavami o obnašanju oz. učinkovitosti sistema za dialog. Na osnovi analize korelacij med izbranimi parametri in korelacij parametrov z odvisno spremenljivko modela učinkovitosti določimo množico vhodnih regresijskih parametrov. Za statistično obdelavo podatkov ter manipulacijo z izbranimi regresijskimi parametri uporabimo lastna programska orodja. Sledi izpeljava funkcij učinkovitosti obeh sistemov Čarovnik iz Oza za odvisno spremenljivko, ki jo dobimo kot vsoto ocen, pridobljenih z vprašalnikom o zadovoljstvu uporabnikov. Vse funkcije učinkovitosti pridobivamo s postopkom vzvratne eliminacije. Zaradi velike razlike v natančnosti dobljenih modelov sklepamo, da te odvisne spremenljivke, ki jo predlaga ogrodje PARADISE in ki naj bi izražala zadovoljstvo uporabnikov, ni mogoče dobro modelirati. Ker sta se oba sistema Čarovnik iz Oza razlikovala le v načinu vodenja dialoga in predstavitvi znanja, argumentiramo, da je za odvisno spremenljivko bolj smiselno vzeti seštevek tistih ocen iz vprašalnika, ki merijo te razlike. Sledi izpeljava in natančna analiza funkcij učinkovitosti obeh sistemov Čarovnik iz Oza za tako dobljeno odvisno spremenljivko. Razliko v natančnosti obeh funkcij učinkovitosti tako dejansko zmanjšamo. Funkciji učinkovitosti nas vodita do spoznanja, da je predstavitev znanja oz. zgradba podatkovne zbirke sistema za dialog izjemnega pomena in da so parametri podatkovne zbirke pri vrednotenju sistemov za podajanje informacij nepogrešljivi. Na koncu preverimo še, kako dobro se dajo modelirati posamezne ocene uporabnikov in kateri parametri na njih najmočneje vplivajo. Ugotovimo, da je parameter podatkovne zbirke, ki je v prvem

eksperimentu statistično najznačilnejši parameter za sedem izmed osmih ocen uporabnikov, statistično neznačilen za skoraj vse ocene drugega eksperimenta. Izpostavimo potrebo po nadaljnjih raziskavah, ki bi razvile veljaven psihometrični vprašalnik o zadovoljstvu uporabnikov sistemov za dialog in bile sposobne razumeti in pravilno opredeliti razlike med funkcijami učinkovitosti.

V petem poglavju se ponovno posvetimo podatkovnim zbirkam. Predstavimo zelo priljubljene relacijske podatkovne zbirke, ki so se pri običajni administrativni obdelavi podatkov izkazale za zelo uspešne. Operacije, ki jih izvajamo nad relacijami v relacijskem podatkovnem modelu, opišemo na algebrajski način, tj. s proceduralno relacijsko algebro. Algebrajskemu načinu vzporeden je sicer logični način, ki porodi relacijski račun. V relacijski algebri ponavadi definiramo pet osnovnih operacij – unijo, razliko, kartezični produkt, projekcijo in izbiro. Obstaja še vrsta drugih zanimivih operacij nad relacijami, ki jih lahko izrazimo z osnovnimi operacijami relacijske algebre. To so npr. presek, kvocient, spoj, naravni spoj in semispoj. Naštejemo nekatere algebrajske zakone, ki se jim podrejajo operacije relacijske algebre. Ti algebrajski zakoni igrajo pomembno vlogo pri algebrajski optimizaciji oz. algebrajski poenostavitvi. Vedno bolj kompleksne aplikacije pa vodijo do spoznanja o nekaterih pomembnih pomanjkljivostih relacijskih podatkovnih zbirk in na njih temelječih podatkovnih sistemov. Pomanjkljivosti se nanašajo predvsem na nesposobnost sodelujočega odgovarjanja, nesposobnost izražanja tranzitivnega zaprtja in komplementa ter nesposobnost ravnanja s kompleksnimi tipi podatkov, kot so npr. slike, zaporedja, tabele, množice, vreče.

Šesto poglavje opiše najbolj znan model deduktivnih podatkovnih zbirk, imenovan Datalog, ki je nastal kot odgovor na pomanjkljivosti relacijskih podatkovnih zbirk. Datalog temelji na logiki prvega reda in v nasprotju z relacijsko algebro in relacijskim računom zajema tudi rekurzivne izraze. V deduktivnih podatkovnih modelih opisujemo podatke in relacije med podatki z jezikom prvega reda, upravljanje in manipuliranje s podatki pa omogoča vrednotenje logičnih formul. V Datalogu nastopata dve vrsti predikatov – predikati ekstenzionalne podatkovne zbirke (EPZ) ustrezajo relacijam, shranjenim v podatkovni zbirki, predikati intenzionalne podatkovne zbirke (IPZ) pa ustrezajo relacijam, definiranim z logičnimi pravili. V relacijskem podatkovnem modelu so vse relacije tipa EPZ. V Datalogu so formule in pravila sklepanja zapisani v obliki Hornovih stavkov, ki jih delimo na dejstva, integritetne omejitve in pravila. Dejstva določajo EPZ relacije, pravila IPZ relacije, integritetne omejitve, ki so v deduktivnih podatkovnih zbirkah zapisane v istem logičnem jeziku kot dejstva in pravila, pa služijo za preprečevanje vnosa neustreznih podatkov, preverjanje konsistentnosti podatkovne zbirke in veljajo za temelj sodelujočih podatkovnih sistemov. Logična pravila interpretiramo na tri različne načine – z dokazi, modeli in algoritmi. Posebej interpretiramo nerekurzivna Datalog pravila, rekurzivna Datalog pravila in Datalog pravila z negacijo. Prednosti logike prvega reda kot teoretične osnove podatkovnih sistemov so razumljiva semantika in enoten jezik za izražanje dejstev, pravil, poizvedb in integritetnih omejitev. Datalogu, ki je zasnovan kot deklarativen podatkovni jezik, a ne doseže čiste deklarativne oblike, očitajo tudi nekatere slabosti. Me drugim so to obsežna optimizacija in neuspešno izkoriščanje celotne izrazne moči.

Sedmo poglavje primerja relacijsko algebro in Datalog kot najbolj znana modela relacij-

skih in deduktivnih podatkovnih zbirk. Ker predstavljajo integritetne omejitve izjemno prednost deduktivnih podatkovnih modelov pred relacijskimi podatkovnimi modeli, je zanimiva edino primerjava relacijske algebre in Dataloga brez integritetnih omejitev. V tem okviru velja, da Datalog brez negacije in relacijska algebra po izrazni moči nista primerljiva. Obstajajo funkcije, izrazljive v relacijski algebri, ki niso izrazljive v Datalogu, in obratno. Če dovolimo negacijo, je Datalog po izrazljivosti močnejši od relacijske algebre. Množica funkcij, izrazljivih v relacijski algebri, pa je ekvivalentna množici funkcij, izrazljivih z nerekurzivnim Datalog programom z negacijo, katerega pravila so tudi varna in plastovita.

Osmo poglavje obravnava sodelujoče odgovarjanje v okviru podatkovnih sistemov. Sodelujoči podatkovni sistemi so informacijski sistemi, ki se na poizvedbo ne odzivajo le z odgovorom, ki poizvedbi strogo zadošča, temveč tudi z dodatnimi informacijami, katerih namen je odpraviti nejasnosti, razčistiti dvoumnosti in pozitivno vplivati na doseganje uporabnikovih ciljev in namer. Uveljavile so se različne tehnike sodelujočega odgovarjanja, ki jih delimo na tehnike upoštevanja uporabnikovih prepričanj, želja in namer, na tehnike vrednotenja domnev, vsebovanih v poizvedbah, na tehnike odkrivanja in odpravljanja nesporazumov, na tehnike oblikovanja intenzionalnih odgovorov ter na tehnike posploševanja poizvedb in odgovorov. Tri izmed omenjenih tehnik temeljijo na integritetnih omejitvah, zadnja, ki je s stališča ponujanja podatkov in usmerjanja uporabnika k izbiri dosegljivih, relevantnih podatkov najpomembnejša, pa temelji na posploševanju oz. razširjanju obsega poizvedb. Tako transformacijo poizvedb imenujemo relaksacija, zasnovana pa je na posebni urejenosti predikatov in konstant podatkovne zbirke. Izjave logike prvega reda, ki opisujejo urejeno množico predikatov in konstant podatkovne zbirke, imenujemo klasifikacijske izjave. V poglavju opišemo postopek relaksacije in preverimo, kdaj je odgovor na začetno poizvedbo podmnožica odgovora na relaksirano poizvedbo. Ker relaksacija raziskuje soseščino začetne poizvedbe, so najzanimivejše tiste relaksirane poizvedbe, ki so dobljene v najmanj korakih. Bolj se od začetne poizvedbe oddaljimo, manj verjetno je, da bo odgovor uporabnika zanimal. Sledi pregled tehnik sodelujočega odgovarjanja, ki jih podpirajo relacijska algebra, Datalog in dialog mIPC Kripkejevi modeli. Relacijska algebra in ostali modeli relacijskih podatkovnih zbirk ne podpirajo nobene izmed tehnik sodelujočega odgovarjanja. To pa pomeni, da jih v njihovi osnovni obliki ne moremo šteti za sodelujoče podatkovne modele. V nadaljevanju se zato posvetimo različnim razširitvam relacijskega podatkovnega modela, tudi razširitvam, ki v vrednostne množice atributov vpeljejo urejenost, in razširitvam, ki podpirajo nekatere tehnike sodelujočega odgovarjanja. Datalog sicer podpira vse tri tehnike sodelujočega odgovarjanja, ki temeljijo na integritetnih omejitvah, dialog mIPC Kripkejevi modeli pa poleg tehnik, ki jih podpira Datalog, podpirajo še tehniko posploševanja poizvedb in odgovorov oz. relaksacijo.

Deveto poglavje želi sodelujoče dialog mIPC Kripkejeve podatkovne modele približati bolj univerzalno zastavljenemu relacijskemu podatkovnemu modelu, kar je eden od glavnih ciljev doktorske disertacije. Formalizem, ki ga v ta namen uporabimo, je teorija kategorij. Eden od čarov teorije kategorij je njena sposobnost združevanja shematičnega formalizma in algebraičnega simbolizma. V povezavi s podatkovnimi zbirkami sicer ločimo dva popolnoma različna načina uporabe teorije kategorij – modularizacijo in sintetično sklepanje. Poiščemo kategorni model relacijskih podatkovnih zbirk, ki temelji na modularizaciji. Relacijsko algebro kot najbolj znan model relacijskih

podatkovnih zbirk (matematično) naravno posplošimo na teorijo kategorij. To je, relacijam in nad njimi definiranim operacijam relacijske algebre (unija, razlika, kartezični produkt, projekcija in izbira) priredimo ustrezne kategorne konstrukte. Ta ideja se porodi, ker je relacijska algebra definirana v kategoriji množic, tj. najnaravnejšem primeru kategorije. Ugotovimo, da morajo kategorije, ki bodo primerna osnova kategornega modela relacijskih podatkovnih zbirk, vsebovati natančen začetni objekt, ki je posplošitev pojma prazne podmnožice, končni objekt, ki je posplošitev pojma podmnožice z enim samim elementom, končne kategorne produkte, slike ter kategorne unije, preseke in razlike tistih podobjektov, s katerimi modeliramo poizvedbe. Predstavimo primer tovrstne kategorije, ki jo imenujemo kategorija podobnosti, in njeni podkategoriji, imenovani kategorija urejenih množic in kategorija kompaktnih metričnih prostorov. Podkategorija urejenih množic se odlikuje po (refleksivni) urejenosti vrednostnih množic, ki omogoča izvajanje relaksacije. Podkategorija kompaktnih metričnih prostorov v množice vpelje metriko, ki pride do izraza, ko želimo opisati razdalje med kraji ali pa želimo kakšno vrednostno množico urediti natančneje, kot omogočajo relacije. Sledi opredelitev zmogljivosti dobljenega kategornega modela in primerjava z dialog mIPC Kripkejevimi modeli. S posplošitvijo relacijske algebre dobimo močan formalizem, ki že v osnovi dovoljuje relaksacijo poizvedb, in izkaže se, da lahko tudi kategorno posplošene relacijske podatkovne zbirke udejanjimo kot zbirko relacijskih tabel. Podobnost na vrednostnih množicah, ki jo podamo z dodatnimi relacijskimi tabelami, je integrirana tako, da je do nje mogoče brez težav dostopati in jo po potrebi spreminjati ali prilagajati uporabniku.

Sklepno poglavje povzame rezultate, dosežene v doktorski disertaciji, opredeli njihov pomen, predlaga nekatere smernice za nadaljnje raziskovalno delo ter predstavi možnosti izboljšave razvitih podatkovnih modelov.

Abstract

Attempts to automate spoken language interaction raise significant research challenges at the forefront of human knowledge. Computer systems that enable users to interact with a certain application via spoken language are called dialogue systems or speech interfaces. This thesis focuses on (automatic) information-providing dialogue systems, which are a special case of more general information systems, and is directed to the development of a bilingual, weather-information-providing, natural-language-spoken dialogue system.

The experimental part of the dialogue-system development process usually consists of three steps. In the first step, human-human dialogues, collected in the goal domain, are analysed. In the second step, at least one Wizard-of-Oz system, which is used to conduct Wizard-of-Oz experiments, is constructed. In these experiments, the subjects are told to interact with a computer system, though in fact they are not. Instead, the interaction is mediated by a human operator, the wizard. The human-computer dialogues, collected with Wizard-of-Oz experiments, reflect the language that would be attempted when communicating with a computer and are used to design the dialogue system of interest. In the last step, the already constructed dialogue system is evaluated, improved, and supplemented.

This thesis addresses the second and the third step of the experimental part of the dialogue-system development process – it describes the data collection process, it presents the planning of a dialogue system, and it reports about the performance evaluation of already constructed modules.

The first chapter introduces the research field in question, including existing dialogue systems and research challenges.

The second chapter describes two Wizard-of-Oz experiments, used as the research and development basis of the thesis. The aim of these experiments was to evaluate the performance of the dialogue-manager component. Therefore, while the task of the wizard in the first experiment was to perform speech understanding and dialogue management, her task in the second experiment was to perform only speech understanding, and the dialogue-management task was assigned to the newly-implemented dialogue manager. The system's weather-information database was based on the relational data model in the first Wizard-of-Oz experiment, and it was based on the cooperative dialogue mIPC Kripke models in the second Wizard-of-Oz experiment. The dialogue-manager component and the cooperative database are described in detail. The dialogue is modeled using the slot-filling approach, which is the most widely used approach for modeling human-computer dialogues. The dialogue strategy is based on conversational game theory and is modeled using recursive transition networks. The dialogue mIPC Kripke

model of the cooperative weather-information database, which was used in the second Wizard-of-Oz experiment and developed within the master's thesis research work, is based on intuitionistic modal logic.

The third chapter deals with the PARADISE (PARAdigm for Dialogue System Evaluation) framework, a potential general methodology for evaluating spoken-language dialogue systems. The PARADISE framework maintains that the system's primary objective is to maximize user satisfaction, and it derives a combined performance metric for a dialogue system as a weighted linear combination of task-success measures and dialogue costs. The PARADISE model of performance posits that a performance function can be derived by applying multivariate linear regression with user satisfaction as the dependent variable and task-success measures and dialogue costs as the independent variables. When the performance function has been derived, it can be used to predict user satisfaction, to evaluate the performance of the dialogue system and to optimise it, to compare different dialogue systems, to automatically identify problematic dialogues, and to switch between different dialogue strategies during the interaction. The chapter focuses on some PARADISE issues (with most of them arising from the application of multivariate linear regression) that have, up to now, not been sufficiently emphasized or have even been neglected by the dialogue-system community. These include considerations regarding the selection of appropriate regression parameters, normalization effects on the accuracy of the prediction, the influence of speech-recognition errors on the performance function, and the selection of an appropriate user-satisfaction measure.

The fourth chapter gives the results of an evaluation of data from two Wizard-of-Oz experiments. It lists all the selected parameters of the model of performance, which must be either automatically logged by the system or be hand-labeled. The database parameters expressing the database size and the database structure are introduced. They have not so far been reported in the literature as costs for user satisfaction. User satisfaction is calculated with a survey, introduced within the PARADISE evaluation framework, that asks users to specify the degree to which they agree with several statements about the performance of the system. A closely examination of the correlations between the selected parameters and the correlations of the selected parameters with the dependent variable of the model of performance leads to a subset of parameters appropriate as the regression input. Statistical data analysis and data manipulation is performed by own software tools, and the performance functions are derived using backward elimination. As the target to be predicted we first take the sum of all the user-satisfaction values, suggested by the PARADISE framework, and afterwards the sum of those user-satisfaction values that (in our opinion) measure the dialogue manager's performance. Great differences in the accuracy of predictions for both Wizard-of-Oz systems lead to the conclusion that the dependent variable, suggested by the PARADISE framework, cannot be well modeled. Therefore, we focus on the modified dependent variable measuring the dialogue manager's performance, and come to the conclusion that the knowledge representation and the structure of the dialogue system's database are of great importance. We argue that the database parameters are indispensable when evaluating the performance of information-providing dialogue systems. In addition, we examine which parameters are significant for individual user-satisfaction measures. Almost all the parameters that were most significant to

an individual user-satisfaction measure in the first experiment were insignificant to the same measure in the second experiment and vice versa. Inaccurate regression models of some individual user-satisfaction measures and the observed differences between pairs of performance equations predicting the same dependent variable require further empirical research. Not only does a reliable user-satisfaction measure that would capture the performance-measures of different dialogue-system components need to be established, but the reasons for the possible differences between several performance equations also need to be understood and properly assessed.

The fifth chapter is again dedicated to databases. It discusses the very popular relational data model, which has been quite successful in developing the database technology required for many traditional business database applications. Operations on relations in the relational data model can be expressed by the procedural relational algebra or by the declarative relational calculus. We shall consider relational algebra only. There are five basic operations that serve to define relational algebra – union, set difference, Cartesian product, projection, and selection. There are some other useful operations on relations that can be expressed in terms of the five basic operations, e.g., intersection, quotient, join, natural join, and semijoin. Like many other algebra, there are laws that the operators of relational algebra obey. These are of great importance for algebraic simplification of queries, which is intended to improve the cost of answering the query independent of the actual data or the physical structure of the data. When more complex database applications must be designed and implemented, the relational data model shows certain shortcomings. These newer applications have requirements and characteristics that differ from those of traditional business applications, such as cooperative answering, implementation of transitive closure, more complex structures for objects, and new data types for storing images, sequences, tables, sets, bags, etc.

The sixth chapter studies the Datalog data model, a well-known data model of deductive databases that is based on first-order logic. Datalog, which was designed to overcome certain shortcomings of the relational data model, in contrast to relational algebra and relational calculus deals with recursive expressions. In Datalog, as in other logic-based languages, a program is built from basic first-order objects, called atomic formulas. It also provides capabilities for defining deduction rules for inferencing new information from the stored database facts. There are two ways relations can be defined – a predicate whose relation is stored in the database is called an extensional database (EDB) relation, while one defined by logical rules is called an intensional database (IDB) relation. In the relational model, all relations are EDB relations. A Datalog database is a particular instance of a more general Horn logic program that permits function symbols. A Horn clause is either a fact, an integrity constraint, or a rule. Facts define EDB relations, rules define IDB relations, and integrity constraints provide semantic information about the data in the database. The purpose of integrity constraints is to prevent the entering of incorrect data into the database and to check the integrity of the database. They are seen as the basis for cooperative database systems. There are three alternative ways to define the meaning of rules – the proof-theoretic interpretation, the model-theoretic interpretation, and the computational interpretation. We study nonrecursive Datalog rules, recursive Datalog rules, and Datalog rules with negated subgoals. The main attraction of first-order-logic database models is a uniform representation for facts, rules, queries, and integrity constraints. Unfortunately, Data-

log, which was designed as a declarative database language, does not compromise its clean declarative style. There are also other weaknesses of Datalog, such as extensive optimization and unsuccessful implementation of its full expressive power.

The seventh chapter relates the expressive power of Datalog to the power of relational algebra. As the integrity constraints represent an exceptional advantage of the deductive database models against the relational database models, the only interesting comparison is the comparison of the expressive power of relational algebra to the power of Datalog without integrity constraints. The answer is that without negation in rules, relational algebra and Datalog are incommensurate in their expressive power – there are things each can express that the other cannot. With negation, Datalog is strictly more expressive than relational algebra. In fact, the set of functions expressible in relational algebra is equivalent to the set of functions expressible in Datalog with negation if rules are restricted to be safe, nonrecursive, and have only stratified negation.

The eighth chapter reviews cooperative answering within database systems. The objective in a cooperative database system is to provide information to a user as to why a particular query succeeded or failed. A cooperative answer with extra or alternative information may be more useful and less misleading to a user. The cooperative answering techniques can be separated into five categories: consideration of users' beliefs and expectations, evaluation of query presuppositions, identification and rectification of misconceptions, formulation of intensional answers, and generalization of queries and answers. Three categories of cooperative-answering techniques are based on integrity constraints. The last listed technique, which is based on the cooperative method of abstraction and refinement of queries, also called relaxation, is the most important when considering information-providing dialogue systems with the objective to direct the user to select relevant, available data. Relaxation relies on the database having explicit hierarchies of the relations and of the terms in the domain. First-order clauses designating a conceptual ordering of database predicates and database constants are called taxonomy clauses. The chapter presents the relaxation process, and it gives the condition that must be fulfilled if the answer to the relaxed query should encompass the answer to the original query. Since relaxation explores the neighbourhood of the original query, we will want to look at the first-level relaxations first, then the second-level, and so forth. The further we get from the original query, the less likely it is that the new query will return answers of interest to the user. In addition, the cooperative-answering techniques supported by relation algebra, Datalog, and dialogue mIPC Kripke models are listed. Relational algebra and all the other relational data models do not enable any of the mentioned cooperative techniques, which means that they are not cooperative data models. We briefly discuss several types of extended relational data models, including a partially ordered relational data model and a cooperative relational data model. In contrast, Datalog enables the implementation of all three cooperative techniques that rely on integrity constraints. Dialogue mIPC Kripke models enable the implementation of one additional technique, i.e., generalization of queries and answers.

The ninth chapter aims to bring the highly cooperative dialogue mIPC Kripke models closer to the more generally defined relational data model, which is one of the main goals of the thesis. The formalism used is category theory. One of the attractions

of category theory is the ability to combine diagrammatic formalism and algebraic symbolism. There are two completely different ways of how to apply category theory to databases – modularization and synthetic reasoning. Our search for a categorical model of relational databases is based on modularization. We generalize relational algebra as the most widely known model of relational databases to category theory, i.e., we search for the categorical counterparts of relations and basic relational operations (union, set difference, Cartesian product, projection, and selection). This seems a natural thing to do since relational algebra is defined in the category of sets, which is a most common category. We discover that the categories that will offer a suitable environment for the categorical data model of relational databases need to have a strict initial object being the categorical counterpart to the empty set, a terminal object being the categorical counterpart to the one-element set, finite categorical products, and images. In addition, they need to have categorical unions, intersections, and differences of those objects that are used to model queries. We present an example category, called the category of similarities, which meets all the established conditions, and two of its subcategories, called the category of ordered sets and the category of compact metric spaces. The subcategory of ordered sets is distinguished for its (reflexive) order of domains, which enables the implementation of relaxation. The subcategory of compact metric spaces is distinguished for its objects, i.e., metric spaces, which can be used to measure distances between geographic objects or serve as an organization method of domains. The generalization of relational algebra results in a strong formalism enabling relaxation. It is demonstrated that the categorical models of relational databases can still be implemented as a collection of relational tables. The ordering of sets, which is given by additional relational tables, can be easily accessed, changed, or adapted to users' needs and requirements. Finally, the expressive power of such categorical models is related to the power of dialogue mIPC Kripke models.

In conclusion, we summarize and discuss the results achieved within this thesis. Some directions for future research and improvement of proposed data models are given.

Kazalo

Zahvala	v
Povzetek	vii
Abstract	xiii
1 Uvod	1
2 Eksperiment Čarovnik iz Oza	5
2.1 Uvod	6
2.2 Prvi eksperiment Čarovnik iz Oza	7
2.3 Drugi eksperiment Čarovnik iz Oza	11
2.4 Modul za vodenje dialoga	14
2.5 Sodelujoča podatkovna zbirka	22
3 Ogradje PARADISE	33
3.1 Uvod	34
3.2 Predstavitev ogradja PARADISE	35
3.3 Analiza ogradja PARADISE	38
3.3.1 Vpliv normalizacije na natančnost napovedovanja zadovoljstva uporabnikov	39
3.3.2 Regresijske predpostavke	42
3.3.3 Pomembnost izbire regresijskih parametrov	42
3.3.4 Merjenje zadovoljstva uporabnikov	44
3.3.5 Vplivi razpoznavanja govora na rezultate vrednotenja	45

4	Vrednotenje učinkovitosti	47
4.1	Uvod	48
4.2	Izbira regresijskih parametrov	48
4.3	Korelacije med regresijskimi parametri	53
4.4	Korelacije z odvisno spremenljivko	56
4.5	Funkcije učinkovitosti	58
4.5.1	Zadovoljstvo uporabnikov kot vsota ocen	59
4.5.2	Zadovoljstvo uporabnikov kot vsota izbranih ocen	61
4.5.3	Zadovoljstvo uporabnikov kot vrednost posameznih ocen	67
5	Relacijski podatkovni model	73
5.1	Uvod	74
5.2	Relacijska algebra	75
5.2.1	Osnovne operacije relacijske algebre	76
5.2.2	Izpeljane operacije relacijske algebre	78
5.2.3	Algebrajski zakoni	81
6	Deduktivni podatkovni model	85
6.1	Uvod	86
6.2	Sintaksa Dataloga	87
6.3	Semantika Dataloga	92
6.3.1	Interpretacija nerekurzivnih Datalog pravil	93
6.3.2	Interpretacija rekurzivnih Datalog pravil	100
6.3.3	Interpretacija Datalog pravil z negacijo	105
6.4	Predpostavka zaprtega sveta	113
6.5	Prednosti in slabosti Dataloga	114
7	Relacijska algebra in Datalog	117
7.1	Uvod	118
7.2	Od relacijske algebre do logike	118
7.3	Od logike do relacijske algebre	121

7.4	Primerjava Dataloga in relacijske algebre	122
8	Sodelujoči podatkovni sistemi	125
8.1	Uvod	126
8.2	Pomen integritetnih omejitev	127
8.3	Sodelujoče odgovarjanje	129
8.3.1	Upoštevanje uporabnikovih prepričanj, želja in namer	129
8.3.2	Vrednotenje domnev, vsebovanih v poizvedbah	130
8.3.3	Odkrivanje in odpravljanje nesporazumov	131
8.3.4	Intenzionalni odgovori	131
8.3.5	Posploševanje poizvedb in odgovorov	133
8.4	Tehnike sodelujočega odgovarjanja, ki jih podpirajo Datalog in dialog mIPC Kripkejevi modeli	141
8.5	Razširitve relacijskega podatkovnega modela	142
8.5.1	Gnezdeni relacijski podatkovni modeli	142
8.5.2	Urejeni relacijski podatkovni modeli	143
8.5.3	Objektno-relacijski podatkovni modeli	145
8.5.4	Sodelujoči relacijski podatkovni modeli	145
9	Kategorni model relacijske algebre	149
9.1	Uvod v teorijo kategorij	150
9.2	Kategorna posplošitev relacijske algebre	154
9.2.1	Posplošitev kartezičnega produkta	158
9.2.2	Posplošitev projekcije	160
9.2.3	Posplošitev izbire, unije in razlike	163
9.2.4	Sklep	166
9.3	Kategorija podobnosti	167
9.3.1	Kategorija urejenih množic	174
9.3.2	Kategorija kompaktnih metričnih prostorov	175
9.4	Zmogljivost kategornega modela relacijskih podatkovnih zbirk	178
9.4.1	Implementacija kategornih relacijskih podatkovnih zbirk	178

9.4.2	Relaksacija	185
9.4.3	Primerjava kategornega modela relacijskih podatkovnih zbirk in dialog mIPC Kripkejevih modelov	189
10	Sklep	193
10.1	Pregled in pomen doseženih rezultatov	194
10.1.1	Vrednotenje učinkovitosti	194
10.1.2	Predstavitev znanja	196
10.2	Nadaljnje delo	198
	Viri in literatura	201
	Izvirni prispevki k znanosti	217
	Izjava o avtorstvu	219

Slike

1.1	Zgradba sistema za dialog.	2
2.1	Konverzacijske igre, udejanjene v modulu za vodenje dialoga v sistemu za podajanje informacij o vremenu.	18
2.2	Hassejev diagram relacije <i>biti manj specifičen</i> med krajevnimi delčki informacij, ki se navezujejo na Slovenijo.	26
2.3	Hassejev diagram relacije <i>biti manj specifičen</i> med krajevnimi delčki informacij, ki se navezujejo na ostalo Evropo.	26
2.4	Hassejev diagram relacije <i>biti manj specifičen</i> med časovnimi delčki informacij. Zaradi majhnega števila možnih dosegljivih časovnih trenutkov so vzpostavljene vse povezave, zaradi česar relacija <i>biti manj specifičen</i> v tem primeru izgubi pomen, ki izvira iz imena.	27
2.5	Hassejev diagram relacije <i>biti manj specifičen</i> med podatkovnimi delčki informacij.	27
3.1	Model učinkovitosti sistema za dialog, ki ga zajema ogrodje PARADISE.	35
5.1	Relacija INFO. Podatki o številu prebivalcev so povzeti po rezultatih popisa prebivalstva, ki ga je leta 2002 opravil Statistični urad Republike Slovenije.	75
6.1	Graf odvisnosti za Datalog program iz primera 6.3.	90
8.1	Primer relacijske sheme relacije ODDELEK gnezdene relacijske podatkovne zbirke.	143
8.2	Drevesna predstavitev relacijske sheme relacije ODDELEK.	143
8.3	Primer TAH drevesa letališč v Tuniziji.	146
9.1	Zožek ali regularni monomorfizem.	153
9.2	Relacija \leq v razredu $Mono(A)$	155

9.3	Natančen začetni objekt.	157
9.4	Končni objekt.	157
9.5	Kategorni produkt objektov X_1 in X_2	159
9.6	Kategorni produkt podobjektov $A' \subseteq A$ in $B' \subseteq B$	160
9.7	Faktorizacija morfizma f	161
9.8	Kategorna projekcija podobjekta $C \subseteq A \times B$	162
9.9	Kategorni produkt v kategoriji podobnosti.	171
9.10	Kategorna projekcija v kategoriji podobnosti.	173

Tabele

2.1	Povprečne ocene uporabnikov v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.	9
2.2	Primer dialoga med uporabnikom in prvim sistemom Čarovnik iz Oza. .	10
2.3	Zgradba SSSQ zbirke spontanega govora.	11
2.4	Primer dialoga med uporabnikom (U) in drugim sistemom Čarovnik iz Oza (S).	13
2.5	Zgradba SSSQ2 zbirke spontanega govora.	14
3.1	Vprašalnik za ocenjevanje zadovoljstva uporabnikov, ki ga predlaga ogródje PARADISE.	36
3.2	Primer kontingenčne tabele.	37
4.1	Srednje vrednosti izbranih regresijskih parametrov v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.	52
4.2	Korelacijski koeficienti parametrov v prvem eksperimentu Čarovnik iz Oza.	54
4.3	Korelacijski koeficienti parametrov v drugem eksperimentu Čarovnik iz Oza.	55
4.4	Korelacijski koeficienti in p -vrednosti za regresijske parametre glede na US v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza. .	56
4.5	Tabela vzvratne eliminacije za prvi sistem Čarovnik iz Oza in odvisno spremenljivko US.	59
4.6	Tabela vzvratne eliminacije za drugi sistem Čarovnik iz Oza in odvisno spremenljivko US.	60
4.7	Korelacijski koeficienti in p -vrednosti za regresijske parametre glede na DM v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza. .	63
4.8	Tabela vzvratne eliminacije za prvi sistem Čarovnik iz Oza in odvisno spremenljivko DM.	64

4.9	Tabela vzratne eliminacije za drugi sistem Čarovnik iz Oza in odvisno spremenljivko DM.	65
4.10	Korelacijski koeficienti in pripadajoče p -vrednosti za regresijske parametre glede na posamezne ocene uporabnikov v prvem eksperimentu Čarovnik iz Oza.	68
4.11	Korelacijski koeficienti in pripadajoče p -vrednosti za regresijske parametre glede na posamezne ocene uporabnikov v drugem eksperimentu Čarovnik iz Oza.	69
6.1	Prikaz delovanja algoritma 3 na Datalog programu iz primera 6.3. Pri tem smo uporabili nekatere okrajšave: imena relacij SESTRA_BRAT, SESTRICNA_BRATRANEC in SORODEN smo zamenjali z $S1$, $S2$ in $S3$, urejene pare oblike (f, h) smo zapisali kot fh , zaradi simetričnosti vseh IPZ relacij pa smo prikazali le tiste elemente, katerih prva komponenta je po abecednem redu pred drugo komponento.	104

1 Uvod

Avtomatizacija sporazumevanja z govorom je še vedno eden izmed največjih raziskovalnih izzivov. Razlogov je več:

- ↪ Govor je naraven – govoriti se naučimo, še preden znamo brati in pisati.
- ↪ Govor je učinkovit – večina ljudi je sposobnih govoriti petkrat hitreje kot tipkati in verjetno celo desetkrat hitreje kot pisati.
- ↪ Govor je fleksibilen – med sporazumevanjem z govorom se nam ni treba ničesar dotikati in ne opazovati.

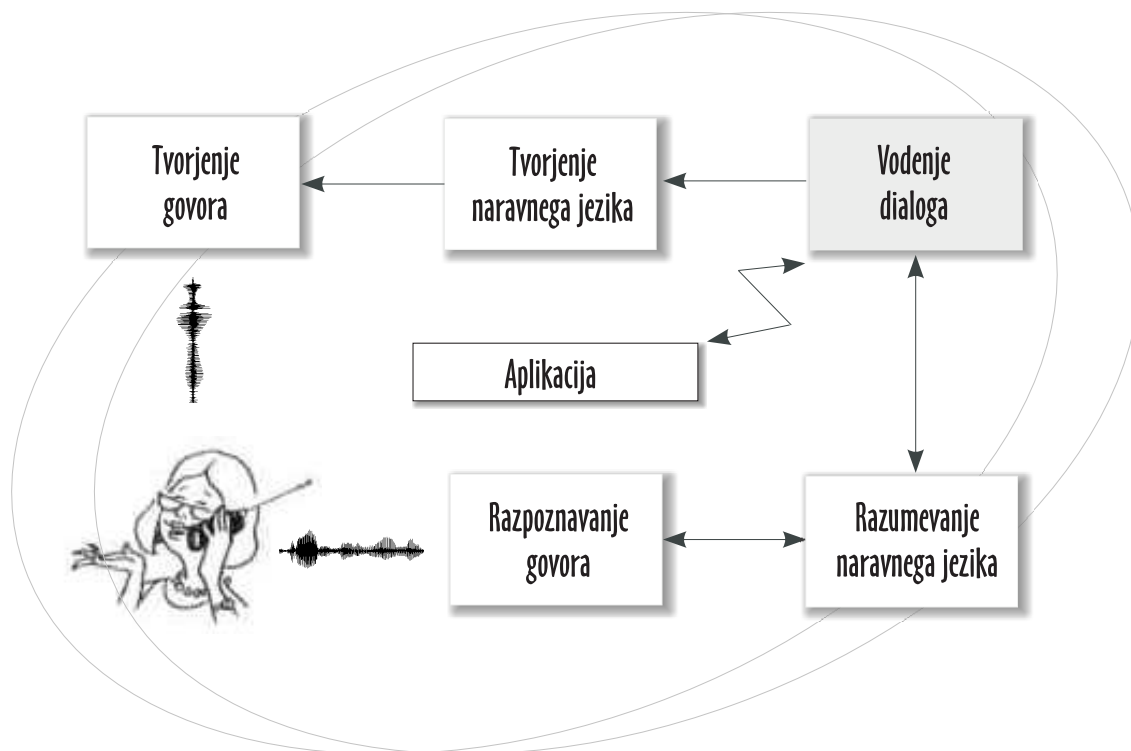
Računalniški sistem, ki uporabniku omogoča, da z govorom dostopa do določenih aplikacij, imenujemo (*samodejni*) *sistem za dialog* ali *govorni vmesnik*. Ločimo več vrst sistemov za dialog [Krahmer-2001], danes najbolj razširjeni pa so *sistemi za podajanje informacij*, *sistemi za govorno upravljanje naprav*, *sistemi za interaktivne govorne odzive* in *sistemi za reševanje problemov*. Prototipi sistemov za dialog so se pojavili že v poznih šestdesetih in zgodnjih sedemdesetih letih prejšnjega stoletja [Woods-1972, Codd-1974].

V disertaciji se bomo osredotočili na sisteme za podajanje informacij, tj. govorne vmesnike za dostop do podatkovnih zbirk, ki so govorna različica splošnejših *podatkovnih sistemov* [Ullman-1988, Ramakrishnan-2003]. Sodobnejši sistemi za podajanje informacij (predvsem po telefonu) [Nöth-2004] so bili zgrajeni za različne domene, npr. za podajanje informacij o restavracijah [Jurafsky-1994], o gledališčih [Hoeven-1995], o potovanjih z železnico [Allen-1995, Sturm-1999], o letalskih prevozih [Ipšić-1999, Stallard-2000], o vremenskih napovedih [Zue-2000, Žibert-2003] in elektronski pošti [Walker-2000]. Cilj sodobnejših sistemov za dialog je strategije interakcije čim bolj približati naravnim strategijam, tj. strategijam komunikacije človek-človek. To pomeni, da je treba razvijati in uporabljati najnovejše oz. najučinkovitejše tehnologije na področju obdelave govornega signala, razumevanja naravnega jezika in predstavitve znanja.

Sistem za dialog je običajno modularno zgrajen (slika 1.1). Osnovni moduli so:

MODUL ZA RAZPOZNAVANJE GOVORA

Vhod v modul za razpoznavanje govora je govorni signal uporabnika. Njegova naloga je določiti zaporedje besed, ki jih je uporabnik izgovoril. V prvi fazi lahko to pomeni določitev mreže hipotez o besedah, iz katere modul za razpoznavanje govora šele s pomočjo pomenske analize posameznih zaporedij, ki jo opravlja modul za razumevanje naravnega jezika, izloči zaporedje besed, ki se zdi najverjetnejše.



Slika 1.1: Zgradba sistema za dialog.

MODUL ZA RAZUMEVANJE NARAVNEGA JEZIKA

Vhod v modul za razumevanje naravnega jezika je zaporedje besed, izhod pa njegova pomenska predstavitev. V veliko pomoč pri razumevanju je lahko tudi informacija o poteku dialoga, ki jo dobimo iz modula za vodenje dialoga.

MODUL ZA VODENJE DIALOGA

Modul za vodenje dialoga je osrednji modul vsakega sistema za dialog, saj vodi in uravnava interakcijo med uporabnikom in aplikacijo. Vhod v modul je pomenska predstavitev, ki strukturirano podaja vsebino uporabnikove izjave. Ker ima neposreden dostop do aplikacije, lahko modul za vodenje dialoga na podlagi vhodne pomenske predstavitve spreminja stanje aplikacije (v primeru govornega upravljanja neke naprave) ali pa iz podatkovne zbirke vrača določene informacije (v primeru podajanja informacij).

MODUL ZA TVORJENJE NARAVNEGA JEZIKA

Vhod v modul za tvorjenje naravnega jezika je pomenska predstavitev, ki jo tvori modul za vodenje dialoga. Njegova naloga je, da pomensko predstavitev pretvori v besedilo, zapisano v naravnem jeziku.

MODUL ZA TVORJENJE GOVORA

Naloga modula za tvorjenje govora je, da odziv sistema, ki je podan v naravnem jeziku, pretvori v govor. Ker je uporabnik z govornim izhodom v neposrednem stiku, ta močno vpliva na uporabnikovo oceno učinkovitosti sistema.

Izjemna privlačnost in uporabnost govornih vmesnikov sta povzročili nastanek številnih vizij o njihovem razvoju. Ena izmed najbolj drznih je vizija raziskovalca Rayja

Kurzweila [Kurzweil-1999], ki predvideva naslednji časovni scenarij:

malo po 2000 Dva človeka, ki ne govorita nujno istega jezika, se pogovarjata po telefonu, ki opravlja nalogo prevajalca. Gluhim in naglušnim osebam pomagajo naprave, ki govor pretvarjajo v besedilo. Na telefonske klice odgovarjajo inteligentne telefonske tajnice, ki so sposobne določiti vrsto in pomembnost klica.

leto 2009 Večina besedila je rezultat razpoznavanja vezanega govora. Povsod najdemo sisteme za dialog. Velik del vsakdanjih poslovnih pogajanj poteka med človekom in navidezno osebo, katere navzočnost je pogosto podkrepljena z vidno podobo človeškega obraza. Obstajajo žepni bralniki za slepe in slabovidne ter razpoznavalniki govora za gluhe in naglušne. Telefoni prevajalci za številne pare jezikov so v splošni rabi.

leto 2019 Trodimenzionalni virtualni prikazovalniki, vgrajeni v očala in kontaktne leče, so osnovni komunikacijski vmesnik do drugih ljudi, svetovnega spleta in virtualne resničnosti. Večji del interakcije, v katero so vključeni stroji, poteka s kretnjami in dvosmernim govornim sporazumevanjem. Gluhe in naglušne osebe berejo govorna sporočila na prikazovalnikih, vgrajenih v leče. Velika večina interakcij zajema virtualno osebo.

leto 2029 Trajni in odstranljivi implantati predstavljajo vhodno in izhodno enoto med človeškim uporabnikom in svetovnim računalniškim omrežjem. Izpopolnitev neposrednih nevrnskih povezav, ki služijo za širokopasovno zvezo s človeškimi možgani. Dosegljivi so številni nevrnski implantati za izboljšanje vidnega in slušnega zaznavanja in razumevanja ter izboljšanje spomina in sklepanja. Samodejni agenti se učijo sami. Razširjenost vseh vrst vidnih, slušnih in tipnih komunikacij, ki temeljijo na neposrednih nevrnskih povezavah. Največji del komunikacije, v katero je vključen človek, predstavlja komunikacija med človekom in strojem.

Podana vizija slikovito ponazarja smernice razvoja govornih vmesnikov. Kljub temu da je nastala nedavnega leta 1999, pa ji razvoj že danes ne zmore slediti. Roger K. Moore [Moore-2005] nastale tehnološke zaostanke pripisuje osnovnim slabostim trenutno uveljavljenih metod in pojasnjuje, da bo te mogoče nadomestiti šele, ko se bo razvoj osredotočil na bolj univerzalne sposobnosti, ki bodo neodvisne od aplikacije, okolja in uporabnikov. Sem spada tudi razvoj univerzalnih podatkovnih zbirk, ki bodo sistemom za podajanje informacij omogočale učinkovito in uporabnikom prijazno odgovarjanje.

Raziskovalci Laboratorija za umetno zaznavanje, sisteme in kibernetiko Fakultete za Elektrotehniko v Ljubljani ter raziskovalci Oddelka za računalništvo in informatiko Filozofske fakultete na Reki smo si zastavili cilj razviti sistem za dialog [Žibert-2003], ki bo po telefonu v slovenskem in hrvaškem jeziku dajal informacije o vremenu in vremenski napovedi. Eden izmed razlogov za to odločitev je veliko število turističnih izmenjav med obema državama, drugi pa čedalje manjši delež prebivalcev obeh držav, ki

govori in razume oba jezika. Sistem naj bi bil sposoben odgovarjati na vprašanja o vremenu in vremenski napovedi (tj. splošna vremenska napoved, biovremenska napoved, temperatura, smer in hitrost vetra, zračni tlak, vidljivost, višina in vrsta snega, vzhod in zahod sonca ipd.) za različne kraje in pokrajine v Sloveniji in na Hrvaškem ter za večja mesta drugih evropskih držav.

Ker naj bi sistem omogočal hkratno sporazumevanje v obeh jezikih, tj. v slovenskem in hrvaškem jeziku, bo uporabnikova izjava najprej potovala v modul za identifikacijo jezika, ki bo ugotovil, v katerem jeziku uporabnik govori. V skladu s to ugotovitvijo bo izjava posredovana v zaporedje modulov za razpoznavanje govora [Martinčič-2003] in razumevanje naravnega jezika, pri čemer si bosta oba modula za razumevanje naravnega jezika pri pretvorbi izjave v njeno pomensko predstavitev pomagala tudi s trenutnim stanjem dialoga. Modul za vodenje dialoga [Hajdinjak-2004a, Hajdinjak-2004c] bo (neodvisno od jezika) v zbirki vremenskih podatkov poiskal ustrezne podatke in jih (primerno strukturirane) poslal modulu za tvorjenje naravnega jezika. Ta bo pomensko predstavitev odziva pretvoril v ustrezen (tj. slovenski ali hrvaški) naravni jezik, ki bo nato še sintetiziran [Vesnicer-2003], tj. umetno pretvorjen v govor, in posredovan uporabniku po telefonski zvezi.

Eksperimentalni del razvoja sistema za dialog običajno poteka v treh korakih. V prvem koraku opravimo analizo dejanskih dialogov človek–človek. Na osnovi te analize konstruiramo enega ali celo več sistemov *Čarovnik iz Oza* [Dahlbäck-1993, Hajdinjak-2004b], ki služijo kot ogrodje za izvedbo istoimenskega eksperimenta. Zadnji korak je vrednotenje, izboljševanje in dopolnjevanje sistema za dialog, ki poteka z uporabo podatkov, pridobljenih od dejanskih uporabnikov.

Doktorska disertacija predstavlja del raziskovalnega dela, opravljenega v okviru razvoja dvojezičnega samodejnega sistema za podajanje informacij o vremenu, zajema pa dve tematski področji:

- vrednotenju učinkovitosti sistemov za dialog z ogrodjem PARADISE in
- posplošitev relacijske algebre na teorijo kategorij, ki vodi do razvoja sodelujočega podatkovnega modela.

Zaradi osredotočenosti k univerzalnim sposobnostim sistemov za dialog, bomo ugotovitve, sklepe in rezultate podali čim bolj neodvisno od aplikacije, okolja in uporabnikov.

2 Eksperiment Čarovnik iz Oza

2.1 Uvod

2.2 Prvi eksperiment Čarovnik iz Oza

2.3 Drugi eksperiment Čarovnik iz Oza

2.4 Modul za vodenje dialoga

2.5 Sodelujoča podatkovna zbirka

Opišemo eksperiment Čarovnik iz Oza, v katerem s pomočjo človeka čarovnika simuliramo delovanje še nedokončanega sistema za podajanje informacij o vremenu. Eksperiment Čarovnik iz Oza smo uporabili dvakrat. Prvi eksperiment je služil za zbiranje podatkov za izdelavo še nedokončanih modulov v sistemu za podajanje informacij o vremenu, drugi pa za zbiranje podatkov za vrednotenje učinkovitosti modula za vodenje dialoga.

Za modeliranje dialoga smo uporabili tehniko polnjenja predalčkov, strategijo vodenja dialoga pa smo formalizirali z uporabo konverzacijske teorije iger. V takem modelu je dialog sestavljen iz konverzacijskih iger in konverzacijskih potez, ki so udejanjene kot rekurzivne mreže prehodov.

Za predstavitev znanja v drugem sistemu Čarovnik iz Oza smo uporabili intuicionistično modalno logiko. Dobljen razred podatkovnih modelov, imenovan dialog mIPC Kripkejevi modeli, podpira vrsto tehnik sodelujočega odgovarjanja, tudi relaksacijo.

2.1 Uvod

Številni raziskovalci menijo, da naj bi komunikacija človek–stroj čim bolj sledila komunikaciji človek–človek. Toda komunikacija človek–stroj in komunikacija človek–človek se tako zelo razlikujeta, da podatki iz slednje hitro postanejo neustrezen in nezanesljiv vir informacij za nekatere pomembne vidike oblikovanja sistemov za dialog, kot sta na primer način in kompleksnost interakcije. Fraser in Gilbert [Fraser-1991] ugotavljata, da čeprav se je boljše zanašati na analizo komunikacije človek–človek kot pa le na intuicijo, dejstvo, da se komunikacija človek–človek in komunikacija človek–stroj razlikujeta, ostane in presenetljivo bi bilo, če bi obe sledili enakim pravilom. Analiza komunikacije človek–človek je sicer dobra osnova za določitev nalog in slovarja besed, ki naj jih sistem za dialog razume, ne pa tudi za ostale vidike interakcije med človekom in strojem [Smith-1997]. Raziskovalci so tako ujeti v začaran krog – za konstrukcijo sistema za dialog po eni strani potrebujejo značilnosti komunikacije človek–stroj, po drugi strani pa, dokler sistem za dialog še ni dograjen, ni mogoče vedeti, kako bodo dialogi potekali.

Trenutno najboljša alternativa za zbiranje podatkov, ki izražajo jezik komunikacije človek–stroj, je t. i. *eksperiment Čarovnik iz Oza*. Ime tehnike se nanaša na moža za zaveso v knjigi in filmu *Čarovnik iz Oza*. V teh eksperimentih so uporabniki prepričani, da se pogovarjajo s strojem – računalnikom, kar pa ni res. V resnici za računalnikom sedi človek (čarovnik), ki vsaj delno simulira delovanje sistema za dialog. V nekaterih primerih [Whittaker-1989, Eskenazi-1999] čarovnik simulira celoten sistem, v drugih [Dahlbäck-1993, Kim-1997] pa le del sistema. Ugotovljeno je bilo [Whittaker-1989, Fraser-1991, Dahlbäck-1993], da podatki, pridobljeni z eksperimentom *Čarovnik iz Oza*, natančneje izražajo jezik komunikacije človek–stroj kot podatki, pridobljeni v človeških dialogih. Glavni razlog za to je prilagajanje udeležencev dialoga jezikovnim sposobnostim sogovornika.

Eksperiment *Čarovnik iz Oza* pa ima tudi nekatere slabosti:

1. Konstrukcija kakovostnega sistema za simulacijo in učenje čarovnikov je zahtevna naloga. Dahlbäck, Jönsson in Ahrenberg [Dahlbäck-1993] to utemeljujejo z dejstvom, da smo ljudje fleksibilni, računalniki pa konsistentni; da ljudje počasi tipkamo, računalniki pa se odzivajo hitro; da računalniki nikoli ne delajo majhnih napak (tipkarske napake), ljudje pa jih delamo pogosto. Torej če želimo, da bodo uporabniki ves čas dialoga ostali v prepričanju, da se pogovarjajo z računalnikom, čarovnik ne sme le sedeti za računalnikom in tipkati svojih odgovorov, ki so nato umetno pretvorjeni v govor. Vmesnik, pred katerim sedi, mora čarovniku pomagati pri njegovem delu, tako da čim bolj zmanjša potrebo po tipkanju in čim bolj poveča konsistentnost njegovih odzivov.
2. Udeleženci eksperimenta le igrajo vloge resničnih uporabnikov, zato med dialogom stežka pokažejo svoje potrebe, motivacije, izhode in ovire. To težavo lahko do neke mere odpravimo z uvedbo vnaprej določenih scenarijev ali situacij. Ker jih ponavadi zastavijo raziskovalci sami, to ni najidealnejši način ugotavljanja nalog, ki naj jih bo sistem sposoben opraviti.

3. Pojavljajo se tudi etična vprašanja. To težavo lahko delno zmanjšamo, če udeležencem eksperimenta na koncu povemo, kaj smo počeli, zakaj smo tako ravnali in ne nazadnje, da bomo podatke nemudoma uničili, če oni tako želijo.

Eksperimentalni del razvoja sistema za dialog običajno poteka v treh korakih. Prvi korak je analiza dejanskih dialogov človek–človek. Na osnovi te analize konstruiramo enega ali celo več sistemov Čarovnik iz Oza, ki služijo kot ogrodje za izvedbo istoimenskega eksperimenta. Zadnji korak je izboljševanje in dopolnjevanje sistema za dialog, ki poteka z uporabo podatkov, pridobljenih od dejanskih uporabnikov.

V 3. poglavju o vrednotenju učinkovitosti sistemov za dialog se bomo sklicevali na podatke, pridobljene v dveh eksperimentih Čarovnik iz Oza [Hajdinjak-2004b], ki sta potekala v sklopu raziskav, predstavljenih v magistrskem delu [Hajdinjak-2004a]. Edina razlika med obema sistemoma Čarovnik iz Oza je način vodenja dialoga – v prvem sistemu je to nalogo opravljal čarovnik, v drugem pa modul za samodejno vodenje dialoga.

Oba sistema Čarovnik iz Oza, ki po telefonu dajeta informacije o vremenu in vremenski napovedi [Hajdinjak-2003a], s katerima smo izvajali eksperimente Čarovnik iz Oza, sta modularno zasnovana. Vsi moduli so povezani z (grafičnim) vmesnikom, za katerim je sedel človek čarovnik in simuliral del sistema za dialog. Sistem smo udejanjili na osebem računalniku z vgrajeno ISDN DIVA Server BRI-2M PCI kartico, s katero smo komunicirali prek posebnega računalniškega programa, ki je omogočal vzpostavljanje in prevzemanje telefonskih pogovorov, poslušanje in snemanje pogovorov, kodiranje in dekodiranje ISDN formata zvočnih datotek, pošiljanje wave datotek po telefonski zvezi ipd.

V zgodnjih fazah razvoja sistema za dialog je primerneje, če avtomatičnega razumevanja govora ne vključimo v sistem, saj v tem primeru ne bomo omejeni z narečjem, spolom in starostjo uporabnikov. V literaturi najdemo dva načina simuliranja razumevanja govora – napake pri razumevanju govora lahko sistematsko vključimo v besedilo, na katero se čarovnik odziva [Pirker-1999, Möller-2005], ali pa simuliramo tako rekoč popolno razumevanje govora [Dahlbäck-1993, Kim-1997, Eskenazi-1999]. Za ta način smo se odločili tudi mi, čeprav smo se zavedali, da so napake pri razumevanju govora izjemno pomemben vidik uporabnikove interakcije s sistemom. Za ta pristop smo se odločili, ker je napake pri razumevanju govora, še posebej pa napake pri razpoznavanju govora, zelo težko ustrezno simulirati. Naš čarovnik je zato poslušal uporabnika, ne da bi ga pri tem kakor koli omejeval.

2.2 Prvi eksperiment Čarovnik iz Oza

V prvem eksperimentu Čarovnik iz Oza [Hajdinjak-2003a] smo uporabljali sistem, sestavljen iz naslednjih komponent:

- platforma za ISDN telefonijo,

- grafični vmesnik, ki je čarovniku omogočal vodenje dialoga,
- relacijska zbirka vremenskih podatkov [Hajdinjak-2002a, Hajdinjak-2002b], do katere je (z uporabo grafičnega vmesnika) dostopal čarovnik,
- modul za tvorjenje naravnega jezika in
- modul za tvorjenje slovenskega govora.

Eksperiment je potekal tako, da je čarovnik sedel za grafičnim vmesnikom, poslušal izjave uporabnika, v skladu s katerimi je po lastni presoji izbiral odzive na grafičnem vmesniku in s tem po potrebi dostopal do relacijske zbirke vremenskih podatkov. Odzivi so potovali v modul za tvorjenje naravnega jezika, katerega izhod (naravno besedilo) je potoval v modul za tvorjenje slovenskega govora, tega pa je sistem po telefonski zvezi poslal do uporabnika.

Grafični vmesnik, ki je čarovniku omogočal vodenje dialoga, je bil oblikovan tako, da so bili odzivi čarovnika približno tako hitri in konsistentni kot odzivi računalnika. Zasnovan je bil kot uporabniku (čarovniku) prijazna spletna aplikacija, ki je vsebovala spletne obrazce, aktivne slike, slikovna polja, spustne sezname, ipd. in je omogočala

- ↪ izbiro krajev s pomočjo aktivnih zemljevidov,
- ↪ predvajanje vnaprej pripravljenih zvočnih datotek,
- ↪ uporabo bližnjic ...

Naloga čarovnika v prvem eksperimentu Čarovnik iz Oza je bila igrati vlogo modulov za razpoznavanje govora in razumevanje naravnega jezika, s pomočjo grafičnega vmesnika pa tudi vlogo modula za vodenje dialoga. Za ta eksperiment smo se odločili v prvi fazi razvoja sistema za podajanje informacij o vremenu, da bi pridobili podatke, na osnovi katerih bi potem gradili omenjene module, ki jih je čarovnik v eksperimentu nadomeščal.

V eksperimentu je sodelovalo 76 uporabnikov, in sicer 38 žensk in 38 moških. Pri izbiri smo pazili, da bi uporabniki predstavljali čim bolj reprezentativen vzorec (spol, starost, izobrazba, narečje, telefonska zveza, okolje telefonskega pogovora). Povprečna starost uporabnikov je bila 34 let, povprečna izobrazba pa srednja šola. Zastopanih je bilo vseh osem slovenskih narečnih skupin, pazili pa smo tudi na vrste telefonskih zvez (analogna, ISDN, GSM) in na okolje (tiho okolje, šolski hodnik, ulica, prostor z več ljudmi, menza, vklopljen radio/TV-sprejemnik ipd.), ki je uporabnike v času telefonskega pogovora s sistemom Čarovnik iz Oza obdajalo.

Pred začetkom eksperimenta smo udeležencem povedali, da se bodo pogovarjali z računalnikom, tj. s samodejnim sistemom za podajanje informacij o vremenu, in jim dali ustna navodila o splošni funkcionalnosti sistema. Ker smo želeli po eni strani pridobiti čim več posnetkov, po drugi strani pa, kljub igranju vloge dejanskih uporabnikov, zagotoviti čim bolj realne in raznolike dialoge, smo vsakemu izmed udeležencev zastavili dve nalogi. Prva naloga je bila pridobiti določeno informacijo, druga pa je zajemala

določen scenarij oz. situacijo, ki naj bi si jo udeleženec poskušal zamisliti. Primera takih nalog sta:

1. Poskušajte ugotoviti, ali v Ljubljani sije sonce.
2. Ob koncu tedna bi radi šli v hribe. Kaj vas zanima?

in

1. Poskušajte ugotoviti, kakšne temperature lahko pričakujemo jutri.
2. Načrtujete izlet s kolesom. Kaj vas zanima?

Udeležencem smo omogočili tudi lastno izbiro vprašanj – povedali smo jim, da lahko po opravljenih nalogah dialog s sistemom nadaljujejo.

Po pogovoru s sistemom Čarovnik iz Oza smo uporabnike prosili, naj izpolnijo vprašalnik, katerega prvi del je vseboval vprašanja o spolu, starosti, izobrazbi, zaposlitvi, narečju, vrsti telefonske zveze in okolju, ki jih je v času dialoga s sistemom obdajalo. Drugi del vprašalnika se je nanašal na dialog s sistemom in je, poleg vprašanja, ali so dobili odgovor na prvo nalogo, zajemal različne vidike njihove interakcije s sistemom (tabela 3.1). Vprašanja, s pomočjo katerih so udeleženci ocenjevali sistem Čarovnik iz Oza, so del ogrodja PARADISE, ki ga bomo predstavili v 3. poglavju, sprašujejo pa po učinku modula za tvorjenje govora, učinku modula za razpoznavanje govora, težavnosti pridobivanja informacij, hitrosti interakcije, izkušenosti uporabnikov, ustreznosti odzivov sistema, pričakovanem obnašanju sistema in načrtovani rabi sistema v prihodnosti. Odgovori so bili podani z lestvico od 1 (nikakor se ne strinjam) do 5 (popolnoma se strinjam). Povprečne ocene uporabnikov so podane v tabeli 2.1.

	WOZ1	WOZ2
Tvorjenje govora	4.42	4.29
Razpoznavanje govora	4.51	4.29
Pridobivanje informacij	4.27	3.74
Hitrost interakcije	3.94	3.76
Izkušnost uporabnikov	4.40	4.28
Ustreznost odzivov	4.23	3.76
Pričakovano obnašanje	4.31	4.04
Raba v prihodnosti	3.99	3.78

Tabela 2.1: Povprečne ocene uporabnikov v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.

V eksperimentu smo opazili nekaj zanimivosti. Prva je ta, da je večina uporabnikov svoja vprašanja oblikovala zelo podobno, ponavadi *Zanima me ...*, *Rad/Rada bi vedel/vedela ...*, *Mi lahko (prosim) poveste ...*, *Ali mi lahko poveste ...*, ipd. Druga zanimivost je prilagajanje uporabnikovega vedenja pričakovanim jezikovnim sposobnostim sistema. V našem eksperimentu smo ugotovili, da je bilo prvo vprašanje, ki so ga uporabniki zastavili, ponavadi veliko daljše in veliko manj jedrnato kot vprašanja, ki so sledila. Primeri dolgih in nejedrnatih začetnih vprašanj so na primer:

SISTEM:	<i>Dober dan. Pogovarjate se s sistemom za podajanje informacij o vremenu. Kakšno je vaše vprašanje?</i>
UPORABNIK:	Mi lahko prosim poveste, kakšna bo danes vidljivost na Štajerskem?
SISTEM:	<i>Počakajte trenutek, prosim ... [čarovnik izbira svoj odziv]</i> <i>Na Štajerskem bo danes podnevi vidljivost nad 10 km. Imate še kakšno vprašanje?</i>
UPORABNIK:	A bo kaj snežilo?
SISTEM:	<i>Kateri kraj vas zanima?</i>
UPORABNIK:	Štajerska.
SISTEM:	<i>Počakajte trenutek, prosim ... [čarovnik izbira svoj odziv]</i> <i>Ne, na Štajerskem danes podnevi – razjasnitve. Imate še kakšno vprašanje?</i>
UPORABNIK:	Kakšno pa bo vreme v prihodnjih dneh na Poljskem?
SISTEM:	<i>Imam podatke za danes.</i>
UPORABNIK:	Torej, kako bo danes na Poljskem?
SISTEM:	<i>Počakajte trenutek, prosim ... [čarovnik izbira svoj odziv]</i> <i>V Varšavi na Poljskem je oblačno, temperatura zraka je -6 stopinj Celzija. Imate še kakšno vprašanje?</i>
UPORABNIK:	A je kaj snega na Poljskem?
SISTEM:	<i>Te informacije ne ponujam. Imate morda kakšno drugo vprašanje?</i>
UPORABNIK:	Ne, najlepša hvala. Na svidenje.
SISTEM:	<i>Hvala lepa za sodelovanje. Na svidenje.</i>

Tabela 2.2: Primer dialoga med uporabnikom in prvim sistemom Čarovnik iz Oza.

- Jaz moram danes na Primorsko, pa me je malo strah burje. Zdaj me pa zanima, kje začenja pihati, a na vetrišču ali šele s Svete gore navzdol proti Gorici.
- Glejte. Trenutno se peljem z Jesenic proti Ljubljani. Razmere na cesti so obupne. Vseskozi piha močan veter s smeri vzhoda. Zdaj me pa zanima, kakšen veter piha v Ljubljani. Se pravi, moč pa smer, od kod piha.
- Danes bi radi prišli v Ljubljano na izlet. Malo bi se sprehajali, pa na ljubljanski grad bi šli radi. Pa nas zanima, če je v Ljubljani kaj sonca, če je lep razgled z gradu.

V primerih, ko je sistem uporabnike prosil, naj vprašanje ponovijo, je njihov govor pogosto postal bolj artikuliran, počasnejši in/ali glasnejši. Do podobnih ugotovitev so prišli tudi drugi raziskovalci [Zoltan-Ford-1991], da namreč jezikovne sposobnosti sistema za dialog opazno vplivajo na vedenje uporabnikov. Cohen in Oviatt [Cohen-1995] pa trdita, da bodo uporabniki svoje besedišče zelo verjetno prilagodili sistemovemu, če bodo v njem zaznali omejitve. Tretja zanimivost, ki smo jo opazili v prvem eksperimentu Čarovnik iz Oza, so smeh, šale in duhovite opazke, ki so jih nekateri uporabniki delali na račun sistema, medtem ko je čarovnik posredoval svoj odziv. Primeri takih opazk so:

- ↪ Kakšen glas ... Grozno ...
- ↪ Išče po računalniku ...
- ↪ Razmišlja ...

Taki odzivi in stranske opazke so najbrž verjetnejši v testiranjih sistemov za dialog kot pa v komercialnih sistemih, saj v takih testiranjih uporabniki ponavadi igrajo le vloge dejanskih uporabnikov, niso časovno omejeni in ne odgovarjajo za stroške telefonskega pogovora. Primer dialoga med uporabnikom in prvim sistemom Čarovnik iz Oza je podan v tabeli 2.2.

Zbirko spontanega govora, ki smo jo pridobili v prvem eksperimentu Čarovnik iz Oza, smo poimenovali SSSQ [Hajdinjak-2003b]. Kratica SSSQ se nanaša na angleško besedno zvezo *Slovenian Spontaneous Speech Queries*. Zbirko smo označili z orodjem TRANSCRIBER [Barras-2001], pri čemer smo poleg oznak besed uporabili še posebne oznake za narečne besede, premore in negovorne zvoke (vdih, izdih, vzdihljaj, tlesk, smeh, pih, kašelj in različne zvoke, ki nastajajo pri obotavljanju oz. razmišljanju).

Zgradba te zbirke spontanega govora je prikazana v tabeli 2.3. Sestavljena je iz 76 dialogov, ki vsebujejo 662 izjav uporabnikov. Skupna dolžina dialogov znaša 103.02 minute, dolžina človeškega govora pa 41.18 minute.

število dialogov	76
število izjav uporabnikov	662
število različnih besed	920
število vseh besed	5442
dolžina vseh dialogov	103.02 min
dolžina človeškega govora	41.18 min

Tabela 2.3: Zgradba SSSQ zbirke spontanega govora.

Dialoge, pridobljene v prvem eksperimentu Čarovnik iz Oza, ki odsevajo strukturo komunikacije človek–stroj, smo podrobno analizirali. Rezultati analize so podani v poglavju 4.

2.3 Drugi eksperiment Čarovnik iz Oza

Na osnovi podatkov iz prvega eksperimenta Čarovnik iz Oza smo zgradili modul za vodenje dialoga (razdelek 2.4) in ga vgradili v sistem, s katerim smo izvajali drugi eksperiment Čarovnik iz Oza. Cilj in namen drugega eksperimenta Čarovnik iz Oza je bil pridobiti podatke, ki bi poleg podatkov iz prvega eksperimenta tvorili osnovo za vrednotenje učinkovitosti modula za vodenje dialoga [Hajdinjak-2004c]. Sistem, ki smo ga pri tem uporabili, je bil sestavljen iz naslednjih komponent:

- platforma za ISDN telefonijo,
- modul za vodenje dialoga,
- sodelujoča zbirka vremenskih podatkov, do katere je dostopal modul za vodenje dialoga,
- modul za tvorjenje naravnega jezika in
- modul za tvorjenje slovenskega govora.

Naloga čarovnika v drugem eksperimentu Čarovnik iz Oza je bila igrati vlogo modulov za razpoznavanje govora in razumevanje naravnega jezika, ne pa tudi vloge modula za vodenje dialoga kot v prvem eksperimentu. V tem eksperimentu je čarovnik sedel pred vmesnikom modula za vodenje dialoga in s tipkovnico vnašal pomensko predstavitev uporabnikove izjave, ki bo v končnem sistemu izhod iz modula za razumevanje naravnega jezika. Vse nadaljnje delo (vodenje dialoga, iskanje podatkov, tvorjenje naravnega jezika, tvorjenje govora) je opravljal sistem. Oba sistema Čarovnik iz Oza sta se torej razlikovala le po načinu vodenja dialoga (čarovnik oz. modul za vodenje dialoga) in vrsti podatkovne zbirke (relacijska oz. sodelujoča podatkovna zbirka). Sodelujoča podatkovna zbirka vremenskih informacij, ki smo jo uporabili v drugem eksperimentu Čarovnik iz Oza, je predstavljena v razdelku 2.5.

V eksperimentu je sodelovalo 68 uporabnikov, 29 žensk in 39 moških, katerih povprečna starost je bila 32 let. Čeprav je 17 izmed njih že sodelovalo v prvem eksperimentu Čarovnik iz Oza in so zato bili malo bolj izkušeni od drugih, med vedenjem obeh skupin nismo opazili nobene razlike.

Uporabnike (tudi tiste, ki so že sodelovali v prvem eksperimentu) smo, tako kot v prvem eksperimentu, prepričali, da se bodo pogovarjali z računalnikom, tj. s samodejnim sistemom za podajanje informacij o vremenu, in jim dali ustna navodila o splošni funkcionalnosti sistema. Vsakemu izmed udeležencev smo zastavili dve nalogi, podobno kot v prvem eksperimentu, od katerih je bila prva pridobiti določeno informacijo, druga pa scenarij oz. situacija, ki naj bi si jo udeleženec poskušal zamisliti. Po končanem pogovoru s sistemom smo uporabnike prosili, naj izpolnijo enak vprašalnik kot v prvem eksperimentu. Del vprašalnika, ki zajema vprašanja, nanašajoča se na ocenjevanje sistema, je podan v tabeli 3.1.

Povprečne ocene udeležencev drugega eksperimenta Čarovnik iz Oza so podane v tabeli 2.1. Zanimivo je, da so udeleženci drugega eksperimenta slabše ocenili prav vse vidike svoje interakcije s sistemom, tudi učinek modulov za tvorjenje in razpoznavanje govora, ki sta v obeh eksperimentih ostala enaka. Glede na to da so tako storili tudi tisti, ki so sodelovali v obeh eksperimentih Čarovnik iz Oza, razlog ne more biti večja kritičnost udeležencev. Zelo verjetno pa so slabše ocene posledica manjšega zadovoljstva z nekaterimi posameznimi vidiki učinkovitosti sistema. Najbolj opazna razlika je v ocenah težavnosti pridobivanja informacij (4.27 oz. 3.74) in ustreznosti odzivov sistema (4.23 oz. 3.76), pri čemer je treba poudariti, da so bile slabše ocene nekaterih vidikov učinkovitosti sistema vsekakor pričakovane, saj bi v nasprotnem primeru modul za vodenje dialoga v drugem eksperimentu bolj opravljal svojo nalogo kot človeški operater v prvem eksperimentu.

Primer dialoga med uporabnikom in drugim sistemom Čarovnik iz Oza je podan v tabeli 2.4.

<i>S1:</i>	<i>Pozdravljeni. Pogovarjate se s sistemom za podajanje informacij o vremenu. Kakšno je vaše vprašanje?</i>
<i>U1:</i>	<i>Zanima me, kakšno vreme bo jutri v Sloveniji.</i>
<i>S2:</i>	<i>V Sloveniji jutri čez dan – padavine bodo ponehale, delno razjasnitve. Želite vedeti še kaj?</i>
<i>U2:</i>	<i>Ja. Zanima me še vreme v gorah.</i>
<i>S3:</i>	<i>Ali sprašujete za jutri čez dan?</i>
<i>U3:</i>	<i>Kako?</i>
<i>S4:</i>	<i>Niste mi povedali vseh potrebnih podatkov ali pa jih nisem razumel. Če vas zanima kaj drugega, vprašajte, sicer pa prosim odgovorite na naslednje vprašanje. Ali sprašujete za jutri čez dan?</i>
<i>U4:</i>	<i>Ne, zanima me vreme za ta trenutek.</i>
<i>S5:</i>	<i>O vremenu v gorah ta trenutek žal nimam podatkov. Imam pa podatke o vremenu ta trenutek na Kredarici, na Voglu, na Kaninu, na Krvavcu in v Predelu. Potem lahko izbirate med podatki o vremenu v gorah za danes čez dan in jutri čez dan. Izbirate lahko tudi med podatki o vremenski napovedi v gorah ta trenutek.</i>
<i>U5:</i>	<i>Povejte mi prosim za Vogel.</i>
<i>S6:</i>	<i>Ali sprašujete za ta trenutek o vremenu?</i>
<i>U6:</i>	<i>Ja.</i>
<i>S7:</i>	<i>Na Voglu ta trenutek – megla. Vam lahko še kako pomagam?</i>
<i>U7:</i>	<i>Ja. Povejte mi, kakšna je temperatura zraka in kakšna je višina snega na Voglu.</i>
<i>S8:</i>	<i>Na Voglu ta trenutek – skupna višina snega je 180 centimetrov, temperatura zraka je -7 stopinj Celzija. Imate še kakšno vprašanje?</i>
<i>U8:</i>	<i>Ja. Kaj vse mi lahko poveste o trenutnem vremenskem stanju v Ljubljani?</i>
<i>S9:</i>	<i>Za ta trenutek v Ljubljani vam lahko ponudim 9 vrst informacij kot na primer podatek o vremenu, vlažnosti zraka, temperaturi zraka, hitrosti vetra, smeri vetra, vrednosti zračnega tlaka, indeksu požarne ogroženosti, vrsti požarne ogroženosti in tako naprej. Ali vas kakšna od teh informacij zanima?</i>
<i>U9:</i>	<i>Ne. To bi bilo vse.</i>
<i>S10:</i>	<i>Hvala lepa za sodelovanje. Na svidenje.</i>

Tabela 2.4: Primer dialoga med uporabnikom (U) in drugim sistemom Čarovnik iz Oza (S).

Zbirko spontanega govora, ki smo jo pridobili v drugem eksperimentu Čarovnik iz Oza, smo (iz podobnega razloga kot v prvem eksperimentu) poimenovali SSSQ2 in jo označili v skladu z oznakami zbirke SSSQ. Zgradba SSSQ2 zbirke spontanega govora je prikazana v tabeli 2.5. Sestavljena je iz 68 dialogov, ki vsebujejo 885 izjav uporabnikov. Skupna dolžina dialogov znaša 148.92 minute, dolžina človeškega govora pa 50.51 minute. Skupno število različnih besed (zajete so tudi govorne različice) v obeh zbirkah spontanega govora (SSSQ in SSSQ2) je 1229. Podatkovni

zbirki SSSQ in SSSQ2, pridobljeni v obeh eksperimentih Čarovnik iz Oza, sta po številnosti uporabnikov primerljivi [Dahlbäck-1993, Walker-2001a, Walker-2001b] ali celo presejata [Walker-1997b, Walker-1998, Walker-1999, Pirker-1999, Eskenazi-1999, Kamm-1999, Whittaker-2002, Litman-2002] podobne zbirke. Redke študije komunikacije človek–stroj ali vrednotenja učinkovitosti sistemov za dialog so temeljile na bogatejših zbirkah [Kim-1997, Larsen-2003].

število dialogov	68
število izjav uporabnikov	885
število novih besed	309
število različnih besed	716
število vseh besed	5403
dolžina vseh dialogov	148.92 min
dolžina človeškega govora	50.51 min

Tabela 2.5: Zgradba SSSQ2 zbirke spontanega govora.

Tudi dialoge, pridobljene v drugem eksperimentu Čarovnik iz Oza, smo podrobno analizirali. Rezultati analize so podani v poglavju 4.

Da bi vsaj delno odpravili etično vprašljivost eksperimentov Čarovnik iz Oza, smo po obeh eksperimentih udeležencem povedali, kaj smo počeli, zakaj smo tako ravnali, in jih vprašali za dovoljenje uporabe pridobljenih podatkov v raziskovalne namene. Prav vsi so pokazali razumevanje in odobravanje teh eksperimentov, dovolili pa so nam tudi uporabo posnetkov in drugih podatkov, ki smo jih v eksperimentih pridobili.

2.4 Modul za vodenje dialoga

Eden izmed ciljev razvoja sistema za podajanje informacij o vremenu je bil razviti modul za vodenje dialoga, ki bi bil od jezika neodvisen in čim enostavneje prenosljiv na druge domene podajanja informacij. Pretehtali smo učinkovitost uveljavljenih tehnik modeliranja dialoga [Hajdinjak-2004a] in po temeljitem razmisleku prišli do sklepa – dialog modelirati s *tehniko polnjenja predalčkov* [Goddeau-1996]. Tehnika polnjenja predalčkov rešuje številne težave dialogov, modeliranih s tehniko končnih avtomatov, in predstavlja ta čas najobičajnejši način modeliranja dialogov v govornih vmesnikih. Vhod v modul za dialog je v tem primeru pomenska predstavitev uporabnikove izjave, ki je podana s predalčki in njihovimi elementi. Napolniti predalčke pomeni pridobiti ustrezne tipe podatkov, ki bodo omogočili dostop do podatkovne zbirke ali izvedbo kake druge akcije. Prednost te tehnike v primerjavi s tehniko končnih avtomatov je boljša prenosljivost na druge domene, kar je omogočeno z ločitvijo naloge in strategije vodenja dialoga; naloga je napolniti predalčke, strategija pa predstavlja način, kako to naredimo. Zaradi relativne preprostosti domene vremenskih podatkov, ki je sicer skupna skoraj vsem domenam za podajanje informacij, smo definirali le tri različne predalčke:

1. *podatek o kraju,*
2. *podatek o času in*
3. *vrsta informacije.*

Strategijo vodenja dialoga smo modelirali s *konverzacijsko teorijo iger*, ki izhaja iz množice raziskav [Power-1979, Houghton-1987, Kowtko-1992, Allen-1997, Poesio-1997, Stolcke-2000], kjer so dialogi sestavljeni iz iger, ki jih načrtujejo razumski agenti. Namen raziskav, predstavljenih v [Power-1979] in [Houghton-1987], je bil izdelati teorijo, ki bo opisovala, kako cilji sogovornika vplivajo na potek dialoga. Na podlagi teh ugotovitev sta Kowtko in Isard [Kowtko-1992] dialoge strukturirala na dve funkcijski ravni:

↪ *konverzacijske igre in*

↪ *konverzacijske poteze.*

Konverzacijske igre povezujemo z željami oz. konverzacijskimi cilji, kot je na primer cilj pridobiti določeno informacijo, in so sestavljene iz zaporedja izjav, ki se začnejo s pobudo in končajo, ko je cilj igre dosežen ali igra prekinjena. Sestavne dele konverzacijskih iger imenujemo konverzacijske poteze. To so izjave, deli izjav ali množice izjav, ki izražajo isto namero, kot je na primer potrditev ali preverjanje. Konverzacijske igre lahko poljubno gnezdimo eno v drugi, če le cilj notranje konverzacijske igre služi za dosego cilja zunanje igre.

Najbolj znana uporaba konverzacijske teorije iger je kodiranje obsežne (15 ur) zbirke spontanih, k cilju usmerjenih govornih dialogov človek–človek [Carletta-1996a], kjer je dodana še ena funkcijska raven, namreč raven *dogovorov*, tj. delov dialoga, ki predstavljajo večje korake pri doseganju končnega cilja. Študija zbirke teh dialogov je pokazala, da v konverzacijskih igrah vprašanjem v veliki večini sledijo odgovori, izjavam pa odobritve ali zavrnitve. Na osnovi te študije avtorji razlikujejo tri osnovne tipe konverzacijskih potez:

↪ *začetne poteze,*

↪ *odzivne poteze in*

↪ *poteze pripravljenosti.*

Začetne poteze se pojavljajo na začetku konverzacijskih iger in v dialog uvajajo nove namere, so pobudniki. Odzivne poteze se pojavljajo znotraj konverzacijskih iger in služijo za izpolnitev pričakovanj in želja, ki se v dialogu pojavijo, poteze pripravljenosti pa se pojavljajo potem, ko je bila neka konverzacijska igra končana, in dialog pripravijo na začetek nove igre.

Ideja pojmovanja konverzacijskih iger in konverzacijskih potez se je pri modeliranju dialoga izkazala za zelo uporabno, in sicer z opisnega [Power-1979, Houghton-1987,

Kowtko-1992, Carletta-1996a] in algoritemskega [Lewin-2000] vidika. Konverzacijska teorija iger se tako ukvarja predvsem z modeliranjem dialogov človek–človek in dialogov človek–stroj. Uporaba te teorije za modeliranje dialoga v samodejnih sistemih je bila predlagana v projektu TRINDI (Task Oriented Instructional Dialogue) [Larsson-2000, Lewin-2000].

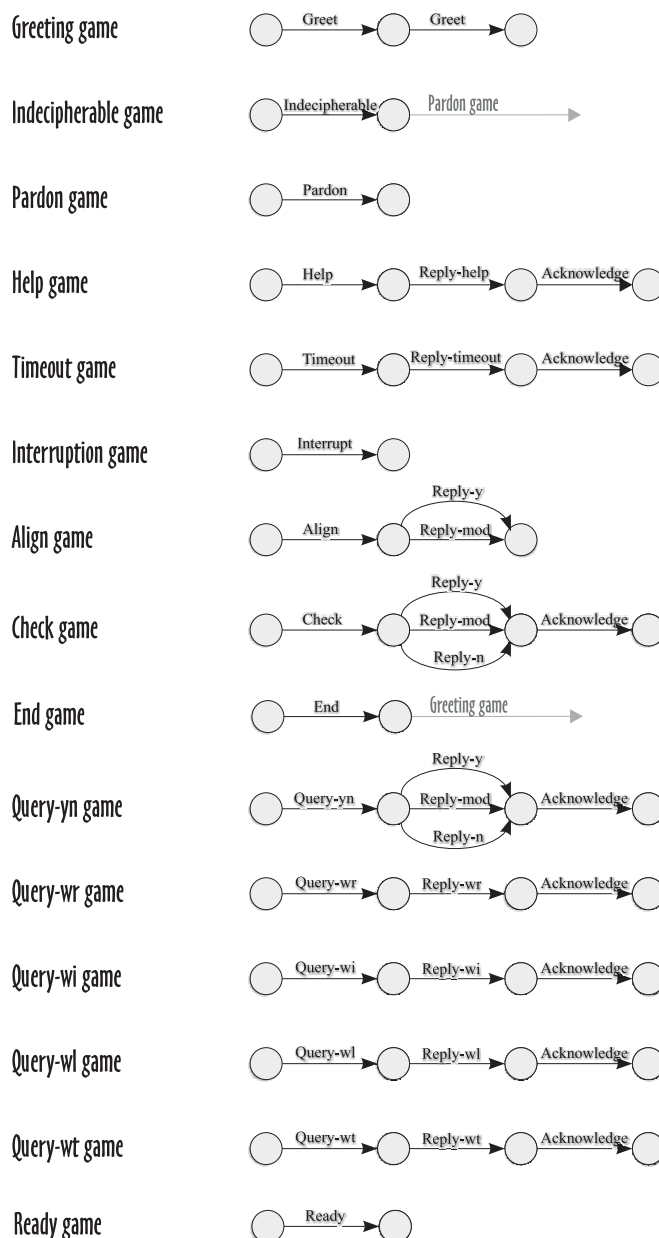
Tudi mi smo, podobno kot [Carletta-1996a], ločili tri osnovne tipe konverzacijskih potez: začetne poteze, odzivne poteze in poteze pripravljenosti. Množici konverzacijskih iger in konverzacijskih potez, udejanjenih v projektu TRINDI [Larsson-2000], smo precej razširili, da bi zagotovili večjo prenosljivost sistema na druge domene. Definirali smo 14 kategorij začetnih in 10 kategorij odzivnih potez ter poteze pripravljenosti READY, ki naznanjajo konec prejšnje igre in dialog pripravljajo na začetek nove konverzacijske igre. Kategorije začetnih konverzacijskih potez, ki smo jih udejanjili v modulu za vodenje dialoga v sistemu za podajanje informacij o vremenu, so:

- GREET
Pozdravi (npr. *Dober dan.* ali *Na svidenje.*).
- INDECIPHERABLE
Izjave uporabnika, ki jih sistem ne razpozna ali ne razume.
- PARDON
Izjave, s katerimi uporabnik ali sistem prosita sogovornika, naj ponovi prejšnjo izjavo (npr. *Prosim?* ali *Ali lahko ponovite vprašanje, prosim?*).
- HELP
Izjave, s katerimi uporabnik prosi za pomoč, ko se v dialogu ne znajde oz. ne ve, kako nadaljevati (npr. *Kaj pa znaš?*).
- TIMEOUT
Sem spadajo tisti odzivi uporabnika, ko sistem v vnaprej določenem času ne zazna govora.
- INTERRUPT
Izjave, s katerimi uporabnik prekine tekočo konverzacijsko igro (npr. *Ne, to me pa ne zanima.*).
- ALIGN
Izjave, s katerimi uporabnik preveri, ali se njegovo prepričanje ujema s prepričanjem sistema.
- CHECK
Izjave, s katerimi sistem prosi za potrditev informacij, za katere sicer že verjame, da držijo, vendar o tem ni čisto prepričan. Te poteze pokrivajo sklepanje na podlagi zgodovine dialoga (npr. *Ali sprašujete za ta trenutek?* ali *Ali sprašujete za ponedeljek o vremenu?*).
- END
Izjave, s katerimi uporabnik konča dialog (npr. *Ne, nimam več vprašanj.* ali *Hvala, to bi bilo vse.*).

- QUERY-YN
Vprašanja, na katera je mogoče odgovoriti z da ali ne in ne sodijo v kategorijo CHECK ali ALIGN potez (npr. *Ali bo deževalo?* ali *Potem me pa zanima, če je burja v Novi Gorici.*).
- QUERY-WR
Vprašanja po določenih informacijah ali pojasnilih. V to kategorijo sodijo tudi vsa vprašanja, ki ne spadajo v nobeno od ostalih kategorij (npr. *Mi lahko poveste, kakšno bo v naslednjih dneh v hribih vreme?* ali *Kateri kraj oziroma katera pokrajina vas zanima?*).
- QUERY-WI
Izjave, s katerimi uporabnik prosi sistem, naj mu našteje vse vrste podatkov za določen kraj in določen časovni trenutek, do katerih trenutno dostopa (npr. *Zanima me, kakšne informacije imate o razmerah na smučišču Vogel.*).
- QUERY-WL
Izjave, s katerimi uporabnik prosi sistem, naj mu našteje vse pokrajine in kraje v nekem območju, za katere lahko ponudi določeno vrsto podatka ob določenem časovnem trenutku (npr. *Zanima me, za katere kraje v Sloveniji imate podatke o višini snega.*).
- QUERY-WT
Izjave, s katerimi uporabnik prosi sistem, naj mu našteje vse časovne trenutke, za katere lahko ponudi določeno vrsto podatka za določeno pokrajino ali kraj (npr. *Zanima me, za koliko naprej lahko napoveste vreme.*).

V skladu s kategorijami začetnih konverzacijskih potez smo udeležili naslednje kategorije odzivnih potez:

- ACKNOWLEDGE
Odzivi, s katerimi uporabnik ali sistem minimalno pokažeta, da sta potezo, na katero se odzivata, slišala in razumela (npr. *A ne?* ali *Hvala.*).
- REPLY-HELP
Odzivi sistema na HELP potezo (npr. *Sedaj me lahko vprašate še kaj v zvezi z mojim odgovorom ali pa mi zastavite novo vprašanje, če ga imate.* ali *Odgovarjati znam na vprašanja o vremenu, vetru, temperaturi in zračnem tlaku za različne kraje v Sloveniji in Evropi, o času vzhoda in zahoda sonca, o snežnih razmerah, o splošni vremenski napovedi, o biovremenski napovedi, o napovedi za letalstvo ter hidrološki napovedi za Slovenijo, o napovedi za slovensko Primorje ... in mogoče še o čem.*).
- REPLY-TIMEOUT
Odzivi sistema na TIMEOUT potezo (npr. *Ali imate še kakšno vprašanje?* ali *Niste mi povedali vseh potrebnih podatkov ali pa jih nisem razumel. Če vas zanima kaj drugega, vprašajte, sicer pa prosim odgovorite na naslednje vprašanje: Ali sprašujete za jutri čez dan?*).



Slika 2.1: Konverzacijske igre, udejanjene v modulu za vodenje dialoga v sistemu za podajanje informacij o vremenu.

- **REPLY-Y**
Odzivi z da na vprašanja, na katera je mogoče odgovoriti z da ali ne (QUERY-YN, CHECK, ALIGN), ki pomenijo potrditev (npr. *Da.* ali *Ja, za ta trenutek ... če piha ali ne piha.*).
- **REPLY-N**
Odzivi z ne na vprašanja, na katera je mogoče odgovoriti z da ali ne (QUERY-YN, CHECK), ki pomenijo zanikanje (npr. *Ne.*).
- **REPLY-MOD**
Odzivi s popravkom na vprašanja, na katera je mogoče odgovoriti z da ali ne

(QUERY-YN, CHECK, ALIGN) (npr. *Ne, sprašujem za Kope, Pohorje. ali Ne, za prejšnji teden.*).

- REPLY-WR

Odzivi na QUERY-WR poteze (npr. *V Sloveniji jutri čez dan – najvišje dnevne temperature bodo od 2 do 5 stopinj Celzija. Imate še kakšno vprašanje? ali Kranjska Gora.*).

- REPLY-WI

Odzivi sistema na QUERY-WI poteze (npr. *Za ta trenutek na Voglu vam lahko ponudim podatke o vremenu, temperaturi zraka, višini novozapadlega snega, skupni višini snega in hitrosti vetra. Ali vas kakšna od teh informacij zanima?*).

- REPLY-WL

Odzivi sistema na QUERY-WL poteze (npr. *Podatke o skupni višini snega za Slovenijo za ta trenutek imam za Vojsko, za Rateče, za Vernerico, za Krvavec, za Kanin, za Kredarico, za Predel in za Vogel. Torej, ali vas sedaj zanima kakšen od teh krajev?*).

- REPLY-WT

Odzivi sistema na QUERY-WT poteze (npr. *Podatke o temperaturi zraka za Slovenijo imam za danes čez dan, za jutri zjutraj in za jutri čez dan. Želite, da vam podam informacijo za kakšnega od naštetih časovnih trenutkov?*).

Na osnovi naštetih konverzacijskih potez smo definirali 15 konverzacijskih iger, za vsako od 14 začetnih potez in potezo pripravljenosti po eno, namreč: GREETING GAME, INDECIPHERABLE GAME, PARDON GAME, HELP GAME, TIMEOUT GAME, INTERRUPTION GAME, ALIGN GAME, CHECK GAME, END GAME, QUERY-YN GAME, QUERY-WR GAME, QUERY-WI GAME, QUERY-WL GAME, QUERY-WT GAME in READY GAME. Sestavljene so iz konverzacijskih potez, udejanjene pa kot *rekurzivne mreže prehodov* (slika 2.1), tj. kot diagrami oz. načrti, sestavljeni iz prehodov, ki predstavljajo konverzacijske poteze, te pa ustrezajo dejanjem. Rekurzivne mreže prehodov so bolj kompleksne od končnih avtomatov, saj lahko eno mrežo poljubno gnezdimo v drugi, če cilj konverzacijske igre, ki je predstavljena z notranjo mrežo, služi dosegu cilja igre, ki je predstavljena z zunanjo mrežo.

Lastnost, ki izrazito vpliva na uporabnost in zato tudi na uspešnost sistemov za dialog, je sposobnost obvladovanja raznolikih konverzacijskih strategij. Ta vsebujejo osnovna ravnanja, kot so preverjanja in potrditve, prevzemanje pobude, nudenje ustrezne pomoči uporabniku, ponovitev zadnje izjave, prekinitev dialoga in ponovna vzpostavitev konteksta. Predstavljen modul za vodenje dialoga, ki bo del sistema za podajanje informacij o vremenu [Žibert-2003], podpira vse omenjene konverzacijske strategije, te pa so udejanjene kot konverzacijske igre. Nepogrešljive pri vzpostavljanju konteksta so konverzacijske igre CHECK GAME, ki se nanašajo na zgodovino dialoga. V teh igrah sistem uporabnika prosi, naj potrdi predpostavke, do katerih je prišel na podlagi prejšnjih konverzacijskih dejanj.

Vhod v modul za vodenje dialoga je pomenska predstavitev uporabnikove izjave, ki je sestavljena iz imena konverzacijske poteze in elementov ustreznih predalčkov. Tipi

predalčkov so zelo odvisni od domene oz. sestave podatkovne zbirke, do katerih dostopamo. Omenili smo že, da smo v primeru domene vremenskih podatkov definirali tri tipe predalčkov, *podatek o kraju*, *podatek o času* in *vrsto informacije*, pri čemer je množica predalčkov, ki jih sistem mora zapolniti, preden se lahko odzove, odvisna od pomena uporabnikove izjave oz. tipa konverzacijske poteze. Pri izjavah, ki ustrezajo potezam QUERY-YN in QUERY-WR, je treba zapolniti vse tri predalčke, pri izjavah, ki ustrezajo potezam QUERY-WI, le predalčka *podatek o kraju* in *podatek o času*, pri QUERY-WL le predalčka *podatek o času* in *vrsta informacije*, če se želimo omejiti na določeno območje, pa tudi *podatek o kraju*, pri QUERY-WT pa le *podatek o kraju* in *vrsta informacije*. Pri potezi ALIGN, razen zapolnitve vsaj enega predalčka, ni omejitev, saj se uporabnik sam odloča, kaj želi preveriti. Ko pa se uporabnik s potezo REPLY-WR odziva na vprašanje QUERY-WR, ki mu ga je zastavil sistem, so število in tipi potrebnih predalčkov odvisni od vprašanja. Vse ostale možne poteze uporabnika ne zahtevajo zapolnitve nobenega predalčka.

Primer dialoga med uporabnikom in sistemom je v tabeli 2.4 podan v naravnem jeziku, spodaj pa je ta dialog preveden v pomenske predstavitve izjav, ki so bile vhod v modul za vodenje dialoga, s katerim je čarovnik komuniciral z uporabo posebnega vmesnika.

GREETING GAME

S1: (GREET) *Dober dan. Pogovarjate se s sistemom za podajanje informacij o vremenu. Kakšno je vaše vprašanje?*

QUERY-WR GAME

U1: QUERY-WR

kraj: Slovenija

čas: jutri podnevi

podatek: vreme

S2: (REPLY-WR) *V Sloveniji jutri čez dan – padavine bodo ponehale, delno razjasnitve. Želite vedeti še kaj?*

READY GAME

U2: READY

QUERY-WR GAME

U2: QUERY-WR

kraj: gore

čas: –

podatek: vreme

CHECK GAME

S3: (CHECK) *Ali sprašujete za jutri čez dan?*

HELP GAME

U3: HELP

S4: (REPLY-HELP) *Niste mi povedali vseh potrebnih podatkov ali pa jih nisem razumel. Če vas zanima kaj drugega, vprašajte, sicer pa prosim odgovorite na naslednje vprašanje. Ali sprašujete za jutri čez dan?*

U4: REPLY-MOD

čas: trenutno

S5: (REPLY-WR) *O vremenu v gorah ta trenutek žal nimam podatkov. Imam pa podatke o vremenu ta trenutek na Kredarici, na Voglu, na Kaninu, na Krvavcu in v Predelu. Potem lahko izbirate med podatki o vremenu v gorah za danes čez dan in jutri čez dan. Izbirate lahko tudi med podatki o vremenski napovedi v gorah ta trenutek.*

QUERY-WR GAME

U5: QUERY-WR
kraj: Vogel
čas: –
podatek: –

CHECK GAME

S6: (CHECK) *Ali sprašujete za ta trenutek o vremenu?*

U6: REPLY-Y

S7: (REPLY-WR) *Na Voglu ta trenutek – megla. Vam lahko še kako pomagam?*

READY GAME

U7: READY

QUERY-WR GAME

U7: QUERY-WR
kraj: Vogel
čas: trenutno
podatek: temperatura, skupna višina snega

S8: (REPLY-WR) *Na Voglu ta trenutek – skupna višina snega je 180 centimetrov, temperatura zraka je -7 stopinj Celzija. Imate še kakšno vprašanje?*

READY GAME

U8: READY

QUERY-WI GAME

U8: QUERY-WI
kraj: Ljubljana
čas: trenutno

S9: (REPLY-WI) *Za ta trenutek v Ljubljani vam lahko ponudim 9 vrst informacij kot na primer podatek o vremenu, vlažnosti zraka, temperaturi zraka, hitrosti vetra, smeri vetra, vrednosti zračnega tlaka, indeksu požarne ogroženosti, vrsti požarne ogroženosti in tako naprej. Ali vas kakšna od teh informacij zanima?*

END GAME

U9: END

Greeting game

S10: (GREET) *Hvala lepa za sodelovanje. Na svidenje.*

Ta primer dialoga ponazarja udejanjenje številnih konverzacijskih strategij: dve igri GREETING GAME, ki se začneta na pobudo izjav S1 in S10; štiri igre QUERY-WR GAME, ki se začnejo na pobudo izjav U1, U2, U5 in U7; dve igri CHECK GAME, ki se začneta na pobudo izjav S3 in S6; HELP GAME se začne na pobudo izjave U3; QUERY-WI GAME se začne na pobudo izjave U8; END GAME se začne na pobudo izjave U9; tri igre READY GAME pa uvajajo začetki izjav U2, U7 in U8 ter dialog pripravljajo

na začetek nove igre. Odzivna konverzacijska poteza REPLY-WR, pripisana izjavi S5, kaže, kako sistem uporabnika usmerja k izbiri relevantnih, dosegljivih podatkov, ko ne najde točnega odgovora na uporabnikovo vprašanje.

Zue in sodelavci [Zue-2000] trdijo, da je eden od odločilnih vidikov uporabnosti sistemov za dialog sposobnost uporabnika informirati o obsegu znanja, ki ga sistem ima. Uporabnik preprosto mora imeti možnost vprašati in celo biti usmerjen k izbiri relevantnih, dosegljivih podatkov, ko sistem ni sposoben odgovoriti na njegovo vprašanje. Le tako bo namreč sposoben zgraditi dober pojmovni model sistema. V našem modulu za vodenje dialoga to potrebo delno zadovoljimo z udejanjenjem konverzacijskih iger QUERY-WI GAME, QUERY-WL GAME in QUERY-WT GAME, ki poleg tega dodatno strukturirajo dialog. Po drugi strani pa usmerjanje uporabnika k izbiri relevantnih, dosegljivih podatkov v veliki meri podpira tudi predstavitev znanja, predstavljena v razdelku 2.5.

2.5 Sodelujoča podatkovna zbirka

Podatkovni model sodelujoče podatkovne zbirke, uporabljene v drugem eksperimentu Čarovnik iz Oza, temelji na *intuicionistični modalni logiki* [Huth-2004]. Modalna logika poleg klasičnih izjavnih veznikov uporablja *modalne operatorje* (enomestne izjavne veznike), s pomočjo katerih je sposobna izraziti eno ali več oblik resnice, denimo *biti nujno res*, *vedeti, da je res*, *verjeti, da je res* in *biti res v prihodnosti*. Idejo uporabljati modalno logiko za sklepanje o znanju je uvedel Jaakko Hintikka [Hintikka-1975].

Najbolj znana modalna operatorja sta \Box in \Diamond , ki ju v *osnovni modalni logiki* beremo *škatla* in *diamant*. Ko modalno logiko uporabimo za izražanje različnih oblik resnic, omenjena modalna operatorja imenujemo dani resnici primerno, npr. *nujno* in *mogoče*, ali v primeru izražanja znanja agenta, *agent ve* in *agent ne izključuje veljavnosti*.

Modeli $\mathcal{M} = (W, R, V)$ osnovne modalne logike so določeni z:

1. množico W , katere elemente imenujemo *svetovi*,
2. relacijo R na množici W , ki jo imenujemo *relacija dosegljivosti*, wRv pa beremo kot *svet v je dosegljiv iz sveta w* ,
3. funkcijo V , ki jo imenujemo *funkcija pripisovanja veljavnosti* in slika svetove v množice izjavnih spremenljivk, pri čemer $p \in V(w)$ pomeni, da ima izjavna spremenljivka p v svetu w resnično vrednost.

V čast Saula Kripkeja, ki je v 50. in 60. letih prejšnjega stoletja močno zaznamoval modalno logiko, take modele ponavadi imenujemo *Kripkejevi modeli* ali *Kripkejeve semantike možnih svetov*. Veljavnost izjav v svetovih Kripkejevih modelov $\mathcal{M} = (W, R, V)$ opišemo z relacijo \Vdash_V (definicija 2.1).

Definicija 2.1. *Naj bo $\mathcal{M} = (W, R, V)$ model osnovne modalne logike, $w \in W$, \top in \perp običajni konstanti resnica in laž, p izjavna spremenljivka, A in B pa izjavi. Relacija*

\Vdash_V ($w \Vdash_V A$ beremo kot v svetu $w \in W$ velja A) je najmanjša relacija med svetovi in izjavami, ki izpolnjuje naslednje pogoje:

1. $w \Vdash_V \top$
2. $w \not\Vdash_V \perp$
3. $w \Vdash_V p$ čee $p \in V(w)$
4. $w \Vdash_V \neg A$ čee $w \not\Vdash_V A$
5. $w \Vdash_V A \wedge B$ čee $w \Vdash_V A$ in $w \Vdash_V B$
6. $w \Vdash_V A \vee B$ čee $w \Vdash_V A$ ali $w \Vdash_V B$
7. $w \Vdash_V A \Rightarrow B$ čee iz $w \Vdash_V A$ sledi $w \Vdash_V B$
8. $w \Vdash_V A \Leftrightarrow B$ čee $w \Vdash_V A$ natanko tedaj, ko $w \Vdash_V B$
9. $w \Vdash_V \Box A$ čee $\forall v \in W: (wRv \text{ sledi } v \Vdash_V A)$
10. $w \Vdash_V \Diamond A$ čee $\exists v \in W: (wRv \text{ in } v \Vdash_V A)$

Prva dva pogoja pravita, da \top vedno velja, \perp pa nikoli. Iz naslednjega pogoja sledi, da je $V(w)$ množica vseh izjavnih spremenljivk, ki so v svetu w veljavne. Pogoji, ki se nanašajo na nemodalne izjavne veznike (\neg , \wedge , \vee , \Rightarrow in \Leftrightarrow), zrcalijo klasično razumevanje teh veznikov v danem svetu. Neobičajna sta le pogoja za \Box in \Diamond . Prvi od njiju pravi, da je izjava $\Box A$ v svetu w resnična, če je izjava A resnična v vsakem svetu, ki je z relacijo R dosegljiv iz sveta w . Drugi pa pravi, da je izjava $\Diamond A$ v svetu w resnična, če obstaja svet, z relacijo R dosegljiv iz sveta w , kjer je izjava A resnična.

Dana izjava je lahko v nekem svetu resnična, v drugem pa neresnična. Resničnost izjave, ki vsebuje samo klasične logične veznike (\wedge , \vee , \neg , \Rightarrow in \Leftrightarrow), je v klasični modalni logiki določena lokalno in neodvisno od ostalih svetov. Ko pa izjava vsebuje katerega izmed modalnih operatorjev (npr. \Box ali \Diamond), je njena resničnost v posameznem svetu odločilno odvisna od ostalih možnih svetov. Pravimo, da je izjava A osnovne modalne logike *veljavna*, če je veljavna v vsakem svetu vsakega modela, in izjava A je *veljavna v modelu* $\mathcal{M} = (W, R, V)$, če je veljavna v vsakem svetu $w \in W$.

Množice veljavnih izjav, pripadajočih različnim interpretacijam modalnega operatorja \Box , se med seboj razlikujejo [Hajdinjak-2004a], saj formalizirajo različne oblike resnice. Izjave, ki so veljavne v osnovni modalni logiki, so sicer veljavne tudi v vseh drugih modalnih logikah, ki izvirajo iz različnih interpretacij modalnega operatorja \Box . Na ravni Kripkejevih modelov pa vsaki interpretaciji pripada ustrezna interpretacija relacije dosegljivosti R , ki je v nekaterih primerih po potrebi lahko npr. tudi reflektivna ali tranzitivna.

Klasično logiko brez Aristotelovega zakona o izključenosti tretji možnosti $A \vee \neg A$, a z zakonom o nasprotovanju $\neg A \wedge A \Rightarrow B$, imenujemo *intuicionistična logika*, obsega pa principe logičnega sklepanja, ki jih je leta 1907 začel uporabljati L. E. J. Brouwer, ko je razvijal svojo intuicionistično matematiko [Troelstra-www]. Principi, ki jih je začel uporabljati Brouwer, so podvrženi tudi ruski rekurzivni analizi in konstruktivni analizi E. Bishopa in njegovih naslednikov, zato lahko na intuicionistično logiko gledamo kot na temelje konstruktivne matematike.

V intuicionistični logiki lahko z gotovostjo trdimo, da objekt z določeno lastnostjo obstaja šele, ko poznamo učinkovito metodo konstrukcije oz. iskanja takega objekta.

Znamenita izjava še bolj znamenitega intuicionista Arenda Heytinga se glasi, da razložiti formulo ne pomeni podati pogoje resničnosti, ampak pogoje dokazljivosti. To izraža intuicionistična razlaga izjavnih veznikov \neg in \vee , ki se od razlage klasične logike razlikuje. Izjavi A in $\neg\neg A$ zaradi izključitve omenjenega Aristotelovega zakona nista ekvivalentni, interpretacija izjave $A \vee B$ pa je veliko močnejša. Nasprotujoča si podatka A in $\neg A$ lahko v intuicionistični modalni logiki predstavimo kot $\diamond A$ in $\diamond\neg A$, kar pomeni, da dopuščamo obe možnosti.

Modele intuicionistične modalne logike, kjer \square in \diamond beremo kot *nujno* in *mogoče*, dobimo, tako da v $\mathcal{M} = (W, R, V)$ dodamo relacijo \leq delne urejenosti (refleksivna, antisimetrična in tranzitivna) svetov, ki je s pogojem

$$(wRv \text{ in } v \leq v') \implies \exists w' \in W : (w'Rv' \text{ in } w \leq w')$$

povezana z obstoječo relacijo R , za funkcijo pripisovanja veljavnosti V pa zahtevamo lastnost *monotonosti*, tj. iz $w \leq v$ mora slediti $V(w) \subseteq V(v)$. Izjavne spremenljivke intuicionistične modalne logike lahko pojmujeemo kot *monotone resnice* – ko je v nekem svetu izjavna spremenljivka resnična, je resnična tudi v vseh, iz tega sveta z relacijo \leq dosegljivih svetovih. Lema o monotonosti pa pravi, da so tudi ostale izjave monotone resnice.

Dialog mIPC Kripkejeve modele (definicija 2.2), na katerih temelji podatkovni model sodelujoče podatkovne zbirke, uporabljene v drugem sistemu Čarovnik iz Oza, dobimo tako, da zbirko vremenskih podatkov pretvorimo v množico svetov $W = W1 \cup W2$. Prvo podmnožico svetov $W1$ dobimo tako, da vsaki izjavni spremenljivki p_{ij} priredimo svet $W1_{ij} \in W1$, v katerem je p_{ij} , ki jo imenujemo *vodilna atomarna izjava*, veljavna. Drugo podmnožico svetov $W2$ dobimo, tako da v svetove iz $W1$ dodamo veljavnost vsaj ene dodatne izjavne spremenljivke, tj. *omejitev*, pri čemer se noben par *atomarnih izjav* (vodilna atomarna izjava in omejitve) ne nanaša na isti predalček. Te atomarne izjave ne obravnavamo tako kot ostale izjave, saj kljub temu da natanko določajo zakone danega sveta, ne vsebujejo informacije o dosegljivosti oz. relevantnosti posameznih delčkov informacij. Atomarne izjave se navezujejo na sestavne dele uporabnikovega vprašanja, ostale izjave, ki so sestavljene iz

- izjavnih spremenljivk p_{ij} ,
- konstant \perp in \top ,
- klasičnih izjavnih veznikov \wedge, \vee, \neg in \implies ter
- modalnih operatorjev *spodnji diamant* ∇ in *zgornji diamant* Δ ,

pa dosegljivost in relevantnost vremenskih informacij opisujejo tako, da je sistem sposoben uporabnika usmerjati k izbiri informacij, ki se glede na vprašanje zdijo smiselne. Vpeljane modalne operatorje interpretiramo takole:

- ∇A pomeni, da je A dosegljivo in relevantno, vendar manj specifično;
- ΔA pomeni, da je A dosegljivo in relevantno, vendar bolj specifično.

Na množici svetov W definiramo reflektivno in tranzitivno relacijo *dosegljivosti* R , kjer

- wRv pomeni, da je svet v bolj specifičen oz. *dosegljiv* iz sveta w , tj. množici atomarnih izjav svetov w in v se razlikujeta le v vodilni atomarni izjavi, pri čemer je vodilna izjava sveta v bolj specifična od vodilne atomarne izjave sveta w ,

in relacijo delne urejenosti (reflektivna, antisimetrična, tranzitivna) \leq , kjer

- $w \leq v$ pomeni, da je množica atomarnih izjav sveta w podmnožica množice atomarnih izjav sveta v .

Relacija R temelji na vnaprej definirani reflektivni in tranzitivni relaciji *biti manj specifičen*, ki povezuje manj specifične oz. splošnejše delčke informacij, navezujoče se na isti predalček, z bolj specifičnimi. Usmerjene Hassejeve diagrame teh relacij, kjer vozlišča ustrezajo posameznim delčkom vremenskih informacij, na katerih je temeljil drugi sistem Čarovnik iz Oza, usmerjene povezave pa vzpostavljenim relacijam, prikazujejo slike 2.2, 2.3, 2.4 in 2.5. Relacija R tedaj povezuje svetova, katerih množici atomarnih izjav se razlikujeta le v vodilni atomarni izjavi, ti pa sta v relaciji *biti manj specifičen*. Kripkejeve modele, ki jim taka predstavitev znanja zadošča, smo poimenovali *dialog mIPC Kripkejevi modeli*.

Definicija 2.2. Dialog mIPC Kripkejevi modeli so strukture $\mathcal{M} = (W, \leq, R, \Vdash)$, kjer je (W, \leq) neprazna, delno urejena množica svetov, R je reflektivna in tranzitivna relacija na W , ki za $w, w', v, v' \in W$ zadošča

- 0**. iz wRv in $v \leq v'$ sledi $\exists w' \in W : (w'Rv' \text{ in } w \leq w')$,
- iz $w \leq v$ in vRv' sledi $\exists w' \in W : (wRw' \text{ in } w' \leq v')$,
- iz $w \leq v$ in wRv' sledi $\exists w' \in W : (vRw' \text{ in } v' \leq w')$,
- iz $v \leq w$ in $v'Rw$ sledi $\exists w' \in W : (w'Rv \text{ in } w' \leq v')$,

\Vdash pa je dvomestna relacija med svetovi in izjavami ($w \Vdash A$ beremo kot izjava A je v svetu w veljavna), da za poljubni izjavi A in B , izjavno spremenljivko p , ter svetova w in v velja:

- 1**. $w \Vdash \top$
- 2**. $w \not\Vdash \perp$
- 3**. $w \Vdash p$ in $w \leq v$ sledi $v \Vdash p$ (MONOTONOST IZJAVNIH SPREMENLJIVK)
- 4**. $w \Vdash \neg A$ čee $\forall v \in W : (iz w \leq v \text{ sledi } v \not\Vdash A)$
- 5**. $w \Vdash A \wedge B$ čee $w \Vdash A$ in $w \Vdash B$
- 6**. $w \Vdash A \vee B$ čee $w \Vdash A$ ali $w \Vdash B$
- 7**. $w \Vdash A \Rightarrow B$ čee $\forall v \in W : (w \leq v \Rightarrow (iz v \Vdash A \text{ sledi } v \Vdash B))$
- 8**. $w \Vdash \nabla A$ čee $\exists v \in W : (vRw \text{ in } v \Vdash A)$
- 9**. $w \Vdash \Delta A$ čee $\exists v \in W : (wRv \text{ in } v \Vdash A)$

V magistrskem delu [Hajdinjak-2004a] smo dokazali, da v dialog mIPC Kripkejevih modelih velja lema o monotonosti. Zaradi tega lahko o svetovih Kripkejevih modelov razmišljamo kot o možnih stopnjah znanja; svet, ki je v delno urejeni množici (W, \leq) uvrščen višje, predstavlja razširitev našega znanja. Ko se pomikamo po delni urejenosti navzgor, se naše znanje kopiči – ničesar ne pozabimo, vemo kvečjemu več.

Izrek 2.3. Dialog mIPC Kripkejevi modeli zadoščajo vsem pravilom sklepanja intuicionističnega izjavnega računa. Za svetova w in v , nekončni izjavi A in B ter kontekst Γ v dialog mIPC Kripkejevih modelih veljajo tudi naslednja pravila sklepanja:

B1. Spodnji diamant

$$\frac{\Gamma \vdash A}{\Gamma \vdash \nabla A} \quad \frac{\Gamma \vdash \nabla \nabla A}{\Gamma \vdash \nabla A}$$

B2. Zgornji diamant

$$\frac{\Gamma \vdash A}{\Gamma \vdash \Delta A} \quad \frac{\Gamma \vdash \Delta \Delta A}{\Gamma \vdash \Delta A}$$

B3. Konjunkcija in spodnji/zgornji diamant

$$\frac{\Gamma \vdash \nabla(A \wedge B)}{\Gamma \vdash \nabla A \wedge \nabla B} \quad \frac{\Gamma \vdash \Delta(A \wedge B)}{\Gamma \vdash \Delta A \wedge \Delta B}$$

B4. Disjunkcija in spodnji/zgornji diamant

$$\frac{\Gamma \vdash \nabla A \vee \nabla B}{\Gamma \vdash \nabla(A \vee B)} \quad \frac{\Gamma \vdash \Delta A \vee \Delta B}{\Gamma \vdash \Delta(A \vee B)}$$

B5. Konjunkcija, implikacija in spodnji/zgornji diamant

$$\frac{\Gamma \vdash \nabla(A \wedge (A \Rightarrow B))}{\Gamma \vdash \nabla B} \quad \frac{\Gamma \vdash \Delta(A \wedge (A \Rightarrow B))}{\Gamma \vdash \Delta B}$$

Pravila sklepanja z dvojno črto beremo kot veljavne sklepe v obe smeri. Na primer, drugi del pravila B1 pravi, da iz $\Gamma \vdash \nabla \nabla A$ sledi $\Gamma \vdash \nabla A$ in obratno. Tako dobljene sklepe, sestavljene le iz ene izjavne spremenljivke in enomestnih izjavnih veznikov, imenujemo *končne izjave* in jih iz nadaljnjih sklepov izločimo.

Končne izjave dialog mIPC Kripkejevih modelov predstavljajo podobno slabost kot negativna dejstva deduktivnega podatkovnega modela Datalog, izpeljana iz predpostavke zaprtega sveta. Tudi v Datalogu je negativna dejstva sicer mogoče izpeljati, ni pa dovoljena njihova uporaba pri izpeljavi novih dejstev.

Primer 2.4. Oglejmo si primere svetov v predstavitvi vremenskih podatkov. Predpostavimo, da vir znanja sistema vsebuje podatke o trenutni temperaturi v Ljubljani, o temperaturi v Ljubljani danes zvečer ter o pričakovani jutrišnji dnevni temperaturi v Sloveniji. Naj bo Ljubljana svet z vodilno atomarno izjavo **Ljubljana**, Slovenija pa svet z vodilno atomarno izjavo **Slovenija**, oba brez dodatnih omejitev. Ker je izjavna spremenljivka Slovenija v relaciji biti manj specifičen z izjavno spremenljivko Ljubljana (slika 2.2), svet Ljubljana vsebuje naslednje izjave:

Ljubljana

temperatura \Rightarrow trenutno (15)

trenutno \Rightarrow temperatura (15)

temperatura \Rightarrow danes zvečer (10)

danes zvečer \Rightarrow temperatura (10)

$\nabla^{\text{Slovenija}}$ (temperatura \Rightarrow jutri podnevi) (od 13 do 17)

$\nabla^{\text{Slovenija}}$ (jutri podnevi \Rightarrow temperatura) (od 13 do 17)

⋮

Svet Slovenija pa vsebuje izjave:

Slovenija

temperatura \Rightarrow jutri podnevi (od 13 do 17)

jutri podnevi \Rightarrow temperatura (od 13 do 17)

$\triangle^{\text{Ljubljana}}$ (temperatura \Rightarrow trenutno) (15)

$\triangle^{\text{Ljubljana}}$ (trenutno \Rightarrow temperatura) (15)

$\triangle^{\text{Ljubljana}}$ (temperatura \Rightarrow danes zvečer) (10)

$\triangle^{\text{Ljubljana}}$ (danes zvečer \Rightarrow temperatura) (10)

⋮

Pri tem smo modalnima operatorjema ∇ in \triangle dodali indeks atomarnih izjav ($\nabla^{\text{Slovenija}}$ oz. $\triangle^{\text{Ljubljana}}$), ki pove, na kateri relevanten svet se izjava nanaša, dejansko informacijo (npr. opis temperature) pa smo zapisali v oklepajih na koncu izjave.

Primer 2.5. V mislih imejmo primer 2.4 in predpostavimo, da uporabnika zanima jutrišnja temperatura v Ljubljani. Sistem (ko nima zahtevane informacije) uporabnika usmerja k izbiri dosegljivih, relevantnih podatkov, tako da ponudi relevantne kraje ali relevantne časovne trenutke, za katere je sposoben podati zahtevano informacijo. Relevantne kraje izbere tako, da se iz sveta Ljubljana premakne v svet, katerega dodatna omejitev je **temperatura**. Svetova Ljubljana in Slovenija iz primera 2.4 tako preideta v svetova Ljubljana-temperatura in Slovenija-temperatura, ki ju dobimo tako, da množici prvotnih izjav dopolnimo z omejitvijo **temperatura** in uporabimo pravila sklepanja iz izreka 2.3. Pri tem je Ljubljana \leq Ljubljana-temperatura in Slovenija \leq Slovenija-temperatura. V svetu Ljubljana-temperatura so veljavne naslednje izjave:

Ljubljana, temperatura

temperatura \Rightarrow trenutno (15)

trenutno \Rightarrow temperatura (15)

temperatura \Rightarrow danes zvečer (10)

danes zvečer \Rightarrow temperatura (10)

$\nabla^{\text{Slovenija-temperatura}}$ (temperatura \Rightarrow jutri podnevi) (od 13 do 17)

$\nabla^{\text{Slovenija-temperatura}}$ (jutri podnevi \Rightarrow temperatura) (od 13 do 17)

⋮

Naslednje končne izjave dobimo s pomočjo pravil sklepanja iz izreka 2.3.

trenutno (15)

danes zvečer (10)

$\nabla^{\text{Slovenija-temperatura}}$ jutri podnevi (od 13 do 17)

⋮

Dane izjave nazorno kažejo, da ima sistem podatke o trenutni temperaturi v Ljubljani in o temperaturi v Ljubljani danes zvečer, dosegljiv pa je tudi podatek o jutrišnji temperaturi v Sloveniji, kar je zagotovo relevantno, saj sta izjavni spremenljivki oz. vodilni atomarni izjavi Slovenija in Ljubljana v relaciji biti manj specifičen (slika 2.2).

*Relevantne časovne trenutke pa sistem najde, tako da se premakne v svet jutri podnevi-Ljubljana, katerega vodilna atomarna izjava je **jutri podnevi**, dodatna omejitev pa Ljubljana. Svetovi jutri podnevi, trenutno in danes zvečer tedaj preidejo v svetove jutri podnevi-Ljubljana, trenutno-Ljubljana in danes zvečer-Ljubljana. Pri tem je jutri podnevi \leq jutri podnevi-Ljubljana, trenutno \leq trenutno-Ljubljana in danes zvečer \leq danes zvečer-Ljubljana, v svetu jutri podnevi-Ljubljana veljajo naslednje izjave:*

jutri podnevi, Ljubljana

temperatura \Rightarrow Slovenija (od 13 do 17)

Slovenija \Rightarrow temperatura (od 13 do 17)

$\Delta^{\text{trenutno-Ljubljana}}$ (temperatura \Rightarrow Ljubljana) (15)

$\Delta^{\text{trenutno-Ljubljana}}$ (Ljubljana \Rightarrow temperatura) (15)

$\nabla^{\text{trenutno-Ljubljana}}$ (temperatura \Rightarrow Ljubljana) (15)

$\nabla^{\text{trenutno-Ljubljana}}$ (Ljubljana \Rightarrow temperatura) (15)

$\Delta^{\text{danes zvečer-Ljubljana}}$ (temperatura \Rightarrow Ljubljana) (10)

$\Delta^{\text{danes zvečer-Ljubljana}}$ (Ljubljana \Rightarrow temperatura) (10)

$\nabla^{\text{danes zvečer-Ljubljana}}$ (temperatura \Rightarrow Ljubljana) (10)

$\nabla^{\text{danes zvečer-Ljubljana}}$ (Ljubljana \Rightarrow temperatura) (10)

⋮

Naslednje končne izjave dobimo s pomočjo pravil sklepanja iz izreka 2.3.

$\Delta^{\text{trenutno-Ljubljana}}$ temperatura (15)

$\nabla^{\text{trenutno-Ljubljana}}$ temperatura (15)

$\Delta^{\text{danes zvečer-Ljubljana}}$ temperatura (10)

$\nabla^{\text{danes zvečer-Ljubljana}}$ temperatura (10)

⋮

Vidimo, da so podatki o trenutni temperaturi v Ljubljani in o temperaturi v Ljubljani danes zvečer relevantni, saj sta izjavni spremenljivki jutri podnevi in trenutno ter izjavni spremenljivki jutri podnevi in danes zvečer v obojesmerni relaciji biti manj specifičen (slika 2.4).

Omenimo še, kdaj svet w vsebuje izjavo $\neg A$ in kako se sistem v tem primeru odziva. Izjava $\neg A$ pomeni, da izjava A ne velja v nobenem svetu v , ki je z relacijo \leq dosegljiv iz sveta w . Taki svetovi $v \geq w$ vsebujejo vse atomarne izjave sveta w in vsaj eno dodatno omejitev. Izjava $\neg A$ zato v svetu w velja, če za nobeno izbiro dodatnih omejitev vir znanja ne vsebuje informacije A . Recimo, da svet Ljubljana z edino atomarno izjavo Ljubljana vsebuje izjavo $\neg vreme$. To pomeni, da sistem nima podatka o vremenu za nobeno izbiro časovnega trenutka, kar sistem ve, še preden se premakne v svet Ljubljana-vreme, da bi uporabniku ponudil podatke o vremenu v dosegljivih časovnih trenutkih. Svet Ljubljana-vreme v tem primeru vsebuje atomarno izjavo $vreme$ in neatomarno izjavo $\neg vreme$ (ekvivalentna $vreme \Rightarrow \perp$), iz česar po enem od aksiomov intuicionističnega izjavnega računa izpeljemo laž \perp .

3 Ogrodje PARADISE

3.1 Uvod

3.2 Predstavitev ogrodja PARADISE

3.3 Analiza ogrodja PARADISE

Opišemo potencialno splošno metodologijo vrednotenja učinkovitosti sistemov za dialog, namreč ogrodje PARADISE (PARAdigm for DIAlogue System Evaluation), ki omogoča izpeljavo ocene učinkovitosti sistema kot uteženo linearno kombinacijo parametrov uspešnosti naloge in cen dialoga. Model učinkovitosti sistema, ki ga zajema ogrodje PARADISE, trdi, da lahko funkcijo učinkovitosti določimo z uporabo multiple linearne regresije (MLR) z zadovoljstvom uporabnikov kot neodvisno spremenljivko.

Osredotočimo se na nekatere pomanjkljivosti, težave in nerešena vprašanja ogrodja PARADISE. Podrobno opišemo vpliv normalizacije na natančnost napovedovanja zadovoljstva uporabnikov, izpostavimo predpostavke, katerih izpolnitev zahteva multipla linearna regresija, poudarimo pomembnost dobre izbire regresijskih parametrov, opozorimo na nepravilnosti pri merjenju zadovoljstva uporabnikov ter predlagamo način zbiranja podatkov, ki zmanjša vpliv razpoznavanja govora in okrepi vplive ostalih parametrov funkcije učinkovitosti.

3.1 Uvod

Z razvojem sistemov za dialog se pojavljajo tudi potrebe po vrednotenju učinkovitosti in medsebojni primerjavi takih sistemov. Težava, ki se pojavi, je ta, da vrednotenja učinkovitosti sistema za dialog ni mogoče omejiti na primerjave z referenčnimi odgovori oz. referenčnimi poteki dialogov [Bates-1991, Polifroni-1992, Price-1992]. Množica sprejemljivih dialogov je namreč lahko zelo velika. Naslednja težava je veliko število potencialnih metrik dialoga. Sistem za dialog lahko npr. vrednotimo glede na njegovo sposobnost pomagati uporabnikom pri doseganju ciljev, glede na njegovo robustnost odkrivanja in premagovanja napak, ki se pojavljajo pri razpoznavanju oz. razumevanju govora, ali glede na skupno kakovost interakcije [Polifroni-1992, Price-1992, Danieli-1995, Sparck-Jones-1996], ki jo omogoča.

Predlogi vrednotenja učinkovitosti sistemov za dialog, ki so se pojavili v zadnjih dveh desetletjih dvajsetega stoletja, se osredotočajo na razvoj različnih metrik dialoga. Predlagani so bili številni *objektivni parametri dialoga* [Price-1992, Danieli-1995, Smith-1997] kot npr.

- število izjav,
- čas dialoga,
- povprečni čas odziva uporabnika,
- povprečni čas odziva sistema,
- delež izjav, sestavljenih iz več kot ene besede,
- povprečna dolžina izjav, sestavljenih iz več kot ene besede,

ki jih lahko določimo brez mnenja človeka, in parametri, ki temeljijo na mnenju človeka, namreč *subjektivni parametri dialoga* [Shriberg-1992, Danieli-1995, Boyce-1996] kot npr.

- delež izjav, s katerimi sistem popravlja napake,
- delež kontekstno primernih izjav sistema,
- hevristično vrednotenje stopnje sodelovanja sistema na podlagi Griceovih maksim [Grice-1975],
- delež pravilnih in delno pravilnih odgovorov,
- delež primernih in neprimernih izjav, s katerimi sistem usmerja uporabnika,
- zadovoljstvo uporabnika [Shriberg-1992].

Izkaže pa se, da lahko kombinacija različnih parametrov pri vrednotenju učinkovitosti sistemov za dialog pripelje do težav. V [Danieli-1995] podana primerjava dveh sistemov za podajanje informacij o železniških povezavah na primer kaže, da lahko en sistem sicer

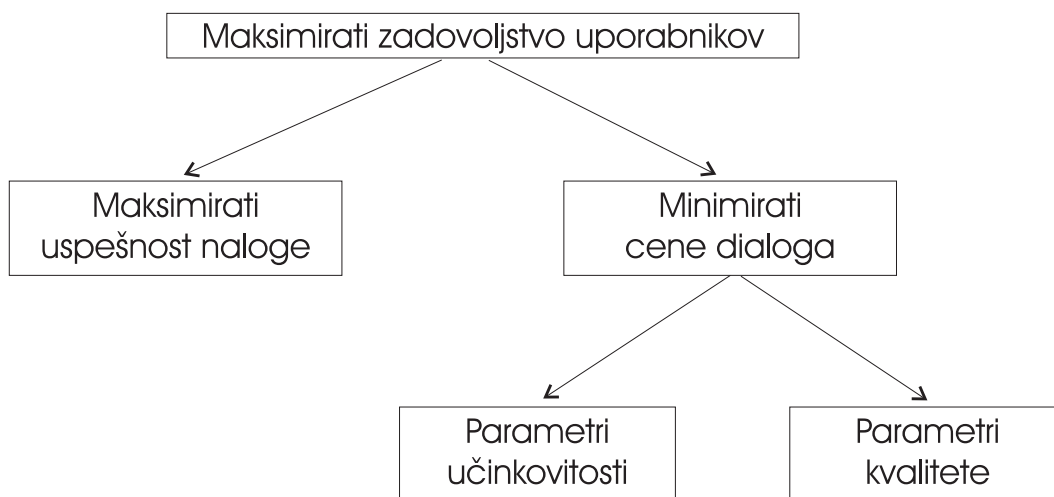
omogoča več uspešno končanih dialogov in povzroča manj neustreznih in popravnih izjav, dialogi z drugim sistemom pa so zato lahko za polovico krajši. V opisanem primeru ni bilo mogoče ugotoviti, kateri od sistemov je bolj uporaben.

Z namenom omogočiti primerjavo sistemov z različnimi domenami, kjer je pomembno vedeti, v kolikšni meri posamezni parametri vplivajo na učinkovitost in kako strategija vodenja dialoga vpliva na zadovoljstvo uporabnikov, je bilo leta 1997 kot potencialna splošna metodologija vrednotenja učinkovitosti sistemov za dialog predlagano ogrodje PARADISE (PARAdigm for DIalogue System Evaluation) [Walker-1997a].

Ko bomo govorili o strukturi dialoga, bomo uporabljali pojma konverzacijskih iger in konverzacijskih potez, ki smo ju definirali v poglavju 2. Tudi če dialoga ne modeliramo s konverzacijsko teorijo iger, lahko o zaključenih delih dialoga govorimo kot o konverzacijskih igrah in o posameznih govornih dejanjih kot o konverzacijskih potezah.

3.2 Predstavitev ogrodja PARADISE

Ogrodje PARADISE omogoča izpeljavo ocene učinkovitosti sistema kot uteženo linearno kombinacijo *parametrov uspešnosti naloge* in *cen dialoga*, zajema pa model učinkovitosti sistema (slika 3.1), ki za osnovni cilj postavlja maksimirati zadovoljstvo uporabnikov, kar doseže z maksimiranjem parametrov uspešnosti naloge in minimiranjem cen dialoga.



Slika 3.1: Model učinkovitosti sistema za dialog, ki ga zajema ogrodje PARADISE.

Zadovoljstvo uporabnikov ponavadi merimo z vprašalniki, v katerih uporabniki podajo stopnjo strinjanja z izjavami o različnih vidikih svoje interakcije s sistemom za dialog. Avtorice ogrodja PARADISE [Walker-1997a] v ta namen uporabljajo vprašalnik, podan v tabeli 3.1. Vprašanja (v naštetem vrstnem redu) sprašujejo po učinku modula za tvorjenje govora, učinku modula za razpoznavanje govora, težavnosti pridobivanja informacij, hitrosti interakcije, izkušnosti uporabnikov, ustreznosti odzivov

sistema, pričakovanem obnašanju sistema in načrtovani rabi sistema v prihodnosti. Večino odgovorov podajo opisno kot *skoraj nikoli*, *redko*, *včasih*, *pogosto* in *skoraj vedno*, nekatere pa le z *da*, *ne* in *mogoče*. Te potem preslikajo v množico naravnih števil od 1 do 5, pri čemer 1 pomeni najmanjšo, 5 pa največjo stopnjo strinjanja. Parameter, ki ocenjuje zadovoljstvo uporabnikov, dobijo kot vsoto vseh ocen in ga poimenujejo **zadovoljstvo uporabnika** (US).

-
-
1. Ali ste sistem brez težav razumeli? (TTS)
 2. Ali vas je sistem razumel? (ASR)
 3. Ali ste brez težav prišli do odgovorov na vaša vprašanja? (TE)
 4. Ali je bila hitrost interakcije s sistemom primerna? (IP)
 5. Ali ste na vsakem koraku dialoga vedeli, kaj morate povedati? (UE)
 6. Ali se je sistem na vaše izjave odzival hitro (brez pojasnilnih vprašanj)? (SR)
 7. Ali se je sistem obnašal tako, kot ste med dialogom od njega pričakovali? (EB)
 8. Glede na vašo trenutno izkušnjo s sistemom, ali mislite, da boste sistem še kdaj poklicali? (FU)
-
-

Tabela 3.1: Vprašalnik za ocenjevanje zadovoljstva uporabnikov, ki ga predlaga ogrodje PARADISE.

Cene dialoga, tj. parametre dialoga, katerih minimizacija ugodno vpliva na zadovoljstvo uporabnikov, razdelimo v dve kategoriji:

↪ *parametri učinkovitosti* dialoga in

↪ *parametri kakovosti* dialoga.

Parametri učinkovitosti dialoga (npr. število izjav, ki jih uporabnik potrebuje, da ureni svojo namero, ali čas dialoga) merijo, kako učinkovito sistem uporabniku pomaga pri doseganju njegove namere. Parametri kakovosti dialoga (npr. kolikokrat mora uporabnik ponoviti svojo izjavo, da ga sistem razume, ali kakšen je čas čakanja na odziv sistema) pa zajemajo ostale vidike, ki lahko na zadovoljstvo uporabnika prav tako močno vplivajo. Ker vnaprej ni jasno, katere cene dialoga bodo najmočnejše vplivale na zadovoljstvo uporabnikov, je pomembno, da v empiričnih raziskavah uporabljamo širok spekter teh parametrov [Walker-1998].

	KLJUČ	kraj		čas		podatek		
ODGOVOR		Krvavec	Rogla	trenutno	vikend	temp.	vreme	viš. snega
kraj	Krvavec	3						
	Rogla	1	1					
čas	trenutno			2	1			
	vikend				2			
podatek	temp.					2		
	vreme						2	
	viš. snega							1
	15	4	1	2	3	2	2	1

Tabela 3.2: Primer kontingenčne tabele.

Uspešnost naloge, ki se lahko nanaša na celoten dialog ali del dialoga, ki predstavlja zaključeno celoto, pomeni stopnjo ujemanja med vsebino zahtev uporabnika in dojemanjem te s strani sistema za dialog. Ogrodje PARADISE uporablja en sam parameter uspešnosti naloge, namreč **Kappa koeficient** [Carletta-1996b]. **Kappa koeficient** (κ) izračunamo z uporabo Cohenove metode [DiEugenio-2004] in kontingenčne tabele, ki podaja ujemanje med vsebino zahtev uporabnika (*ključ*) in dojemanjem te s strani sistema (*odgovor*). Primer kontingenčne tabele prikazuje tabela 3.2.

Vsako ujemanje ključa z odgovorom poveča število v pripadajoči diagonalni celici kontingenčne tabele za 1. Zunajdiagonalne celice ustrezajo neujemanjem, ki znotraj tekoče konverzacijske igre niso bila odpravljena. Začasna neujemanja, ki jih sogovornika znotraj tekoče konverzacijske igre uspešno odpravita, se kažejo le v cenah dialoga. **Kappa koeficient**, ki meri ujemanje opisov različnih označevalcev, izračunamo takole:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

Pri tem verjetnost

$$P(A) = \frac{\sum_{i=1}^n M(i, i)}{T}$$

izraža ujemanje ključa in odgovora v kontingenčni tabeli M z vsoto vseh števil tabele enako T , ki opisuje dialog z n konverzacijskimi igrami,

$$P(E) = \sum_{i=1}^n \left(\frac{t_i}{T}\right)^2$$

pa je verjetnost pričakovanega ujemanja dveh različnih označevalcev, ki dane kategorije tabele izbirata z enako verjetnostjo, pri čemer je t_i vsota števil i -tega stolpca tabele. Če je verjetnost $P(A)$ enaka verjetnosti pričakovanega ujemanja $P(E)$, je $\kappa = 0$, pri popolnem ujemanju, to je pri $P(A) = 1$, pa je $\kappa = 1$.

Če torej želimo sistem za dialog vrednotiti z ogrođjem PARADISE, moramo podatke zbrati v eksperimentu, v katerem bodo uporabniki ocenili svoje zadovoljstvo. Ostale parametre modela (parametri uspešnosti naloge, cene dialoga) pa je treba določiti samodejno ali jih ročno označiti.

Model učinkovitosti sistema, ki ga zajema ogrođje PARADISE, trdi, da lahko funkcijo učinkovitosti sistema določimo z uporabo *multiple linearne regresije* (MLR) z zadovoljstvom uporabnikov kot neodvisno spremenljivko ter parametri uspešnosti naloge,

parametri učinkovitosti dialoga in parametri kakovosti dialoga kot neodvisnimi spremenljivkami:

$$\text{Učinkovitost} = \alpha \mathcal{N}(\kappa) - \sum_{i=1}^n w_i \mathcal{N}(c_i)$$

Pri tem je α utež edinega parametra uspešnosti naloge, namreč Kappa koeficienta κ , w_i so uteži cen dialoga c_i , \mathcal{N} pa je funkcija normalizacije:

$$\mathcal{N}(x) = \frac{x - \bar{x}_0}{\sigma_{x_0}}$$

Z \bar{x}_0 in σ_{x_0} smo označili srednjo vrednost in standardni odklon spremenljivke x_0 v učni množici, pridobljeni v ustreznem eksperimentu. Srednja vrednost s funkcijo normalizacije \mathcal{N} preslikanih parametrov učne množice je 0, varianca in standardni odklon pa 1. Tako se znebimo težav, ki se pojavijo, če primerjamo vrednosti parametrov, ki se raztezajo na različnih intervalih in/ali so njihove vrednosti različno razpršene. Z normalizacijo parametrov κ in c_i dosežemo relevantnost in primerljivost uteži preslikanih parametrov $\mathcal{N}(\kappa)$ in $\mathcal{N}(c_1), \dots, \mathcal{N}(c_n)$.

Rezultat multiple linearne regresije na učni množici parametrov, ki praviloma tvorijo predoločen sistem, je torej množica uteži, ki pomenijo sorazmeren prispevek teh parametrov k učinkovitosti sistema. Funkcija učinkovitosti, ki jo uvaja ogrodje PARADISE, zato omogoča:

- ↪ napovedovanje zadovoljstva uporabnikov,
- ↪ vrednotenje učinkovitosti sistema za dialog, tj. ugotavljanje vpliva posameznih parametrov na zadovoljstvo uporabnikov,
- ↪ izboljšanje sistema za dialog, tj. odpravljanje ali zmanjšanje vpliva parametrov, ki imajo najbolj negativne uteži in povečanje vpliva parametrov, ki imajo najbolj pozitivne uteži,
- ↪ primerjavo različnih sistemov za dialog, tj. primerjavo vplivov posameznih parametrov v pripadajočih funkcijah učinkovitosti, iz katerih lahko razberemo razlike med sistemi,
- ↪ samodejno iskanje problematičnih dialogov, tj. iskanje dialogov, katerih napovedano zadovoljstvo uporabnikov negativno izstopa, ter
- ↪ spreminjanje strategije vodenja dialoga med samo interakcijo, tj. spreminjanje načina sporazumevanja na osnovi napovedanega zadovoljstva uporabnika v že izvedenem delu interakcije.

V zadnjih letih je bilo opravljenih veliko študij učinkovitosti sistemov za dialog, ki so uporabljale ogrodje PARADISE [Walker-1998, Kamm-1998, Kamm-1999, Walker-2000, Litman-2002, Larsen-2003, Möller-2005, Hajdinjak-2005]. Ogrodje PARADISE je postala celo najbolj citirana metoda vrednotenja učinkovitosti sistemov za dialog.

3.3 Analiza ogrodja PARADISE

Omenili smo že, da če želimo sistem za dialog vrednotiti z ogrođjem PARADISE, moramo podatke zbrati v eksperimentu, v katerem bodo uporabniki ocenili svoje zadovoljstvo z obnašanjem danega sistema za dialog. V eksperimentu pridobljene vrednosti odvisne spremenljivke **zadovoljstvo uporabnika** in izbranih neodvisnih spremenljivk (tj. parametrov uspešnosti naloge, parametrov učinkovitosti dialoga in parametrov kakovosti dialoga) tvorijo predoločen sistem linearnih enačb. Nezanke sistema so uteži normaliziranih parametrov modela učinkovitosti. Ogrodje PARADISE trdi tudi, da te uteži (in s tem funkcijo učinkovitosti) dobimo kot rezultat multiple linearne regresije. V eksperimentu pridobljene vrednosti tedaj predstavljajo učno množico modela učinkovitosti sistema za dialog.

V nadaljevanju se bomo osredotočili na nekatere pomanjkljivosti, težave in nerešena vprašanja ogrodja PARADISE [Hajdinjak-2006]. Večina jih izvira ravno iz uporabe multiple linearne regresije.

3.3.1 Vpliv normalizacije na natančnost napovedovanja zadovoljstva uporabnikov

Multipla linearna regresija temelji na metodi najmanjših kvadratov, tj. minimira vsoto kvadratov razlik med v eksperimentu pridobljenimi vrednostmi (tj. učne množice) in napovedanimi vrednostmi zadovoljstva uporabnikov. Za dano vrednost zadovoljstva uporabnika US torej velja

$$\mathcal{N}(US) = \widehat{\mathcal{N}(US)} + \epsilon,$$

kjer je $\mathcal{N}(US)$ normalizirana pridobljena vrednost zadovoljstva uporabnika, $\widehat{\mathcal{N}(US)}$ napovedana normalizirana vrednost zadovoljstva uporabnika, ϵ pa napaka napovedi. Ker je srednja vrednost napake ϵ enaka 0 (razdelek 3.3.2), sta srednji vrednosti odvisne spremenljivke in njene napovedi enaki. Nenormalizirano zadovoljstvo uporabnika US lahko tedaj ocenimo kot

$$US = \widehat{\mathcal{N}(US)}\sigma_{US_0} + \overline{US}_0 + \epsilon\sigma_{US_0} = \widehat{US} + \epsilon\sigma_{US_0},$$

kjer sta \overline{US}_0 in σ_{US_0} srednja vrednost in standardni odklon v eksperimentu pridobljenih vrednosti zadovoljstva uporabnikov. Vidimo, da se napaka ocene normaliziranega zadovoljstva uporabnika $\mathcal{N}(US)$ pri tem poveča za faktor σ_{US_0} .

Kako dobro \widehat{US} napoveduje US , kaže razmerje absolutnih vrednosti njune razlike in pridobljene vrednosti zadovoljstva uporabnika US :

$$q(US, \widehat{US}) = \frac{|US - \widehat{US}|}{|US|}$$

Naslednje razmerje pa kaže, da ocena normalizirane vrednosti zadovoljstva uporabnika

$\widehat{\mathcal{N}}(US)$ ni vedno tako dobra kot ocena nenormalizirane vrednosti \widehat{US} :

$$\frac{q(\mathcal{N}(US), \widehat{\mathcal{N}}(US))}{q(US, \widehat{US})} = \frac{\frac{|\mathcal{N}(US) - \widehat{\mathcal{N}}(US)|}{|\mathcal{N}(US)|}}{\frac{|US - \widehat{US}|}{|US|}} = \frac{\frac{|US - \overline{US}_0 - \widehat{\mathcal{N}}(US)\sigma_{US_0}|}{|US - \overline{US}_0|}}{\frac{|US - \widehat{\mathcal{N}}(US)\sigma_{US_0} - \overline{US}_0|}{|US|}} = \frac{|US|}{|US - \overline{US}_0|}$$

Za $US > \frac{\overline{US}_0}{2}$ namreč velja:

$$\frac{q(\mathcal{N}(US), \widehat{\mathcal{N}}(US))}{q(US, \widehat{US})} = \frac{|US|}{|US - \overline{US}_0|} > 1$$

Iz prikazanega sledi, da je napovedano normalizirano vrednost zadovoljstva uporabnika $\widehat{\mathcal{N}}(US)$ treba transformirati nazaj na začetni interval, saj je ocena nenormalizirane vrednosti zadovoljstva uporabnika \widehat{US} v večini primerov veliko boljša. To naredimo s transformacijo

$$\widehat{US} = \widehat{\mathcal{N}}(US)\sigma_{US_0} + \overline{US}_0,$$

ki je inverzna normalizaciji.

Ne samo da ustrezna literatura [Walker-1997a, Walker-1998, Kamm-1999, Walker-2000, Litman-2002, Larsen-2003] vplivu normalizacije ne posveča pozornosti, ampak tudi ne omenja, da je treba vrednosti zadovoljstva uporabnikov, preden začnemo izpeljavo modela učinkovitosti, normalizirati, če želimo preprečiti prevelike napake ocen [Demmel-1997]. V naših eksperimentih je bilo pogojenostno število problema najmanjših kvadratov več kot sedemkrat večje, ko vrednosti US nismo normalizirali.

Obstaja več načinov merjenja natančnosti MLR modelov [Johnson-2002c]. Najpogosteje se uporablja *koeficient (multiple) determinacije*,

$$R^2 = \frac{\sum_{i=1}^m (\widehat{\mathbf{X}}_i - \overline{\mathbf{X}})^2}{\sum_{i=1}^m (\mathbf{X}_i - \overline{\mathbf{X}})^2},$$

tj. razmerje pojasnjene variance in celotne variance $var(\mathbf{X})$, pri čemer smo z m označili število enačb učne množice. Celotna varianca je vsota pojasnjene variance in nepojasnjene variance:

$$var(\mathbf{X}) = \frac{1}{m} \sum_{i=1}^m (\mathbf{X}_i - \overline{\mathbf{X}})^2 = \frac{1}{m} \sum_{i=1}^m (\widehat{\mathbf{X}}_i - \overline{\mathbf{X}})^2 + \frac{1}{m} \sum_{i=1}^m (\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2$$

Nepojasnjena varianca oz. srednja kvadratna napaka,

$$\overline{\epsilon^2} = \frac{1}{m} \sum_{i=1}^m (\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2,$$

je ravno količina, ki jo multipla linearna regresija minimira. Koeficient determinacije zavzame vrednosti med 0 in 1. Vrednosti, ki so bližje 1, pomenijo večjo natančnost modela, tj. boljšo linearno zvezo med odvisno spremenljivko in neodvisnimi spremenljivkami. Če koeficient determinacije R^2 pomnožimo s faktorjem 100, rezultat imenujemo *odstotek pojasnjene variance*.

Izkaže se, da je v MLR modelu z normaliziranimi spremenljivkami koeficient determinacije R^2 enak varianci napovedanih vrednosti:

$$R^2 = \frac{\sum_{i=1}^m (\widehat{\mathcal{N}(\mathbf{US}_i)} - \overline{\mathcal{N}(\mathbf{US})})^2}{\sum_{i=1}^m (\mathcal{N}(\mathbf{US}_i) - \overline{\mathcal{N}(\mathbf{US})})^2} = \frac{\sum_{i=1}^m \widehat{\mathcal{N}(\mathbf{US}_i)}^2}{m} = \text{var}(\widehat{\mathcal{N}(\mathbf{US})})$$

Pri tem smo z \mathbf{US}_i označili i -to komponento vektorja \mathbf{US} pridobljenih vrednosti zadovoljstva uporabnikov, z $\widehat{\mathcal{N}(\mathbf{US}_i)}$ pa i -to komponento vektorja $\widehat{\mathcal{N}(\mathbf{US})}$ napovedanih normaliziranih vrednosti zadovoljstva uporabnikov. Zadnja enakost velja zato, ker je $\widehat{\mathcal{N}(\mathbf{US})}$, torej srednja vrednost napovedanih normaliziranih vrednosti zadovoljstva uporabnikov, enaka $\overline{\mathcal{N}(\mathbf{US})} = 0$. Zanimiva posledica te ugotovitve je, da so uteži funkcije učinkovitosti po absolutni vrednosti navzgor omejene z 1. Za MLR model

$$\widehat{\mathbf{X}} = \sum_{i=1}^n \alpha_i \mathbf{X}_i$$

namreč velja naslednje:

$$\text{var}(\widehat{\mathbf{X}}) = \text{var}\left(\sum_{i=1}^n \alpha_i \mathbf{X}_i\right) = \sum_{i=1}^n \alpha_i^2 + 2 \sum_{i=1}^n \sum_{j=i+1}^n \alpha_i \alpha_j \text{corr}(\mathbf{X}_i, \mathbf{X}_j),$$

pri čemer je

$$\text{corr}(\mathbf{X}_i, \mathbf{X}_j) = \frac{\frac{1}{m} \sum_{k=1}^m (\mathbf{X}_{ik} - \overline{\mathbf{X}_i})(\mathbf{X}_{jk} - \overline{\mathbf{X}_j})}{\sigma_{\mathbf{X}_i} \sigma_{\mathbf{X}_j}}$$

korelacija oz. korelacijski koeficient spremenljivk \mathbf{X}_i in \mathbf{X}_j . Ker ima dvojna vsota v $\text{var}(\widehat{\mathbf{X}})$ same nenegativne člene, sledi

$$1 \geq R^2 = \text{var}(\widehat{\mathcal{N}(\mathbf{US})}) \geq \alpha^2 + \sum_{i=1}^n w_i^2$$

in zato napovedan rezultat za uteži funkcije učinkovitosti:

$$|\alpha| \leq 1$$

$$|w_i| \leq 1 \text{ za } i = 1, \dots, n$$

Velja, da je koren koeficienta determinacije R enak korelaciji spremenljivke \mathbf{X} z njeno oceno $\widehat{\mathbf{X}}$ [Seber-1977]:

$$R = \text{corr}(\mathbf{X}, \widehat{\mathbf{X}})$$

Če upoštevamo običajno pojmovanje visoke koreliranosti, tj. korelacijski koeficient, ki je po absolutni vrednosti večji od 0.7, to pomeni, da lahko šele pri $R^2 \geq 0.5$ govorimo o zadovoljivi natančnosti MLR modela.

Omenimo še en način merjenja natančnosti MLR modela, namreč F statistiko, imenovano tudi *Snedecorjeva F statistika* ali *Fisher-Snedecorjeva statistika* [Seber-1977, Hladnik-2002]:

$$F = \frac{m - n}{n - 1} \frac{\sum_{i=1}^m (\widehat{\mathbf{X}}_i - \overline{\mathbf{X}})^2}{\sum_{i=1}^m (\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2} = \frac{m - n}{n - 1} \frac{R^2}{1 - R^2}$$

kjer je n število parametrov MLR modela, m pa število enačb danega predoločenega sistema. Snedecorjeva F statistika temelji na Snedecorjevi porazdelitvi z $(n - 1, m - n)$ prostostnimi stopnjami. F statistika zavzame nenegativne vrednosti. Večje vrednosti F statistike, ki ustrezajo večjim vrednostim koeficienta determinacije R^2 , pomenijo večjo natančnost modela.

3.3.2 Regresijske predpostavke

Uporaba multiple linearne regresije pri reševanju predoločenega linearnega sistema zahteva izpolnitev naslednjih pogojev [Johnson-2002c]:

1. LINEARNOST SPREMENLJIVK: Obstajati mora približno linearna zveza med odvisno spremenljivko \mathbf{X} na eni strani in neodvisnimi spremenljivkami $\mathbf{X}_1, \dots, \mathbf{X}_n$ na drugi strani, tj. pričakovana vrednost oz. matematično upanje odvisne spremenljivke mora biti linearna funkcija neodvisnih spremenljivk. Indikator linearnosti med odvisno spremenljivko in neodvisnimi spremenljivkami modela je velik koeficient determinacije R^2 . Literatura o vrednotenju učinkovitosti sistemov za dialog z ogrođjem PARADISE v glavnem poroča o koeficientih determinacije R^2 , ki so blizu mejne vrednosti 0.5 [Kamm-1999, Walker-2000, Larsen-2003], pogosto precej nižje [Walker-1997b, Walker-1998, Walker-1999, Walker-2001a, Walker-2001b, Möller-2005], le redko pa presežejo vrednost 0.6 [Litman-2002].
2. NEODVISNOST SPREMENLJIVK: Noben par neodvisnih spremenljivk $\mathbf{X}_i, \dots, \mathbf{X}_n$ ne sme biti preveč koreliran, tj. korelacijski koeficienti $corr(\mathbf{X}_i, \mathbf{X}_j)$ morajo biti po absolutni vrednosti manjši od 0.7. Če to ni tako, je dobljen model lahko zelo občutljiv na majhne merske napake ali spremembe vrednosti neodvisnih spremenljivk. Temu pojavu rečemo *multikolinearnost*. Odvečne neodvisne spremenljivke je zato treba odstraniti iz MLR modela. Zaradi težnje k čim večji natančnosti modela je smiselno odstraniti tiste spremenljivke, ki so z odvisno spremenljivko v nižji korelaciji.

Za napake dobljenega modela napovedovanja odvisne spremenljivke pa velja oz. mora veljati še naslednje:

3. NEPOŠEVNOST NAPAK: Srednja vrednost napake ϵ je enaka 0. To je posledica metode najmanjših kvadratov, na kateri temelji linearna regresija.
4. HOMOSKEDASTIČNOST NAPAK: Varianca napake ϵ mora biti po celotni učni množici enaka. V nasprotnem primeru je korelacija med odvisno spremenljivko in parametri modela lahko zavajajoče povprečje vzorcev višje in nižje korelacije.
5. NORMALNOST NAPAK: Napaka ϵ mora biti normalno porazdeljena slučajna spremenljivka.

Zanimivo vrsto vzorcev predstavljajo t. i. *osamelci*. Tako imenujemo meritve, ki se nenavadno razlikujejo od velike večine ostalih meritev in zato nepredvidljivo vplivajo na natančnost modela [Tabachnick-1996]. Odstranitev osamelcev iz učne množice MLR modela je eden od običajnih regresijskih postopkov.

3.3.3 Pomembnost izbire regresijskih parametrov

Ko izbiramo podmnožico parametrov oz. neodvisnih spremenljivk MLR modela, se zastavi vprašanje, zakaj ne bi vzeli vseh parametrov, ki jih lahko pridobimo. To se zdi smiselno predvsem zato, ker koeficient determinacije R^2 s številom parametrov narašča. Izkaže pa se, da je uporaba vseh parametrov lahko neprimerna iz več razlogov:

- ↪ Pridobiti celotno množico parametrov je včasih težko, časovno zahtevno in/ali samodejno nemogoče.
- ↪ Če se omejimo na manjšestevilnejšo množico parametrov, lahko to včasih bolj natančno določimo.
- ↪ Varčnost je pomembna lastnost dobrih modelov – modeli z manj parametri omogočajo boljši vpogled v odnose med regresijskimi spremenljivkami.
- ↪ Izračuni regresijskih koeficientov so v modelih z veliko spremenljivkami zaradi multikolinearnosti pogosto nestabilni.
- ↪ Pokazati se da, da lahko neodvisne spremenljivke, ki so z odvisno spremenljivko v zelo nizki korelaciji (po absolutni vrednosti pod 0.1), povečajo srednjo kvadratno napako [Patel-2004]. Če take spremenljivke iz modela odstranimo, zmanjšamo napako napovedi.
Za preizkus hipoteze o nekoreliranosti neodvisne spremenljivke \mathbf{X}_i z odvisno spremenljivko \mathbf{X} lahko uporabimo testno statistiko, ki temelji na Studentovi porazdelitvi [Hladnik-2002].
- ↪ Pokazati se tudi da, da lahko neodvisne spremenljivke, ki imajo v MLR modelu majhne neničelne (regresijske) koeficiente oz. uteži, povečajo srednjo kvadratno napako [Patel-2004]. Če takšne spremenljivke iz modela odstranimo, zmanjšamo napako napovedi.

V statistiki obstaja več načinov izbire ‘dobre’ podmnožice MLR parametrov, od katerih ima vsak svoje prednosti in slabosti. Najpogosteje se uporabljajo: *sprednja izbira*, *vzratna eliminacija* in *postopna regresija* [Seber-1977]. Pri vrednotenju učinkovitosti sistemov Čarovnik iz Oza (poglavje 4) bomo uporabljali metodo vzratne eliminacije.

Pri vzratni eliminaciji v izbrano podmnožico parametrov S najprej zberemo vse parametre, s katerimi razpolagamo. V naslednjih korakih izbrano množico S manjšamo – v vsakem koraku odstranimo le en parameter, namreč parameter, ki najmanj pripomore k natančnosti MLR modela. Parameter, ki ga odstranimo v $(i + 1)$ -tem koraku, lahko določimo z uporabo $(i + 1)$ -te *delne F statistike*. Naj bo

$$\overline{\epsilon_i^2} = \frac{1}{m} \sum_{k=1}^m (\mathbf{X}_k - \widehat{\mathbf{X}}_{i_k})^2$$

srednja kvadratna napaka modela po i -tem koraku vzratne eliminacije, ko iz začetne množice S odstranimo i parametrov. V $(i + 1)$ -tem koraku za vsakega izmed preostalih

$n - i$ parametrov izračunamo $(i + 1)$ -to delno F statistiko [Seber-1977]

$$F_{i+1} = (m - n + i) \frac{\overline{\epsilon_{i+1}^2} - \overline{\epsilon_i^2}}{\overline{\epsilon_i^2}},$$

kjer je $\overline{\epsilon_{i+1}^2}$ srednja kvadratna napaka, ki bi jo dobili, če bi iz MLR modela odstranili še ravnokar izbran parameter. Delna F statistika temelji na Snedecorjevi porazdelitvi z $(1, m - n + i)$ prostostnimi stopnjami. Minimum vseh delnih F statistik, izračunanih v tekočem koraku vzvratne eliminacije, ustreza parametru, ki najmanj poveča napako in je zato najboljši kandidat za izključitev iz množice S . Postopek končamo, ko postane napaka modela (oz. najmanjša delna F statistika) ‘prevelika’. Za prag F_{out} običajno izberemo vrednost med 2 in 4, ki pri prostostni stopnji $m - n + i$ večji od 50 ustreza p -vrednosti med 0.18 in 0.05 [Abramowitz-1972]. Torej, postopek vzvratne eliminacije ustavimo pred i -tim korakom, če za najmanjšo delno F statistiko tega koraka velja:

$$F_i > F_{out}$$

Dobra lastnost vzvratne eliminacije je ta, da je vsak parameter vsaj v enem koraku v izbrani množici S , slabost pa predstavljata časovna zahtevnost in numerična nestabilnost začetnega (polnega) modela.

3.3.4 Merjenje zadovoljstva uporabnikov

Omenili smo že, da zadovoljstvo uporabnikov, ki ga izraža odvisna spremenljivka **zadovoljstvo uporabnika** funkcije učinkovitosti, avtorice ogrodja PARADISE merijo z vprašalnikom, podanim v tabeli 3.1. Hone in Graham [Hone-2000] sta opozorila na dejstvo, da ta vprašalnik ne temelji niti na teoriji niti na ustreznih empiričnih raziskavah. Vprašanja so bila najbrž izbrana tako, kot se je v danem trenutku avtoricam zdelo prav. Vsaka merilna tehnika, subjektivna ali objektivna, ki šteje za veljaven psihometrični instrument, pa mora izpolnjevati osnovne pogoje, ki sta jih določila Sanders in McCormick [Sanders-1993]:

1. ZANESLJIVOST: Razlike v ponovljeni meritvi morajo biti posledica dejanskih sprememb v opazovanem pojavu.
2. VELJAVNOST: Meriti moramo tisto, kar mislimo, da merimo.
3. OBJEKTIVNOST: Ocenjevalec ne sme vplivati na meritve.
4. DISKRIMINATIVNOST: Postopek mora biti sposoben meriti tudi majhne spremembe v merilni količini.

Hone in Graham opozarjata na naslednjo sporno točko določanja zadovoljstva uporabnikov, namreč seštevanje ocen, ki naj bi merile popolnoma različne količine, tj. učinkovitost modula za tvorjenje govora, učinkovitost modula za razpoznavanje govora, težavnost pridobivanja informacij, hitrost interakcije, izkušnost uporabnikov, ustreznost odzivov sistema, pričakovano obnašanje sistema in načrtovano rabo sistema v

prihodnosti. To dejanje je upravičeno le, če vsa vprašanja merijo isto količino. V nasprotnem primeru predstavlja vsota nesmiselno vrednost.

Da bi bila vsota ali celo povprečje ocen, ki se nanašajo na učinkovitost katerega izmed modulov sistema za dialog, popolnoma nesmiselna, ni čisto res. Na izbran modul lahko gledamo kot na merjeno količino. Res je sicer, da lahko opazujemo različne vidike obnašanja tega modula, vendar nas ponavadi ne zanimajo le izolirane lastnosti, temveč tudi uspešnost modula kot celote. Idealna za merjenje zadovoljstva uporabnikov z učinkovitostjo izbranega modula bi zato bila ustrezno utežena vsota ali utežena srednja vrednost posameznih ocen – lastnostim, ki so za uporabnike bolj pomembne, bi dali večje uteži. Odgovor na vprašanje, kakšne naj bodo te uteži, pa mora temeljiti na ustreznih empiričnih raziskavah, v katere bodo vključeni uporabniki sistemov za dialog. S tem v zvezi menimo, da tudi seštevanje ocen, dodeljenih vprašanjem za določanje zadovoljstva uporabnikov z različnimi vidiki delovanja sistema za dialog, ni popolnoma nesmiselno. Res je, da metoda ni dodelana, je pa lahko dober kazalec učinkovitosti sistema za dialog.

Ker za nobeno od obstoječih tehnik merjenja zadovoljstva uporabnikov sistemov za dialog [Shriberg-1992, Walker-1998, Bouwman-1998, Hartikainen-2004] ni dokazano, da izpolnjuje pogoje za veljaven psihometrični instrument, je treba vse sklepe, ki zajemajo zadovoljstvo uporabnikov, obravnavati zelo previdno. Žal je bil prvi resen poskus razvoja vprašalnika, ki bi zanesljivo, veljavno, objektivno in diskriminativno meril zadovoljstvo uporabnikov sistemov za dialog, (začasno) prekinjen [Hone-2000]. Pristop, ki prav tako veliko obeta, je modifikacija SERVQUAL metode [Hartikainen-2004].

Vrednotenje učinkovitosti sistemov za dialog bi lahko zasnovali tudi na ocenah, dodeljenih posamičnim vprašanjem (znotraj ogrodja PARADISE) uveljavljenega vprašalnika (tabela 3.1). Ta pristop je sicer sporen, zato ker si ljudje vsebino vprašanja pogosto različno razlagamo. Zaradi tega vsi veljavni psihometrični instrumenti za vsako od merjenih količin vsebujejo več, različno oblikovanih, vprašanj. Še vedno pa se zdi predlagan pristop smiseln vsaj za pridobitev osnovnih podatkov o sposobnosti modeliranja posameznih ocen in o statistično najbolj značilnih parametrih.

Če se pojavi želja po vrednotenju katerega od modulov danega sistema za dialog (npr. modula za vodenje dialoga ali modula za razpoznavanje govora), pa je bolj smiselno sešteti ocene, dodeljene le tistim vprašanjem, ki se nanašajo na učinkovitost oz. obnašanje izbranega modula.

3.3.5 Vplivi razpoznavanja govora na rezultate vrednotenja

Parameter, ki ima v funkciji učinkovitosti zaradi (po absolutni vrednosti) največje uteži najpogosteje najpomembnejšo vlogo, je parameter, ki meri učinkovitost modula za razpoznavanje govora [Walker-1997b, Walker-1998, Kamm-1999, Litman-2002, Larsen-2003]. To je, kakovost razpoznavanja govora ključno vpliva na zadovoljstvo uporabnikov – ob povečani učinkovitosti razpoznavanja govora se poveča tudi zadovoljstvo uporabnikov.

Kaj pa, če nas zanima npr. učinkovitost modula za vodenje dialoga ali razumevanje

naravnega jezika? Na osnovi rezultatov, ki jih podaja literatura, smo prišli do sklepa, da bo vrednotenje učinkovitosti posameznih modulov zelo verjetno zanesljivejše in natančnejše, če odstranimo vpliv razpoznavanja govora, torej simuliramo tako rekoč popolno razpoznavanje. To lahko naredimo tako, da za pridobivanje regresijskih podatkov uporabimo eksperiment Čarovnik iz Oza, v katerem vlogo razpoznavalnika govora ali celo vlogo modulov za razumevanje govora prevzame človek [Hajdinjak-2005, Hajdinjak-2006]. Menimo, da bodo v tem primeru prišli v ospredje tudi tisti parametri modela učinkovitosti, ki jim zaradi izjemnega vpliva učinkovitosti razpoznavanja govora svoje vloge v preteklih študijah ni uspelo dokazati. Trdimo tudi, da bodo tako dobljene uteži funkcije učinkovitosti realneje izražale vpliv parametrov na zadovoljstvo uporabnikov. Nedavno je na ta način razumevanje govora simuliral tudi Möller [Möller-2005]. To je počel zaradi prenizke kakovosti razpoznavalnika govora. V enem primeru je v jezikovne prepise načrtno vključil napake, v drugem primeru, ko zaradi narave eksperimenta to ni bilo mogoče, pa je simuliral popolno razumevanje govora.

V skladu z našim razmišljanjem so tudi ugotovitve, do katerih so prišle Walker, Boland in Kamm [Walker-1999]. Ugotovile so, da se značilnosti in uteži parametrov modela učinkovitosti lahko spremenijo, če izboljšamo razpoznavanje govora.

4 Vrednotenje učinkovitosti

4.1 Uvod

4.2 Izbira regresijskih parametrov

4.3 Korelacije med regresijskimi parametri

4.4 Korelacije z odvisno spremenljivko

4.5 Funkcije učinkovitosti

Učinkovitost obeh sistemov Čarovnik iz Oza vrednotimo z ogrodjem PARADISE. Opišemo postopek izpeljave funkcij učinkovitosti za različne odvisne spremenljivke oz. različne mere zadovoljstva uporabnikov. Zaradi razlikovanja obeh sistemov v vodenju dialoga in z njim povezani predstavitvi znanja utemeljimo definicijo nove mere zadovoljstva uporabnikov kot vsoto ocen, dodeljenih štirim izmed osmih vprašanj, kolikor jih predlaga ogrodje PARADISE.

Pri vrednotenju učinkovitosti obeh sistemov Čarovnik iz Oza pridemo do sklepa o izjemni pomembnosti predstavitve znanja v sistemih za dialog in ugotovimo pozitiven vpliv sodelujočega odgovarjanja na zadovoljstvo uporabnikov.

Z vzvratno eliminacijo poiščemo funkcije učinkovitosti, ki modelirajo ocene uporabnikov, dodeljene posamičnim vprašanjem o njihovem zadovoljstvu z obnašanjem sistema. Identificiramo presenetljive razlike med obema sistemoma Čarovnik iz Oza in pridemo do spoznanja, da je natančnost dobljenih modelov v drugem eksperimentu veliko slabša.

4.1 Uvod

Ogrodje PARADISE bomo uporabili pri vrednotenju učinkovitosti v poglavju 2 predstavljenih sistemov Čarovnik iz Oza. Človek čarovnik je prvem sistemu simuliral razumevanje govora (razpoznavanje govora in razumevanje naravnega jezika) ter vodenje dialoga, v drugem sistemu pa le razumevanje govora. Oba sistema Čarovnik iz Oza sta se torej razlikovala le v načinu vodenja dialoga (v prvem sistemu je to nalogo opravljal čarovnik, v drugem pa modul za samodejno vodenje dialoga) in vrsti podatkovne zbirke. V sistemu, s katerim smo izvajali prvi eksperiment Čarovnik iz Oza, je čarovnik dostopal do relacijske zbirke vremenskih podatkov [Hajdinjak-2002a], v sistemu, s katerim smo izvajali drugi eksperiment Čarovnik iz Oza, pa je modul za samodejno vodenje dialoga (razdelek 2.4) dostopal do sodelujoče podatkovne zbirke, ki je temeljila na dialog mIPC Kripkejevih modelih, predstavljenih v razdelku 2.5.

Dialog mIPC Kripkejevi modeli so nastali na podlagi analize težav, ki so se pojavljale v prvem eksperimentu. Tudi brez objektivnega vrednotenja je bilo namreč jasno, da so uporabniki s sistemom precej manj zadovoljni, če jim ta informacije ponuja skopo, preobširno in/ali na nesodelujoč način. Pričakovali smo, da bo sodelujoča podatkovna zbirka, uporabljena v drugem eksperimentu Čarovnik iz Oza, ki se odlikuje po sposobnosti sodelujočega odgovarjanja, pozitivno vplivala na zadovoljstvo uporabnikov. Da bi preskusili to hipotezo in hkrati ovrednotili še spremembe, ki so bile povezane z vključitvijo samodejnega modula za vodenje dialoga v drugi sistem Čarovnik iz Oza, smo v sklopu raziskav, povezanih z magistrskim delom [Hajdinjak-2004a], uporabili ogrodje PARADISE. Ker smo dobili funkciji učinkovitosti, katerih natančnosti sta se zelo razlikovali ($R^2 = 0.62$ za prvi sistem in $R^2 = 0.24$ za drugi sistem), smo se odločili ogrodje PARADISE podrobno analizirati (poglavje 3) in postopek vrednotenja učinkovitosti ponoviti.

Posebne pomena pri vrednotenju učinkovitosti sistemov za dialog z ogrođjem PARADISE je izbira cen dialoga (razdelek 3.3.3), ki jih delimo na parametre učinkovitosti dialoga in parametre kakovosti dialoga.

4.2 Izbira regresijskih parametrov

Ko smo oba sistema Čarovnik iz Oza vrednotili prvič, smo izbrali 10 regresijskih parametrov. Izbira je temeljila na podobnih raziskavah [Walker-1998, Kamm-1999, Walker-2000, Litman-2002, Larsen-2003], od katerih nobena ni presegla števila 10. Ob ponovnem vrednotenju smo se odločili množico regresijskih parametrov razširiti na 25. Tako kot avtorice ogrođja PARADISE [Walker-1997a] smo izbrali en sam parameter uspešnosti naloge:

- **Kappa koeficient** (κ) meri uspešnost sistema pri reševanju nalog, ki mu jih naloži uporabnik. Napake, do katerih pride pri razumevanju govora in jih sistem v tekoči konverzacijski igri odpravi, ne znižajo vrednosti tega koeficienta. Ker je v naših eksperimentih razumevanje govora simuliral čarovnik, koeficient

κ , izračunan iz podatkov prvega eksperimenta, kaže uspešnost oz. spretnost čarovnika in fleksibilnost grafičnega vmesnika, ki je čarovniku pomagal voditi dialog, pri reševanju navideznih nesporazumov med uporabnikom in čarovnikom. V drugem eksperimentu, ko je vodenje dialoga prevzel posebej za to nalogo zgrajen modul, koeficient κ kaže uspešnost tega modula za vodenje dialoga pri reševanju navideznih nesporazumov med uporabnikom in čarovnikom, ki so nastali ali zaradi tipkarskih napak čarovnika ali zaradi neavtoriziranih posegov čarovnika v pomenske predstavitev uporabnikovih izjav.

Za parametre učinkovitosti dialoga smo izbrali:

- **Povprečni čas dialoga** (MET) meri povprečni čas trajanja konverzacijskih iger QUERY-YN GAME, QUERY-WR GAME, QUERY-WI GAME, QUERY-WL GAME in QUERY-WT GAME, katerih namen je pridobiti določeno informacijo in jih uporabnik vpelje v času svoje interakcije s sistemom. V trajanje ni všteti čas podajanja odzivov sistema, ampak le čas govorne interakcije uporabnika in čas čakanja na odziv sistema. Kratica MET se nanaša na angleško besedno zvezo *Mean Elapsed Time*. Naštete konverzacijske igre bomo v nadaljevanju imenovali *informacijske igre*.
- **Povprečno število potez** (MUM) meri povprečno število konverzacijskih potez, ki jih uporabnik potrebuje za izvedbo ali prekinitve vpeljanih informacijskih iger. Kratica MUM se nanaša na angleško besedno zvezo *Mean User Moves*.

V poglavju 3 smo povedali, da so cene dialoga parametri, katerih minimiranje ugodno vpliva na zadovoljstvo uporabnikov. Včasih je za cene dialoga naravneje vzeti količine, katerih učinek je ravno obraten. Tudi za nekatere oblike parametrov kakovosti dialoga, ki smo jih določili mi, velja, da na zadovoljstvo uporabnika ugodno vpliva njihovo maksimiranje. Izbrali smo naslednje parametre kakovosti dialoga:

- **Izpolnitev naloge** (Comp) se nanaša na mnenje uporabnika o tem, ali je od sistema dobil odgovor na prvo vprašanje oz. prvo nalogo, ki smo mu jo v eksperimentu zastavili (poglavje 2). Parameter Comp zavzame vrednost 0, če uporabnik meni, da ni dobil odgovora na svoje vprašanje, in vrednost 1 v nasprotnem primeru. Oznaka Comp je okrajšava za angleško besedo *Completed*.
- **Število uporabnikovih iniciativ** (NUI) šteje začetne konverzacijske poteze, s katerimi uporabnik vpelje informacijske igre. Kratica NUI se nanaša na angleško besedno zvezo *Number of User Initiatives*.
- **Povprečno število besed** (MWT) meri povprečno število besed, vsebovanih v konverzacijskih potezah uporabnika. Kratica MWT se nanaša na angleško besedno zvezo *Mean Words per Turn*.
- **Povprečni čas odziva** (MRT) meri povprečni čas, ki ga sistem porabi, da se odzove. V prvem eksperimentu je bil ta čas povezan z izbiro odgovorov na grafičnem vmesniku, v drugem pa s tipkanjem pomenskih predstavitev uporabnikovih potez. Kratica MRT se nanaša na angleško besedno zvezo *Mean Response Time*.

- **Število manjkajočih odzivov** (NMR) meri razliko med številom potez sistema in številom potez uporabnika. Ta parameter izraža tako število TIMEOUT potez kakor tudi nepripravljenost uporabnika, da bi sistem odzdravil. Kratica NMR se nanaša na angleško besedno zvezo *Number of Missing Responses*.
- **Število neprimernih iniciativ** (NUR) in **delež neprimernih iniciativ** (URR) merita število oz. delež začetnih potez uporabnika, katerih vsebina ne ustreza domeni sistema. Kratici NUR in URR se nanašata na angleški besedni zvezi *Number of Unsuitable Requests* in *Unsuitable-Request Ratio*.
- **Število neprimernih odzivov** (NIR) in **delež neprimernih odzivov** (IRR) merita število oz. delež kontekstno neprimernih potez sistema. Sem štejemo tudi PARDON poteze, s katerimi sistem uporabnika prosi, naj ponovi zadnjo izjavo. Kratici NIR in IRR se nanašata na angleški besedni zvezi *Number of Inappropriate Responses* in *Inappropriate-Response Ratio*.
- **Število napak** (Error) meri napake sistema, kamor štejemo prekinitve telefonske povezave, neustrezno oblikovane povedi in nasprotujoče si odgovore.
- **Število pomoči** (NHM) in **delež pomoči** (HMR) merita število oz. delež potez sistema, ki uporabniku pomagajo nadaljevati dialog, tj. potez tipa REPLY-HELP in REPLY-TIMEOUT. Kratici NHM in HMR se nanašata na angleški besedni zvezi *Number of Help Messages* in *Help-Message Ratio*.
- **Število preverjanj** (NCM) in **delež preverjanj** (CMR) merita število oz. delež potez tipa CHECK, ki jih izvaja sistem. V prvem eksperimentu čarovnik ni izvajal potez tega tipa. Čarovnik, ki je simuliral popolno razumevanje govora, je sicer na podlagi zgodovine dialoga sklepal o nenavedenih podatkih, za katere pa uporabnika ni prosil, da jih potrdi. Kratici NCM in CMR se nanašata na angleški besedni zvezi *Number of Check Moves* in *Check-Move Ratio*.
- **Število podanih informacij** (NGD) in **delež podanih informacij** (GDR) merita število oz. delež potez, s katerimi sistem uporabniku poda iskane informacije, ki jih najde v podatkovni zbirki. Kratici NGD in GDR se nanašata na angleški besedni zvezi *Number of Given Data* in *Given-Data Ratio*.
- **Število relevantnih informacij** (NRD) in **delež relevantnih informacij** (RDR) merita število oz. delež potez sistema, ki uporabnika usmerjajo k izbiri relevantnih, dosegljivih podatkov. Kratici NRD in RDR se nanašata na angleški besedni zvezi *Number of Relevant Data* in *Relevant-Data Ratio*.
- **Število nepodanih informacij** (NND) in **delež nepodanih informacij** (NDR) merita število oz. delež potez, s katerimi sistem uporabniku sporoča, da nima zahtevanega podatka in ga pri tem ne usmerja k izbiri relevantnih, dosegljivih podatkov. V prvem eksperimentu so to poteze, ki pravijo, da sistem zahtevane informacije trenutno nima ali je sploh ne ponuja. V drugem eksperimentu pusti sistem to vprašanje odprto. Kratici NND in NDR se nanašata na angleški besedni zvezi *Number of No Data* in *No-Data Ratio*.

- **Število prekinjenih zahtev (NAR)** in **delež prekinjenih zahtev (ARR)** merita število oz. delež informacijskih iger, ki jih uporabnik prekine še preden se končajo. Kratici NAR in ARR se nanašata na angleški besedni zvezi *Number of Abandoned Requests* in *Abandoned-Request Ratio*.

Izbrane parametre je treba določiti samodejno, če je to mogoče, v skrajnem primeru pa jih ročno označiti. Zavedati se namreč moramo, da neodvisne spremenljivke funkcije učinkovitosti, ki niso samodejno določljive, skrajšajo uporabnost ogrodja PARADISE – samodejno iskanje problematičnih dialogov in spreminjanje strategije vodenja dialoga med interakcijo tedaj nista več mogoča.

V prvem eksperimentu smo morali večino parametrov določiti ročno. Šele modul za vodenje dialoga, vključen v drugi sistem Čarovnik iz Oza, ki je potek dialoga zelo dobro strukturiral, je omogočil samodejno določljivost velike večine izbranih parametrov. Še vedno je bilo samodejno nemogoče določiti naslednje parametre: **Kappa koeficient** (κ), **izpolnitev naloge** (Comp), **število neprimernih iniciativ** (NUR) in **število napak** (Error). To ni moteče, če funkcija učinkovitosti teh parametrov ne vsebuje.

Zanimivo je, da se **število podanih informacij** (NGD) in **delež podanih informacij** (GDR), **število relevantnih informacij** (NRD) in **delež relevantnih informacij** (RDR) ter **število nepodanih informacij** (NND) in **delež nepodanih informacij** (NDR), ki jih bomo imenovali *parametri podatkovne zbirke*, v literaturi o vrednotenju učinkovitosti sistemov za dialog ne pojavljajo. Razlog je verjetno ta, da imajo razvijalci sistemov za dialog le redko na razpolago podatkovno zbirko, katere struktura bi bila tako zelo časovno odvisna in skopa, kot je naša [Hajdinjak-2002a, Hajdinjak-2002b]. S podobno podatkovno zbirko so imeli opravka npr. razvijalci sistema JUPITER [Zue-2000], ki po telefonu podaja informacije o vremenu. Omenjen tip parametrov pa ni ostal popolnoma neopažen. Walker, Litman, Kamm in Abella [Walker-1998] razmišljajo, da bi velikost podatkovne zbirke lahko značilno vplivala na učinkovitost sistema za dialog.

Ker nas je zanimalo, ali so uporabniki bolj občutljivi na kvantitativne (tj. število) ali na proporcionalne parametre (tj. delež), smo tam, kjer se je zdelo smiselno, merili obe količini.

Srednje vrednosti izbranih regresijskih parametrov v obeh eksperimentih Čarovnik iz Oza so podane v tabeli 4.1. Vrstice s parametri, katerih razlika srednjih vrednosti v obeh eksperimentih je statistično značilna ($p < 0.05$), so potemnjene in navedena je pripadajoča p vrednost. Uporabili smo Studentov primerjalni test [Hladnik-2002].

Tabela 4.1 pravi, da sta bili srednji vrednosti MET in MUM značilno večji ($p < 0.0005$ oz. $p < 0.05$) v drugem eksperimentu. To je zagotovo posledica značilno večje srednje vrednosti parametra NIR ($p < 0.01$) ter implementacije konverzacijske igre CHECK GAME v drugem eksperimentu. Ker je bila večina odzivov na CHECK poteze sestavljena iz manj kot treh besed, je ta strategija vodenja dialoga vodila tudi do značilno nižje srednje vrednosti parametra MWT ($p < 0.0005$). Ne nazadnje pa so na povečanje srednje vrednosti MET vplivali še značilno daljši ($p < 0.0005$) odzivni časi sistema (MRT) v drugem eksperimentu, ki so bili posledica časovno zahtevnega tipkanja pomenskih predstavitev uporabnikovih izjav.

		WOZ1	WOZ2	p
uspešnost				
naloge	Kappa koeficient (κ)	0.94	0.98	
učinkovitost	povprečni čas dialoga (MET)	13.76 s	17.39 s	0.000
dialoga	povprečno število potez (MUM)	1.48 s	1.68 s	0.047
	izpolnitev naloge (Comp)	0.97	0.96	
	število uporabnikovih iniciativ (NUI)	6.49	7.51	0.005
	povprečno število besed (MWT)	9.32 s	7.56 s	0.000
	povprečni čas odziva (MRT)	5.13 s	6.38 s	0.000
	število manjkajočih odzivov (NMR)	0.60	0.75	
	število neprimernih iniciativ (NUR)	0.48	0.13	0.011
	delež neprimernih iniciativ (URR)	0.08	0.02	
	število neprimernih odzivov (NIR)	0.41	0.90	0.009
	delež neprimernih odzivov (IRR)	0.04	0.06	
kakovost	število napak (Error)	0.12	0.06	
	število pomoči (NHM)	0.32	0.40	
dialoga	delež pomoči (HMR)	0.03	0.03	
	število preverjanj (NCM)*	-	2.19	
	delež preverjanj (CMR)*	-	0.16	
	število podanih informacij (NGD)	4.07	4.35	
	delež podanih informacij (GDR)	0.67	0.58	
	število relevantnih informacij (NRD)	0.70	2.06	0.000
	delež relevantnih informacij (RDR)	0.10	0.28	0.005
	število nepodanih informacij (NND)	1.67	0.94	0.000
	delež nepodanih informacij (NDR)	0.22	0.12	
	število prekinjenih zahtev (NAR)	0.05	0.16	
	delež prekinjenih zahtev (ARR)	0.01	0.02	
	zadovoljstvo uporabnika (US)	34.08	31.96	0.015

* V prvem eksperimentu čarovnik ni izvajal CHECK potez.

Tabela 4.1: Srednje vrednosti izbranih regresijskih parametrov v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.

V drugem eksperimentu Čarovnik iz Oza sta bili srednji vrednosti parametrov podatkovne zbirke NRD in RDR skoraj trikrat večji ($p < 0.0005$ oz. $p < 0.005$) kot v prvem eksperimentu. Že ta podatek potrjuje, da je imel implementiran modul za vodenje dialoga skupaj s sodelujočo predstavitvijo znanja veliko večjo sposobnost usmerjanja uporabnika k izbiri relevantnih, dosegljivih podatkov kot čarovnik, ki je prek grafičnega vmesnika dostopal do relacijske zbirke vremenskih podatkov. Posledično je bila srednja vrednost parametra podatkovne zbirke NND nižja ($p < 0.0005$) v drugem eksperimentu.

4.3 Korelacije med regresijskimi parametri

Omenili smo (razdelek 3.3.2), da noben par neodvisnih spremenljivk, na katerih izvajamo vzvratno eliminacijo, ne sme biti previsoko koreliran, saj bi bil dobljen MLR model sicer lahko zelo občutljiv na majhne merske napake ali spremembe vrednosti regresijskih spremenljivk. Tabela 4.2 podaja korelacijske koeficiente med izbranimi regresijskimi parametri, dobljene v prvem eksperimentu, tabela 4.3 pa korelacijske koeficiente, dobljene v drugem eksperimentu Čarovnik iz Oza.

Zaradi uporabnikov, ki so sodelovali v prvem eksperimentu, niso pa oddali izpolnjenega vprašalnika o svojem zadovoljstvu z obnašanjem sistema, se je število vzorcev za prvi eksperiment zmanjšalo s 76 na 73. V drugem eksperimentu je vprašalnik izpolnilo vseh 68 uporabnikov.

Za podatke iz prvega eksperimenta smo ugotovili, da sta mejni vrednosti 95% intervala zaupanja za korelacijski koeficient 0.7 (meja izločanja zaradi previsoke korelacije) enaki 0.56 in 0.80. Polja s korelacijskimi koeficienti, ki po absolutni vrednosti presegajo spodnjo mejo 0.56, so v tabeli 4.2 potemnjena.

Pozitivne korelacije med parametri, ki na različen način (tj. kot število ali kot delež) merijo iste dogodke, so povsem pričakovane. Presenetli edino popolna nekoreliranost parametrov podatkovne zbirke NGD in GDR, pri čemer povejmo, da dobimo vrednost parametra GDR kot kvocient vrednosti NGD in NUI. Zanimivo je tudi, da korelira NND z NGD pozitivno (0.60), z GDR pa negativno (-0.57).

Visoke korelacije med NUI, na eni strani, ter NGD (0.93), NRD (0.68) in NND (0.83), na drugi strani, so popolnoma razumljive – s številom iniciativ se poveča tudi število podanih informacij, število ponujenih relevantnih informacij ter število izjav, s katerimi sistem uporabniku sporoča, da zahtevane informacije nima. Najnižja izmed omenjenih korelacij je korelacija med NUI in NRD. Razlog je zelo verjetno opazna nekonsistentnost čarovnika v izbiranju odgovorov. Če to res drži, se to v drugem eksperimentu ne bi smelo ponoviti.

Tudi koreliranost parametrov NMR in NHM (0.67) ni presenetljiva – ko se uporabnik ni odzval, mu je čarovnik pomagal ponovno vzpostaviti dialog. Podobno je bilo v drugem eksperimentu. Korelacijski koeficient ni še večji, zato ker na vrednost parametra NMR vpliva tudi obstoj začetnega in končnega pozdrava uporabnika.

Za podatke iz drugega eksperimenta smo ugotovili, da sta mejni vrednosti 95% intervala zaupanja za korelacijski koeficient 0.7 (meja izločanja zaradi previsoke korelacije) enaki 0.56 in 0.81. Polja s korelacijskimi koeficienti, ki po absolutni vrednosti presegajo spodnjo mejo 0.56, so v tabeli 4.3 potemnjena.

V drugem eksperimentu je NUI visoko koreliral z NCM (0.58) in NGD (0.86). Korelacija NUI z NRD (0.52) in NND (0.45) je bila nekoliko nižja kot v prvem eksperimentu. Zelo visoka koreliranost NUI z NGD v primerjavi s koreliranostjo z NRD ali NND kaže na uspešno usmerjanje sistema oz. modula za vodenje dialoga k izbiri dosegljivih informacij in s tem k doseganju uporabnikovih ciljev.

Zanimiva je relativno visoka negativna korelacija med NUI in MUM. V prvem ekspe-

κ	MET	MUM	Comp	NUI	MWT	MRT	NMR	NUR	URR	NIR	IRR	Error	NHM	HMR	NGD	GDR	NRD	RDR	NND	NDR	NAR	ARR	
1.0																							
MET	-0.4	1.0																					
MUM	-0.2	0.5	1.0																				
Comp	0.0	0.1	0.1	1.0																			
NUI	0.2	-0.3	-0.5	-0.3	1.0																		
MWT	-0.2	0.6	-0.0	-0.0	-0.1	1.0																	
MRT	-0.2	0.4	-0.3	0.0	0.2	0.1	1.0																
NMR	-0.1	0.2	-0.5	-0.2	0.4	0.2	0.4	1.0															
NUR	0.2	-0.2	-0.1	0.1	0.2	-0.2	-0.2	0.0	1.0														
URR	0.2	-0.2	0.1	-0.1	-0.2	-0.3	-0.0	0.8	1.0														
NIR	-0.2	-0.0	-0.3	-0.2	0.5	0.0	0.3	0.4	-0.1	-0.1	1.0												
IRR	-0.4	0.1	-0.1	-0.0	0.2	0.0	0.2	0.2	-0.2	0.8	1.0												
Error	0.1	-0.1	-0.3	0.1	0.4	0.1	0.2	0.1	0.1	-0.1	0.3	0.3	1.0										
NHM	0.0	0.2	-0.1	-0.2	0.3	0.1	0.3	0.7	0.1	0.1	0.4	0.1	0.1	1.0									
HMR	0.0	0.4	0.0	-0.0	0.0	0.2	0.2	0.6	0.1	0.2	0.1	-0.1	-0.1	0.9	1.0								
NGD	0.2	-0.3	-0.5	-0.3	0.9	-0.1	0.2	0.4	0.0	-0.2	0.4	0.1	0.4	0.3	0.0	1.0							
GDR	-0.2	0.4	0.4	0.1	-0.3	0.1	0.1	0.0	-0.4	-0.4	-0.1	-0.1	-0.1	0.0	0.1	1.0							
NRD	0.1	-0.3	-0.4	-0.3	0.7	-0.1	0.1	0.3	0.1	-0.0	0.4	0.1	0.2	0.3	0.0	0.6	-0.3	1.0					
RDR	0.2	-0.3	-0.2	-0.0	0.1	-0.2	-0.0	-0.0	0.1	-0.0	0.1	0.0	-0.0	0.0	-0.1	0.1	-0.3	0.7	1.0				
NND	0.1	-0.3	-0.4	-0.2	0.8	-0.1	0.2	0.2	0.3	0.1	0.4	0.2	0.4	0.2	0.0	0.6	-0.6	0.3	-0.1	1.0			
NDR	0.1	-0.3	-0.2	-0.1	0.2	-0.1	-0.1	-0.1	0.4	0.4	0.1	0.0	0.1	0.0	-0.0	-0.1	-0.8	-0.0	-0.2	0.7	1.0		
NAR	0.0	0.1	-0.1	0.0	-0.1	0.1	0.1	0.2	-0.1	-0.1	0.0	0.1	-0.1	0.1	0.2	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1.0	
ARR	-0.0	0.2	-0.1	0.0	-0.1	0.2	0.1	0.2	-0.1	-0.1	0.0	0.1	-0.1	0.1	0.3	-0.1	-0.0	-0.1	-0.2	-0.1	-0.1	0.9	1.0

Tabela 4.2: Korelacijski koeficienti parametrov v prvem eksperimentu Čarovnik iz Oza.

	κ	MET	MUM	Comp	NUI	MWT	MRT	NMR	NUR	URR	NIR	IRR	Error	NHM	HMR	NCM	CMR	NGD	GDR	NRD	RDR	NND	NDR	NAR	ARR	
κ	1.0																									
MET	-0.3	1.0																								
MUM	-0.2	0.6	1.0																							
Comp	0.0	-0.3	-0.3	1.0																						
NUI	0.1	-0.3	-0.5	0.2	1.0																					
MWT	-0.3	0.3	-0.2	0.1	-0.0	1.0																				
MRT	-0.1	0.6	-0.1	-0.2	0.0	0.2	1.0																			
NMR	0.0	0.1	-0.1	0.1	0.2	-0.1	0.0	1.0																		
NUR	-0.1	0.2	0.0	-0.1	0.3	-0.0	0.0	0.1	1.0																	
URR	-0.2	0.4	0.2	-0.2	-0.0	0.1	0.1	0.1	0.8	1.0																
NIR	-0.3	0.2	0.2	0.1	0.5	-0.1	0.1	0.2	0.5	0.2	1.0															
IRR	-0.6	0.4	0.4	0.1	0.1	-0.0	0.1	0.1	0.3	0.3	0.9	1.0														
Error	0.1	0.1	0.1	0.1	0.2	0.0	0.1	0.0	-0.1	-0.1	0.2	0.1	1.0													
NHM	0.1	0.2	0.1	0.0	0.0	-0.3	-0.0	0.7	0.2	0.1	0.1	0.0	-0.1	1.0												
HMR	0.1	0.3	0.1	-0.0	-0.1	-0.3	0.0	0.6	0.1	0.1	-0.0	-0.0	-0.1	0.9	1.0											
NCM	-0.1	0.0	0.1	0.1	0.6	-0.3	-0.1	0.2	0.3	0.1	0.4	0.3	0.2	0.1	0.0	1.0										
CMR	-0.2	0.2	0.5	0.0	0.1	-0.3	-0.0	0.1	0.1	0.2	0.2	0.1	0.1	0.1	0.1	0.8	1.0									
NGD	0.1	-0.3	-0.4	0.2	0.9	-0.1	-0.0	0.1	0.3	0.0	0.5	0.1	0.1	0.0	-0.1	0.5	0.1	1.0								
GDR	-0.1	-0.1	0.1	0.1	-0.0	-0.1	-0.1	-0.2	0.0	0.0	0.1	0.1	-0.0	-0.1	-0.1	0.0	0.1	0.4	1.0							
NRD	0.1	-0.1	-0.3	0.1	0.5	0.0	0.0	0.4	-0.0	-0.1	0.1	-0.0	0.1	0.2	0.1	0.4	0.1	0.2	-0.5	1.0						
RDR	-0.0	0.1	0.0	-0.1	-0.1	0.0	-0.1	0.3	-0.1	-0.1	-0.1	-0.0	-0.1	0.2	0.2	0.1	0.1	-0.3	-0.6	0.7	1.0					
NND	0.1	-0.1	-0.3	0.1	0.4	0.2	0.1	-0.0	0.1	0.0	0.1	-0.0	0.4	-0.1	-0.2	0.2	-0.1	0.1	-0.2	0.1	-0.2	1.0				
NDR	0.2	-0.0	-0.2	0.0	0.1	0.2	0.1	-0.1	0.0	0.0	-0.1	-0.1	0.2	-0.1	-0.2	-0.1	-0.2	-0.1	-0.5	-0.1	-0.2	0.8	1.0			
NAR	-0.1	-0.0	-0.0	-0.1	0.2	-0.1	-0.0	-0.0	0.4	0.2	0.3	0.2	-0.2	0.2	0.1	0.1	0.1	0.2	-0.2	-0.1	-0.1	-0.0	-0.1	1.0		
ARR	-0.1	0.0	0.1	-0.2	-0.1	-0.2	0.1	0.0	0.2	0.2	0.2	0.2	-0.1	0.1	0.2	0.0	0.1	-0.1	-0.2	-0.1	-0.0	-0.1	-0.1	0.9	1.0	

Tabela 4.3: Korelacijski koeficienti parametrov v drugem eksperimentu Čarovnik iz Oza.

rimentu je bila -0.53 , v drugem pa -0.51 . Ta podatek kaže, da so se bili uporabniki sposobni prilagajati obnašanju sistema in so se v času interakcije z njim uspešno učili vodenja učinkovitejšega dialoga.

4.4 Korelacije z odvisno spremenljivko

V obeh eksperimentih Čarovnik iz Oza so uporabniki ocenili svoje zadovoljstvo tako, da so podali stopnjo strinjanja z izjavami o obnašanju oz. učinkovitosti sistema (tabela 3.1). Splošno **zadovoljstvo uporabnika** (US) smo dobili kot vsoto ocen, zbranih z omenjenim vprašalnikom. Vrednosti parametra US, ki bo odvisna spremenljivka osnovnega MLR modela učinkovitosti, zato ležijo med 8 in 40. Srednja vrednost US za prvi eksperiment je enaka 34.08 (s standardnim odklonom 5.07), za drugega pa 31.96 (s standardnim odklonom 4.99). Obe srednji vrednosti zadovoljstva uporabnikov se statistično značilno razlikujeta ($p < 0.015$). Glej tabelo 4.1.

	WOZ1		WOZ2	
	kor. koef.	p -vrednost	kor. koef.	p -vrednost
Kappa koeficient (κ)	0.06	0.615	0.27	0.028
povprečni čas dialoga (MET)	0.18	0.127	-0.29	0.015
povprečno število potez (MUM)	0.38	0.001	-0.29	0.015
izpolnitev naloge (Comp)	0.29	0.014	0.20	0.101
število uporabnikovih iniciativ (NUI)	-0.60	0.000	0.13	0.306
povprečno število besed (MWT)	-0.02	0.872	0.01	0.913
povprečni čas odziva (MRT)	-0.18	0.119	-0.12	0.325
število manjkajočih odzivov (NMR)	-0.16	0.191	-0.05	0.693
število neprimernih iniciativ (NUR)	-0.16	0.175	-0.12	0.337
delež neprimernih iniciativ (URR)	0.08	0.520	-0.17	0.171
število neprimernih odzivov (NIR)	-0.33	0.005	-0.06	0.650
delež neprimernih odzivov (IRR)	-0.17	0.143	-0.22	0.077
število napak (Error)	-0.19	0.107	0.05	0.670
število pomoči (NHM)	-0.15	0.209	-0.09	0.475
delež pomoči (HMR)	0.00	0.981	-0.10	0.408
število preverjanj (NCM)*	-	-	-0.16	0.191
delež preverjanj (CMR)*	-	-	-0.36	0.003
število podanih informacij (NGD)	-0.47	0.000	0.16	0.181
delež podanih informacij (GDR)	0.33	0.004	0.20	0.108
število relevantnih informacij (NRD)	-0.32	0.005	0.03	0.821
delež relevantnih informacij (RDR)	-0.00	0.986	-0.21	0.092
število nepodanih informacij (NND)	-0.64	0.000	0.03	0.786
delež nepodanih informacij (NDR)	-0.33	0.004	0.04	0.735
število prekinjenih zahtev (NAR)	-0.03	0.815	-0.15	0.230
delež prekinjenih zahtev (ARR)	-0.02	0.886	-0.19	0.126

* V prvem eksperimentu čarovnik ni izvajal CHECK potez.

Tabela 4.4: Korelacijski koeficienti in p -vrednosti za regresijske parametre glede na US v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.

V razdelku 3.3.3 smo povedali, da je treba tiste neodvisne spremenljivke, ki so z odvisno spremenljivko v zelo nizki korelaciji ($p > 0.05$), iz modela odstraniti. Korelacijski koeficienti in p -vrednosti, ki glede na zadovoljstvo uporabnikov US pripadajo izbranim regresijskim parametrom, so podani v tabeli 4.4. Uporabili smo Studentov test z $n - 2$ prostostnimi stopnjami [Hladnik-2002], kjer je n velikost učne množice, tj. $n = 73$ v prvem eksperimentu in $n = 68$ v drugem eksperimentu.

Za podatke iz prvega eksperimenta smo ugotovili, da je zgornja meja 95-odstotnega intervala zaupanja za korelacijski koeficient 0.1 (meja izločanja zaradi prenizke korelacije) enaka 0.23, za podatke iz drugega eksperimenta pa 0.24. Polja, ki vsebujejo značilne korelacijske koeficiente ($p < 0.05$), to je korelacijske koeficiente, ki po absolutni vrednosti presegajo zgornjo mejo 95-odstotnega intervala zaupanja, so v tabeli 4.4 potemnjena.

V prvem eksperimentu je z neodvisno spremenljivko US značilno koreliralo 9 parametrov (in sicer MUM, Comp, NUI, NIR, NGD, GDR, NRD, NND in NDR), v drugem pa le 4 (in sicer κ , MET, MUM in CMR). Presenetljivo je, da je le en parameter zastopan v obeh množicah, namreč MUM. Iz teh množic moramo odstraniti še parametre, ki bi lahko povzročali multikolinearnost modela (razdelek 3.3.2).

Z uporabo tabele 4.2 pridemo do sklepa, da bo iskana funkcija učinkovitosti prvega sistema Čarovnik iz Oza dovolj natančna, če v množico neodvisnih regresijskih spremenljivk, na kateri bomo izvajali vzvratno eliminacijo, vključimo naslednje parametre:

- ↪ **povprečno število potez** (MUM),
- ↪ **izpolnitev naloge** (Comp),
- ↪ **število neprimernih odzivov** (NIR),
- ↪ **število relevantnih informacij** (NRD) in
- ↪ **število nepodanih informacij** (NND).

Iskana funkcija učinkovitosti drugega sistema Čarovnik iz Oza bo dovolj natančna (tabela 4.3), če v množico neodvisnih regresijskih spremenljivk, na kateri bomo izvajali vzvratno eliminacijo, vključimo naslednje parametre:

- ↪ **Kappa koeficient** (κ),
- ↪ **povprečni čas dialoga** (MET) in
- ↪ **delež preverjanj** (CMR).

Že število vhodnih parametrov da slutiti, da lahko v drugem primeru pričakujemo funkcijo učinkovitosti, katere koeficient determinacije R^2 bo manjši kot v prvem primeru.

Tabela 4.4 skriva še eno zanimivost. Opazimo namreč, da so bili uporabniki prvega sistema bolj dojemljivi za kvantitativne parametre (tj. NUR, NIR, NHM, NGD, NRD, NND, NAR), kot pa za njim pripadajoče proporcionalne parametre (tj. URR, IRR, HMR, GDR, RDR, NDR, ARR). V drugem eksperimentu je bilo ravno obratno – vsi parametri, ki merijo delež, bolj močno korelirajo z zadovoljstvom uporabnikov US kot parametri, ki merijo število. Nadaljnje raziskave glede tega vprašanja so nujne.

V študiji učinkovitosti sistemov za branje elektronske pošte [Walker-1998] je bilo ugotovljeno, da **izpolnitev naloge** (Comp) močneje vpliva na **zadovoljstvo uporabnika**

(US) kot **Kappa koeficient** (κ). Razlog, ki ga navajajo, je ta, da naj bi uporabniki velikokrat drugače dojemali delovanje sistema, kot ga podaja κ . V našem prvem eksperimentu Čarovnik iz Oza je κ izražal predvsem delo čarovnika, ki je simuliral razumevanje govora in vodenje dialoga, v drugem eksperimentu pa delo modula za samodejno vodenje dialoga. Parameter Comp se je v obeh eksperimentih nanašal le na prvo nalogo, ki smo jo uporabniku naložili, kar je najverjetneje razlog, zakaj sami nismo prišli do podobne ugotovitve. Po eni strani sta bila v naših eksperimentih κ in Comp nekorelirana, po drugi strani pa je v drugem eksperimentu κ celo močnejše vplival na zadovoljstvo uporabnikov (tabela 4.4).

4.5 Funkcije učinkovitosti

Za statistično obdelavo podatkov ter manipulacijo z izbranimi regresijskimi parametri smo razvili lastna programska orodja, napisana v programskem jeziku MATLAB. Ta orodja omogočajo naslednje:

- ↪ izbiro in definicijo poljubnih parametrov, tj. odvisnih in neodvisnih spremenljivk,
- ↪ izračun vrednosti izbranih parametrov na dani učni množici vzorcev,
- ↪ izračun raznih statističnih količin, kot so npr. srednja vrednost, varianca in standardni odklon,
- ↪ normalizacijo parametrov,
- ↪ izračun korelacijskih koeficientov, p -vrednosti in intervalov zaupanja za izbrano množico parametrov,
- ↪ izbiro podmnožice parametrov, primerne kot vhod v postopek multiple linearne regresije,
- ↪ izbiro regresijskega postopka, regresijskega kriterija in regresijskega praga,
- ↪ izvedbo multiple linearne regresije,
- ↪ identifikacijo osamelcev v učni množici vzorcev in njihovo odstranitev iz učne množice,
- ↪ izpis poteka multiple linearne regresije,
- ↪ merjenje natančnosti dobljenih modelov, vključno s koeficienti determinacije, ki ustrezajo posameznim korakom regresijskega postopka,
- ↪ izračun uteži posameznih parametrov ter določitev funkcije učinkovitosti.

Vzvratno eliminacijo bomo izvajali za različne odvisne spremenljivke oz. različne načine merjenja zadovoljstva uporabnikov.

4.5.1 Zadovoljstvo uporabnikov kot vsota ocen

Najprej izpeljimo funkciji učinkovitosti obeh sistemov Čarovnik iz Oza za odvisno spremenljivko **zadovoljstvo uporabnika** (US), ki jo uporablja ogrodje PARADISE. Osamelce, ki se nenavadno razlikujejo od velike večine ostalih vzorcev in zato nepredvidljivo vplivajo na natančnost MLR modela (razdelek 3.3.2), moramo iz učne množice odstraniti.

Po postopku vzratne eliminacije za $F_{out} = 4$ pri p -vrednosti približno enaki 0.05 (razdelek 3.3.3) na celotni učni množici, pridobljeni v prvem eksperimentu Čarovnik iz Oza, pri čemer vzamemo **zadovoljstvo uporabnika** (US) kot odvisno spremenljivko, **povprečno število potez** (MUM), **izpolnitev naloge** (Comp), **število neprimer-nih odzivov** (NIR), **število relevantnih informacij** (NRD) in **število nepodanih informacij** (NND) pa kot neodvisne spremenljivke, identificiramo 5 osamelcev. Ti vzorci, ki predstavljajo slabih 7% vseh vzorcev, po napaki napovedi dejanske vrednosti $\mathcal{N}(\text{US})$ z dobljenim MLR modelom,

$$0.14 * \mathcal{N}(\text{Comp}) - 0.60 * \mathcal{N}(\text{NND}),$$

zelo negativno izstopajo. Pripadajoče napake napovedi so za vsaj 25% večje od napak, pripadajočih ostalim vzorcem. To je meja, na osnovi katere smo izločali osamelce tudi v ostalih primerih. Velikost učne množice končnega MLR modela se zaradi odstranitve osamelcev zmanjša s 73 na 68.

Postopek vzratne eliminacije ponovimo na zmanjšani učni množici vzorcev. Tabela 4.5 podaja dobljene delne F statistike, pripadajoče koeficiente determinacije R^2 ter parametre, ki jih v posameznih korakih iz modela odstranimo. Postopek vzratne eliminacije ustavimo, ko delna F statistika preseže vrednost 4 ali ko v modelu ostaneta le dva parametra. Dodatno omejitev vsebovanja vsaj dveh parametrov smo vpeljali zato, ker se zdi vrednotenje sistema za dialog na osnovi funkcije učinkovitosti, ki vsebuje en sam parameter, preveč poenostavljeno. Postopek vzratne eliminacije na podatkih iz prvega eksperimenta zato ustavimo pred 4. korakom, ko (pri $F_4 = 3.81$) v modelu ostaneta le dva parametra.

	F_i	R^2	odstranjen parameter
poln model	-	0.59	-
1. korak ($i = 1$)	0.36	0.59	NIR
2. korak ($i = 2$)	0.82	0.59	MUM
3. korak ($i = 3$)	1.29	0.58	Comp
4. korak ($i = 4$)	3.81	0.55	NRD

Tabela 4.5: Tabela vzratne eliminacije za prvi sistem Čarovnik iz Oza in odvisno spremenljivko US.

Iz začetnega MLR modela z vzratno eliminacijo odstranimo tri parametre, namreč NIR, MUM in Comp. Funkcija učinkovitosti za prvi sistem Čarovnik iz Oza in

odvisno spremenljivko US_1 , ki se nanaša na podatke, pridobljene v prvem eksperimentu Čarovnik iz Oza, je zato taka:

$$\widehat{\mathcal{N}}(US_1) = -0.16 * \mathcal{N}(\text{NRD}) - 0.69 * \mathcal{N}(\text{NND})$$

Dobljena funkcija učinkovitosti pojasnjuje 58% variance, tj. $R^2 = 0.58$. Njena najizrazitejša parametra, ki oba negativno vplivata na **zadovoljstvo uporabnika** (US_1), sta parametra podatkovne zbirke **število relevantnih informacij** (NRD) in **število nepodanih informacij** (NND).

Postopek identifikacije osamelcev in izpeljave funkcije učinkovitosti ponovimo še za drugi sistem Čarovnik iz Oza. Po postopku vzvratne eliminacije za $F_{out} = 4$ pri p -vrednosti približno enaki 0.05 na celotni učni množici, pridobljeni v drugem eksperimentu Čarovnik iz Oza, pri čemer vzamemo **zadovoljstvo uporabnika** (US) kot odvisno spremenljivko, **Kappa koeficient** (κ), **povprečni čas dialoga** (MET) in **delež preverjanj** (CMR) pa kot neodvisne spremenljivke, spet identificiramo 5 osamelcev. Ti vzorci sedaj predstavljajo dobrih 7% vseh vzorcev in po napaki napovedi dejanske vrednosti $\mathcal{N}(US)$ z dobljenim MLR modelom,

$$-0.22 * \mathcal{N}(\text{MET}) - 0.30 * \mathcal{N}(\text{CMR}),$$

zelo negativno izstopajo. Velikost učne množice končnega MLR modela se zaradi odstranitve osamelcev zmanjša z 68 na 63.

Postopek vzvratne eliminacije ponovimo na zmanjšani učni množici vzorcev. Tabela 4.6 podaja dobljene delne F statistike, pripadajoče koeficiente determinacije R^2 ter parametre, ki jihv posameznih korakih iz modela odstranimo. Postopek vzvratne eliminacije ustavimo pred 2. korakom, ko delna F statistika preseže vrednost 4.

	F_i	R^2	odstranjen parameter
poln model	-	0.26	-
1. korak ($i = 1$)	2.19	0.24	κ
2. korak ($i = 2$)	6.12	0.16	MET

Tabela 4.6: Tabela vzvratne eliminacije za drugi sistem Čarovnik iz Oza in odvisno spremenljivko US.

Iz začetnega MLR modela z vzvratno eliminacijo odstranimo en sam parameter, namreč κ . Funkcija učinkovitosti za drugi sistem Čarovnik iz Oza in odvisno spremenljivko US_2 , ki se nanaša na podatke, pridobljene v drugem eksperimentu Čarovnik iz Oza, je zato taka:

$$\widehat{\mathcal{N}}(US_2) = -0.29 * \mathcal{N}(\text{MET}) - 0.33 * \mathcal{N}(\text{CMR})$$

Dobljena funkcija učinkovitosti pojasnjuje 24% variance, tj. $R^2 = 0.24$, in ima dva parametra – **povprečni čas dialoga** (MET) in **delež preverjanj** (CMR), ki oba negativno vplivata na **zadovoljstvo uporabnika** (US_2).

Tudi razširitev množice regresijskih parametrov, ponovno označevanje posnetkov in ponovno vrednotenje učinkovitosti obeh sistemov ni odpravilo velike razlike v natančnosti obeh funkcij učinkovitosti ($R^2 = 0.58$ proti $R^2 = 0.24$). Sklenemo lahko, da

nam odvisne spremenljivke **zadovoljstvo uporabnika (US)** v naših eksperimentih ni uspelo dovolj dobro modelirati. Razlog je zelo verjetno ta, da ne merimo tistega, kar bi želeli meriti. Po drugi strani pa uporaba seštevka vseh pridobljenih ocen tudi ni najzanesljiviši kazalec zadovoljstva uporabnikov z učinkovitostjo modula za vodenje dialoga, katerega merjenje smo si zastavili za cilj.

4.5.2 Zadovoljstvo uporabnikov kot vsota izbranih ocen

V razdelku 3.3.4 smo argumentirali, da je v primeru, kot je naš, ko želimo poiskati razlike med dvema različicama sistemov Čarovnik iz Oza, bolj smiselno sešteti le ocene, dodeljene vprašanjem, ki se nanašajo na vpeljane spremembe. V našem primeru je to modul za vodenje dialoga v povezavi s predstavitvijo znanja. Menimo, da so vprašanja, ki te spremembe najbolje merijo, naslednja:

2. *Ali vas je sistem razumel?* (ASR)

Vprašanje naj bi merilo učinkovitost razumevanja govora. Ker pa je v naših eksperimentih čarovnik simuliral tako rekoč popolno razumevanje govora, to ni bilo tako. V prvem eksperimentu je to vprašanje dobilo najvišjo oceno izmed vseh (tabela 2.1). V drugem eksperimentu, ko čarovnik, v nasprotju s prvim eksperimentom, v pomenske predstavitve uporabnikovih potez ni dodajal podatkov, na katere se je dalo sklepati iz zgodovine dialoga, se to vprašanje nanaša predvsem na modul za vodenje dialoga oz. njegovo učinkovitost pri polnjenju predalčkov.

3. *Ali ste brez težav prišli do odgovorov na vaša vprašanja?* (TE)

Vprašanje naj bi merilo težavnost pridobivanja informacij. Nedvomno se nanaša na uspešnost čarovnika pri uravnavanju dialoga oz. učinkovitost modula za vodenje dialoga. Pri tem ima pomembno vlogo tudi predstavitev znanja.

6. *Ali se je sistem na vaše izjave odzival hitro (brez pojasnilnih vprašanj)?* (SR)

Vprašanje naj bi merilo ustreznost sistemovih odzivov. Uporabnike sprašuje po mnenju o strategiji vodenja dialoga, ki je bila v drugem eksperimentu del modula za vodenje dialoga.

7. *Ali se je sistem obnašal tako, kot ste med dialogom od njega pričakovali?* (EB)

Vprašanje naj bi merilo ujemanje med pričakovanim in dejanskim obnašanjem sistema. Vsekakor je tesno povezano z načinom vodenja dialoga in predstavitvijo znanja, ki je predpogoj sodelujočega načina odgovarjanja.

Vsoto ocen, dodeljenih naštetim vprašanjem, imenujmo **zadovoljstvo uporabnika z vodenjem dialoga in ravniyo sodelujočega odgovarjanja (DM)**. Ta spremenljivka zavzame vrednosti med 4 in 20. Povejmo še, zakaj ostala vprašanja iz tabele 3.1 niso dobra:

1. *Ali ste sistem brez težav razumeli?* (TTS)

Vprašanje naj bi merilo kakovost umetnega tvorjenja govora. Modul za umetno tvorjenje govora je bil v obeh sistemih isti [Gros-1997], povrh pa še neodvisen od modula za vodenje dialoga.

4. *Ali je bila hitrost interakcije s sistemom primerna?* (IP)

Vprašanje naj bi merilo ustreznost hitrosti interakcije s sistemom. V naših eksperimentih je bila hitrost odzivov v največji meri odvisna od čarovnika in hkrati nerealen pokazatelj hitrosti dograjena sistema. Pri tem omenimo, da so uporabniki najslabše ocenili ravno ta vidik svoje interakcije s sistemom (tabela 2.1). Čarovnik je namreč za odziv na uporabnikovo potezo v prvem eksperimentu potreboval v povprečju 5.13 sekund, v drugem pa celo 6.38 sekund (tabela 4.1).

5. *Ali ste na vsakem koraku dialoga vedeli, kaj morate povedati?* (UE)

Vprašanje naj bi merilo izkušnost uporabnikov. Ocene, ki so jih uporabniki dodelili temu vprašanju, so bile v obeh eksperimentih podobno visoke kot ocene, ki so jih dodelili učinkovitosti razumevanja govora. To pa najbrž zato, ker je bil čarovnik tisti, ki je v trenutku negotovosti sprožil potezo, ki naj bi uporabniku pomagala ponovno navezati dialog. Oba sistema Čarovnik iz Oza se tako rekoč nista razlikovala niti v načinu nudenja pomoči niti v kakovosti razumevanja govora ali umetnega tvorjenja govora, ki bi lahko prav tako opazno vplivala na zmedenost uporabnika.

8. *Glede na vašo trenutno izkušnjo s sistemom, ali mislite, da boste sistem še kdaj poklicali?* (FU)

Vprašanje naj bi merilo rabo sistema v prihodnosti oz. možnost za komercialno uveljavitev sistema. Po eni strani vemo, da uporabniki v naših eksperimentih niso bili pod časovnimi omejitvami in niso odgovarjali za stroške telefonskih pogovorov. Po drugi strani pa menimo, da pozitivne izkušnje uporabnikov v testiranjih sistemov niso kazalec njihovih prihodnjih, težko predstavljenih navad.

Izpeljimo torej funkciji učinkovitosti obeh sistemov Čarovnik iz Oza za odvisno spremenljivko **zadovoljstvo uporabnika z vodenjem dialoga in ravniyo sodelujočega odgovaranja** (DM).

Korelacijski koeficienti in p -vrednosti, ki glede na DM pripadajo izbranim regresijskim parametrom, so podani v tabeli 4.7. Spet smo uporabili Studentov test z $n-2$ prostostnimi stopnjami [Hladnik-2002] in ničelno hipotezo o nekoreliranosti, kjer je n velikost učne množice, tj. $n = 73$ v prvem eksperimentu in $n = 68$ v drugem eksperimentu. Polja, ki vsebujejo značilne korelacijske koeficiente ($p < 0.05$), smo potemnil.

V prvem eksperimentu je z neodvisno spremenljivko DM značilno koreliralo 10 parametrov (in sicer MUM, Comp, NUI, NIR, IRR, NGD, GDR, NRD, NND, NDR), en več kot z US, v drugem pa 8 parametrov (in sicer κ , MET, MUM, IRR, CMR, GDR, RDR, ARR), štirje več kot z US. Trije parametri so zastopani v obeh množicah, namreč MUM, IRR in GDR. Iz teh množic moramo spet odstraniti parametre, ki bi lahko povzročili multikolinearnost modela (razdelek 3.3.2).

Po postopku vzvratne eliminacije za $F_{out} = 4$ pri p -vrednosti približno enaki 0.05 (razdelek 3.3.3) na celotni učni množici, pridobljeni v prvem eksperimentu Čarovnik iz Oza, pri čemer vzamemo DM kot odvisno spremenljivko,

\rightsquigarrow **povprečno število potez** (MUM),

	WOZ1		WOZ2	
	kor. koef.	<i>p</i> -vrednost	kor. koef.	<i>p</i> -vrednost
Kappa koeficient (κ)	0.10	0.413	0.36	0.003
povprečni čas dialoga (MET)	0.14	0.228	-0.33	0.006
povprečno število potez (MUM)	0.31	0.007	-0.27	0.024
izpolnitev naloge (Comp)	0.36	0.002	0.19	0.124
število uporabnikovih iniciativ (NUI)	-0.54	0.000	0.12	0.347
povprečno število besed (MWT)	-0.09	0.474	-0.05	0.677
povprečni čas odziva (MRT)	-0.13	0.269	-0.14	0.266
število manjkajočih odzivov (NMR)	-0.16	0.182	-0.11	0.383
število neprimernih iniciativ (NUR)	-0.17	0.153	-0.08	0.512
delež neprimernih iniciativ (URR)	0.03	0.785	-0.15	0.207
število neprimernih odzivov (NIR)	-0.35	0.002	-0.12	0.336
delež neprimernih odzivov (IRR)	-0.25	0.031	-0.31	0.010
število napak (Error)	-0.15	0.207	-0.01	0.950
število pomoči (NHM)	-0.16	0.172	-0.07	0.590
delež pomoči (HMR)	-0.00	0.969	-0.07	0.582
število preverjanj (NCM)*	-	-	-0.20	0.105
delež preverjanj (CMR)*	-	-	-0.38	0.001
število podanih informacij (NGD)	-0.40	0.000	0.20	0.094
delež podanih informacij (GDR)	0.34	0.004	0.25	0.039
število relevantnih informacij (NRD)	-0.33	0.005	-0.04	0.750
delež relevantnih informacij (RDR)	-0.02	0.866	-0.24	0.050
število nepodanih informacij (NND)	-0.57	0.000	-0.01	0.958
delež nepodanih informacij (NDR)	-0.31	0.009	0.03	0.792
število prekinjenih zahtev (NAR)	-0.10	0.423	-0.17	0.177
delež prekinjenih zahtev (ARR)	-0.10	0.407	-0.25	0.038

* V prvem eksperimentu čarovnik ni izvajal CHECK potez.

Tabela 4.7: Korelacijski koeficienti in *p*-vrednosti za regresijske parametre glede na DM v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.

- ↪ **izpolnitev naloge (Comp)**,
- ↪ **število neprimernih odzivov (NIR)**,
- ↪ **število relevantnih informacij (NRD)** in
- ↪ **število nepodanih informacij (NND)**

pa kot neodvisne spremenljivke, identificiramo 7 osamelcev. Ti vzorci, ki predstavljajo slabih 10% vseh vzorcev, po napaki napovedi dejanske vrednosti $\mathcal{N}(\text{DM})$ z dobljenim MLR modelom,

$$0.24 * \mathcal{N}(\text{Comp}) - 0.51 * \mathcal{N}(\text{NND}),$$

zelo negativno izstopajo. Velikost učne množice končnega MLR modela se zaradi odstranitve osamelcev zmanjša s 73 na 66.

Postopek vzratne eliminacije ponovimo na zmanjšani učni množici vzorcev. Tabela 4.8 podaja dobljene delne *F* statistike, pripadajoče koeficiente determinacije R^2 ter parametre, ki jih v posameznih korakih iz modela odstranimo. Postopek vzratne eliminacije ustavimo pred 4. korakom, ko delna *F* statistika preseže vrednost 4.

Iz začetnega MLR modela z vzratno eliminacijo odstranimo tri parametre, in sicer NIR, MUM in NRD. Funkcija učinkovitosti za prvi sistem Čarovnik iz Oza in odvisno spremenljivko DM_1 , ki se nanaša na podatke, pridobljene v prvem eksperimentu Ča-

	F_i	R^2	odstranjen parameter
poln model	-	0.59	-
1. korak ($i = 1$)	0.00	0.59	NIR
2. korak ($i = 2$)	0.21	0.59	MUM
3. korak ($i = 3$)	3.32	0.57	NRD
4. korak ($i = 4$)	9.01	0.51	Comp

Tabela 4.8: Tabela vzvratne eliminacije za prvi sistem Čarovnik iz Oza in odvisno spremenljivko DM.

rovnik iz Oza, je zato taka:

$$\widehat{\mathcal{N}}(\text{DM}_1) = 0.25 * \mathcal{N}(\text{Comp}) - 0.65 * \mathcal{N}(\text{NND})$$

Dobljena funkcija učinkovitosti pojasnjuje 57% variance, tj. $R^2 = 0.57$. Najizrazitejši parameter, ki negativno vpliva na DM_1 , je parameter podatkovne zbirke NND.

Postopek identifikacije osamelcev in izpeljave funkcije učinkovitosti ponovimo še za drugi sistem Čarovnik iz Oza. Po postopku vzvratne eliminacije za $F_{out} = 4$ pri p -vrednosti približno enaki 0.05 na celotni učni množici, pridobljeni v drugem eksperimentu Čarovnik iz Oza, pri čemer vzamemo DM kot odvisno spremenljivko,

- ↪ **Kappa koeficient** (κ),
- ↪ **povprečni čas dialoga** (MET),
- ↪ **delež preverjanj** (CMR),
- ↪ **delež podanih informacij** (GDR) in
- ↪ **delež prekinjenih zahtev** (ARR)

pa kot neodvisne spremenljivke, identificiramo 5 osamelcev. Ti vzorci predstavljajo dobrih 7% vseh vzorcev in po napaki napovedi dejanske vrednosti $\mathcal{N}(\text{DM})$ z dobljenim MLR modelom,

$$0.31 * \mathcal{N}(\kappa) - 0.33 * \mathcal{N}(\text{CMR}) + 0.30 * \mathcal{N}(\text{GDR}),$$

zelo negativno izstopajo. Velikost učne množice končnega MLR modela se zaradi odstranitve osamelcev zmanjša z 68 na 63.

Postopek vzvratne eliminacije ponovimo na zmanjšani učni množici vzorcev. Tabela 4.9 podaja dobljene delne F statistike, pripadajoče koeficiente determinacije R^2 ter parametre, ki jih v posameznih korakih odstranimo iz modela. Postopek vzvratne eliminacije ustavimo pred 3. korakom, ko delna F statistika preseže vrednost 4.

Iz začetnega MLR modela z vzvratno eliminacijo odstranimo dva parametra, in sicer MET in ARR. Funkcija učinkovitosti za drugi sistem Čarovnik iz Oza in odvisno spremenljivko DM_2 , ki se nanaša na podatke, pridobljene v drugem eksperimentu Čarovnik

	F_i	R^2	odstranjen parameter
poln model	-	0.48	-
1. korak ($i = 1$)	1.71	0.46	MET
2. korak ($i = 2$)	2.84	0.44	ARR
3. korak ($i = 3$)	12.59	0.32	κ

Tabela 4.9: Tabela vzvratne eliminacije za drugi sistem Čarovnik iz Oza in odvisno spremenljivko DM.

iz Oza, je zato taka:

$$\mathcal{N}(\widehat{DM_2}) = 0.36 * \mathcal{N}(\kappa) - 0.38 * \mathcal{N}(CMR) + 0.40 * \mathcal{N}(GDR)$$

Dobljena funkcija učinkovitosti pojasnjuje 44% variance, tj. $R^2 = 0.44$, in ima tri parametre – **Kappa koeficient** (κ) in **delež podanih informacij** (GDR), ki pozitivno vplivata na DM_2 , ter **delež preverjanj** (CMR), ki negativno vpliva na DM_2 .

Funkciji učinkovitosti $\mathcal{N}(\widehat{US_1})$ in $\mathcal{N}(\widehat{US_2})$ ne vsebujeta nobenega skupnega parametra. Še več, nobeden od parametrov, ki jih vsebuje $\mathcal{N}(\widehat{US_1})$, ni značilen za US_2 , in nobeden od parametrov, ki jih vsebuje $\mathcal{N}(\widehat{US_2})$, ni značilen za US_1 (tabela 4.4). Za funkciji učinkovitosti $\mathcal{N}(\widehat{DM_1})$ in $\mathcal{N}(\widehat{DM_2})$ to ne velja več. Res je, da nobeden od parametrov, ki jih vsebuje $\mathcal{N}(\widehat{DM_1})$, ni značilen za $\mathcal{N}(\widehat{DM_2})$. Obratno pa ni res. Parameter podatkovne zbirke GDR, ki ima zelo velik pozitiven vpliv na DM_2 , sicer ni vsebovan v $\mathcal{N}(\widehat{DM_1})$, je pa visoko (negativno) koreliran s parametrom podatkovne zbirke NND (tabela 4.2), tj. najmočnejšim (negativnim) parametrom funkcije $\mathcal{N}(\widehat{DM_1})$.

Analiza obeh funkcij učinkovitosti za DM omogoča vrednotenje učinkovitosti modula za vodenje dialoga, povezanega s sodelujočo podatkovno zbirko (razdelek 2.5):

- Edini parameter, ki nastopa v funkciji učinkovitosti za DM_2 in je statistično značilen tudi za DM_1 ($p < 0.004$), je parameter podatkovne zbirke **delež podanih informacij** (GDR). V funkciji učinkovitosti za DM_1 namesto GDR sicer nastopa parameter podatkovne zbirke **število nepodanih informacij** (NND), ki je z njim visoko negativno koreliran in hkrati bolj značilen za DM_1 ($p < 0.0005$). Torej, parametri podatkovne zbirke predstavljajo edino podobnost med funkcijama učinkovitosti obeh sistemov Čarovnik iz Oza. Ta ugotovitev kaže na izjemno pomembnost predstavitve znanja oz. zgradbe podatkovne zbirke sistema za dialog. Pridemo do spoznanja, da so parametri podatkovne zbirke nepogrešljivi pri vrednotenju učinkovitosti sistemov za dialog, še posebej pa pri vrednotenju učinkovitosti sistemov za podajanje informacij.
- Medtem ko je parameter podatkovne zbirke **število nepodanih informacij** (NND) v prvem eksperimentu pomembno (negativno) vplival na zadovoljstvo uporabnikov, je njegov (negativni) vpliv v drugem eksperimentu izjemno splahnel. Vemo že (tabela 4.1), da se je srednja vrednost parametra **število relevantnih informacij** (NRD) v drugem eksperimentu značilno povečala, srednja

vrednost NND pa zato značilno zmanjšala. Vse torej kaže na to, da zmanjšanje števila odzivov, s katerimi sistem uporabniku sporoča, da zahtevane informacije nima, hkrati pa mu ne ponudi nobenih dosegljivih, relevantnih informacij, negativno vpliva na zadovoljstvo uporabnika. Razvijalci sistemov za dialog morajo zato težiti k zmanjšanju števila takih odzivov oz. povečanju stopnje sodelujočega odgovarjanja, predvsem *relaksacije* (poglavje 8). Sklepamo lahko tudi, da strategija usmerjanja uporabnika k izbiri dosegljivih, relevantnih podatkov, ki je implementirana v modulu za samodejno vodenje dialoga, na zadovoljstvo uporabnikov ne vpliva negativno.

- Ugotovili smo že (razdelek 4.4), da so bili uporabniki v prvem eksperimentu bolj občutljivi na kvantitativne parametre, uporabniki v drugem eksperimentu pa na proporcionalne parametre. Funkcija učinkovitosti za DM_1 vsebuje, poleg parametra Comp, še kvantitativni parameter **število nepodanih informacij** (NND). Funkcija učinkovitosti za DM_2 pa vsebuje, poleg parametra κ , še dva proporcionalna parametra, namreč **delež preverjanj** (CMR) in **delež podanih informacij** (GDR). Menimo, da je to posledica konsistentno povečanega ponujanja relevantnih informacij v drugem eksperimentu, ki je vodilo do več novih informacijskih iger in s tem do večje dojemljivosti uporabnikov za proporcionalne količine. Vsekakor so glede tega potrebne nadaljnje raziskave.
- Omenili smo že, da sta bila parametra **Kappa koeficient** (κ) in **izpolnitev naloge** (Comp) v naših eksperimentih nekorelirana. V prvem eksperimentu je na zadovoljstvo uporabnikov DM_1 močno (pozitivno) vplival Comp, κ ni imel statistično značilnega vpliva. V drugem eksperimentu je bilo ravno obratno – na zadovoljstvo uporabnikov DM_2 je močno (pozitivno) vplival κ , Comp pa ni imel statistično značilnega vpliva. Ugotovitev, do katere so prišle Walker, Litman, Kamm in Abella [Walker-1998], da **izpolnitev naloge** (Comp) močnejše vpliva na zadovoljstvo uporabnika kot **Kappa koeficient** (κ), torej ni vedno resnična. Le parameter Comp, katerega vrednost mora posredovati uporabnik, za vrednotenje učinkovitosti sistemov za dialog zato ni dovolj. Še vedno je dobro meriti tudi κ , ki pa ga na žalost ni mogoče določiti samodejno.
- Parameter, ki na zadovoljstvo uporabnikov DM_2 najmočnejše negativno vpliva, je **delež preverjanj** (CMR). Sistem za dialog lahko torej izboljšamo, če zmanjšamo delež potez tipa CHECK, ki preverjajo točnost podatkov, pridobljenih na osnovi zgodovine dialoga, ki jih uporabnik v svoji izjavi ne poda ali jih sistem ne razume. Vpliv parametra CMR v sistemih za dialog ni mogoče popolnoma odpraviti, zato ker je določeno število CHECK potez nujno vsakič, ko imamo opravka s samodejnim razumevanjem govora. Napake, ki se pojavljajo pri samodejnem razumevanju govora, sistem namreč prisilijo, da svoje razumevanje uporabnikovih izjav preveri vsakič, ko o njihovi pravilnosti ni popolnoma prepričan. Če tega ne bi počel, bi nekontrolirano podajal napačne odgovore. To bi povečalo srednjo vrednost parametra **delež neprimernih odzivov** (IRR) in tako zelo verjetno vodilo do večjega nezadovoljstva s sistemom.

Potem ko smo za odvisno spremenljivko vzeli **zadovoljstvo uporabnika z vodenjem dialoga in ravniyo sodelujočega odgovarjanja** (DM), nam je uspelo razliko

v natančnosti funkcij učinkovitosti obeh sistemov Čarovnik iz Oza izjemno zmanjšati ($R^2 = 0.57$ proti $R^2 = 0.44$). Upravičeno lahko torej trdimo, da se da DM veliko bolje modelirati kot US.

4.5.3 Zadovoljstvo uporabnikov kot vrednost posameznih ocen

Zanimivo bi bilo videti tudi, kako dobro se dajo modelirati posamezne ocene uporabnikov in ne nazadnje, kateri parametri na njih najmočneje vplivajo. Izpeljimo torej funkcije učinkovitosti obeh sistemov Čarovnik iz Oza za vseh osem ocen, dodeljenih osmim vprašanjem (tabela 3.1), s katerimi so uporabniki ocenjevali učinkovitost sistemov. Postopek vzvratne eliminacije bomo izvajali pod enakimi pogoji kot prej, le da iz učne množice ne bomo odstranjevali osamelcev. Natančnost tako dobljenih modelov namreč ne bo tako zelo pomembna, saj ne bodo primerni za vrednotenje učinkovitosti sistemov za dialog (razdelek 3.3.4), ampak le za pridobivanje osnovnih podatkov o sposobnosti modeliranja posameznih ocen in o statistično najznačilnejših parametrih.

Tabeli 4.10 in 4.11 vsebujeta korelacijske koeficiente regresijskih parametrov glede na posamezne ocene uporabnikov ter pripadajoče p -vrednosti, po vrsti označene takole:

1. TTS (iz angl. *Text-To-Speech*),
2. ASR (iz angl. *Automatic Speech Recognition*),
3. TE (iz angl. *Task Ease*),
4. IP (iz angl. *Interaction Pace*),
5. UE (iz angl. *User Expertise*),
6. SR (iz angl. *System Response*),
7. EB (iz angl. *Expected Behaviour*) in
8. FU (iz angl. *Future Use*).

Uporabili smo Studentov test z $n - 2$ prostostnimi stopnjami [Hladnik-2002] in ničelno hipotezo o nekoreliranosti, kjer je n velikost učne množice, tj. $n = 73$ v prvem eksperimentu in $n = 68$ v drugem eksperimentu. Polja, ki vsebujejo statistično značilne korelacijske koeficiente ($p < 0.05$), smo potemnil.

Najprej opazimo, da v drugem eksperimentu nobeden od parametrov ni bil statistično značilen za IP (tabela 4.11). Poleg tega imajo vse ocene, pridobljene v drugem eksperimentu, kvečjemu manj značilnih parametrov kot ocene, pridobljene v prvem eksperimentu. Zelo izrazit je podatek, da imajo ocene iz prvega eksperimenta v povprečju več kot dvakrat več statistično značilnih parametrov kot ocene iz drugega eksperimenta.

V postopku vzvratne eliminacije s TTS kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

	TTS		ASR		TE		IP		UE		SR		EB		FU	
	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p
κ	0.05	0.701	0.02	0.876	0.09	0.428	0.16	0.178	-0.10	0.392	0.14	0.252	0.04	0.707	-0.03	0.773
MET	-0.12	0.305	-0.21	0.079	-0.12	0.315	-0.15	0.194	-0.10	0.379	-0.07	0.512	-0.12	0.304	-0.09	0.463
MUM	0.27	0.021	0.23	0.050	0.22	0.059	0.24	0.037	0.41	0.000	0.21	0.073	0.33	0.005	0.23	0.050
Comp	0.27	0.019	0.42	0.000	0.32	0.006	0.20	0.082	0.08	0.520	0.31	0.007	0.07	0.583	-0.00	0.987
NUI	-0.53	0.000	-0.55	0.000	-0.32	0.006	-0.34	0.004	-0.58	0.000	-0.39	0.001	-0.45	0.000	-0.27	0.022
MWT	0.08	0.518	-0.11	0.342	-0.18	0.124	-0.02	0.843	0.09	0.453	0.02	0.854	0.01	0.935	0.01	0.943
MRT	-0.16	0.177	-0.10	0.384	-0.06	0.604	-0.24	0.045	-0.15	0.208	-0.11	0.374	-0.15	0.213	-0.10	0.399
NMR	-0.14	0.244	-0.16	0.166	-0.18	0.118	-0.05	0.659	-0.30	0.011	0.05	0.696	-0.21	0.073	0.06	0.598
NUR	-0.13	0.284	-0.14	0.247	-0.27	0.022	0.04	0.736	-0.17	0.162	0.03	0.787	-0.16	0.174	-0.11	0.374
URR	0.01	0.927	0.07	0.569	-0.14	0.253	0.27	0.022	0.00	0.993	0.15	0.219	0.03	0.781	0.07	0.570
NIR	-0.39	0.001	-0.37	0.001	-0.21	0.076	-0.18	0.117	-0.27	0.022	-0.30	0.011	-0.25	0.035	0.04	0.731
IRR	-0.19	0.099	-0.23	0.049	-0.21	0.078	-0.07	0.551	-0.11	0.358	-0.19	0.107	-0.16	0.166	0.12	0.311
Error	-0.12	0.312	-0.21	0.070	-0.10	0.411	-0.02	0.838	-0.28	0.016	0.04	0.712	-0.13	0.271	-0.16	0.183
NHM	-0.22	0.068	-0.10	0.407	-0.24	0.045	-0.16	0.179	-0.15	0.202	0.01	0.901	-0.19	0.103	0.12	0.322
HMR	-0.05	0.691	0.07	0.538	-0.14	0.224	-0.01	0.929	-0.06	0.632	0.16	0.169	-0.11	0.367	0.11	0.355
NGD	-0.39	0.001	-0.44	0.000	-0.19	0.110	-0.25	0.032	-0.50	0.000	-0.31	0.008	-0.36	0.002	-0.24	0.044
GDR	0.31	0.008	0.22	0.064	0.37	0.001	0.24	0.044	0.21	0.079	0.24	0.044	0.22	0.065	0.11	0.367
NRD	-0.30	0.010	-0.29	0.012	-0.25	0.033	-0.25	0.035	-0.19	0.117	-0.27	0.022	-0.22	0.067	-0.10	0.402
RDR	0.01	0.958	0.04	0.757	-0.05	0.686	-0.07	0.581	0.10	0.418	-0.08	0.477	0.05	0.682	0.00	0.967
NND	-0.61	0.000	-0.57	0.000	-0.36	0.002	-0.35	0.002	-0.61	0.000	-0.41	0.000	-0.47	0.000	-0.28	0.018
NDR	-0.33	0.004	-0.22	0.062	-0.29	0.012	-0.22	0.059	-0.27	0.019	-0.22	0.064	-0.22	0.058	-0.13	0.281
NAR	0.02	0.862	-0.08	0.514	-0.20	0.093	0.02	0.886	0.03	0.812	0.07	0.561	-0.09	0.430	0.06	0.643
ARR	0.04	0.722	-0.08	0.504	-0.22	0.059	0.06	0.635	0.04	0.722	0.08	0.501	-0.09	0.456	0.06	0.631

Tabela 4.10: Korelacijski koeficienti in pripadajoče p -vrednosti za regresijske parametre glede na posamezne ocene uporabnikov v prvem eksperimentu Čarovnik iz Oza.

	TTS		ASR		TE		IP		UE		SR		EB		FU	
	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p	kor.	p
κ	0.15	0.210	0.28	0.019	0.24	0.050	0.12	0.348	0.04	0.723	0.34	0.004	0.25	0.043	0.05	0.693
MET	0.04	0.745	-0.10	0.439	-0.12	0.314	0.04	0.764	-0.01	0.920	0.04	0.756	0.01	0.967	0.12	0.347
MUM	-0.35	0.004	-0.24	0.045	-0.08	0.521	-0.02	0.858	-0.21	0.080	-0.31	0.009	-0.22	0.068	-0.22	0.068
Comp	0.09	0.469	0.08	0.525	0.15	0.209	0.03	0.829	0.24	0.050	0.09	0.446	0.25	0.036	0.16	0.184
NUI	0.13	0.303	0.05	0.681	0.13	0.288	-0.18	0.140	0.08	0.524	0.08	0.542	0.09	0.451	0.27	0.025
MWT	0.13	0.288	0.02	0.891	-0.21	0.081	-0.11	0.351	0.12	0.343	0.08	0.511	-0.03	0.834	0.11	0.375
MRT	-0.10	0.431	-0.22	0.078	-0.09	0.441	0.03	0.825	-0.09	0.466	-0.10	0.411	-0.02	0.850	-0.08	0.504
NMR	-0.07	0.574	-0.24	0.045	0.04	0.746	0.12	0.311	-0.00	0.969	-0.03	0.788	-0.13	0.273	0.02	0.872
NUR	-0.13	0.284	-0.16	0.193	0.05	0.670	-0.13	0.294	-0.15	0.238	-0.07	0.586	-0.10	0.436	-0.03	0.778
URR	-0.10	0.440	-0.11	0.353	-0.03	0.809	-0.11	0.374	-0.15	0.217	-0.14	0.247	-0.20	0.097	-0.10	0.404
NIR	-0.09	0.473	-0.20	0.110	0.08	0.496	-0.02	0.858	0.05	0.693	-0.23	0.058	-0.04	0.721	0.10	0.431
IRR	-0.15	0.222	-0.28	0.020	-0.10	0.405	-0.01	0.934	0.01	0.933	-0.42	0.000	-0.16	0.205	-0.08	0.527
Error	0.07	0.556	-0.01	0.912	0.06	0.600	-0.00	0.970	0.06	0.601	-0.07	0.587	-0.01	0.919	0.18	0.152
NHM	-0.22	0.077	-0.19	0.121	0.09	0.469	0.12	0.323	-0.22	0.071	-0.04	0.766	-0.10	0.395	-0.00	0.994
HMR	-0.22	0.072	-0.21	0.092	0.09	0.449	0.18	0.149	-0.29	0.016	-0.02	0.885	-0.12	0.329	-0.04	0.718
NCM	-0.12	0.317	-0.19	0.130	-0.05	0.696	-0.07	0.588	-0.17	0.159	-0.24	0.046	-0.15	0.232	0.06	0.630
CMR	-0.27	0.027	-0.30	0.012	-0.19	0.119	-0.02	0.876	-0.31	0.010	-0.44	0.000	-0.26	0.036	-0.18	0.138
NGD	0.13	0.305	0.07	0.571	0.20	0.094	-0.20	0.102	0.06	0.645	0.15	0.233	0.20	0.106	0.24	0.048
GDR	0.18	0.139	0.07	0.576	0.29	0.016	0.01	0.927	0.01	0.918	0.16	0.195	0.23	0.058	0.11	0.390
NRD	0.05	0.691	-0.07	0.572	-0.02	0.889	0.03	0.784	0.09	0.443	0.02	0.885	-0.06	0.606	0.11	0.393
RDR	-0.20	0.096	-0.22	0.074	-0.16	0.184	0.03	0.792	-0.03	0.786	-0.12	0.340	-0.25	0.038	-0.19	0.129
NND	0.05	0.685	0.13	0.301	-0.01	0.926	-0.14	0.252	0.01	0.906	-0.06	0.650	-0.06	0.620	0.24	0.048
NDR	-0.05	0.705	0.22	0.076	-0.14	0.268	-0.10	0.416	0.03	0.789	0.04	0.762	0.02	0.843	0.19	0.116
NAR	-0.01	0.933	-0.12	0.310	-0.09	0.447	-0.02	0.895	-0.04	0.752	-0.20	0.099	-0.09	0.477	-0.20	0.104
ARR	0.07	0.556	-0.17	0.171	-0.18	0.145	0.12	0.327	-0.03	0.796	-0.29	0.016	-0.13	0.287	-0.32	0.009

Tabela 4.11: Korelacijski koeficienti in pripadajoče p -vrednosti za regresijske parametre glede na posamezne ocene uporabnikov v drugem eksperimentu Čarovnik iz Oza.

$$\begin{aligned}\widehat{\mathcal{N}(\text{TTS}_1)} &= -0.16 * \mathcal{N}(\text{NIR}) - 0.54 * \mathcal{N}(\text{NND}) \\ \widehat{\mathcal{N}(\text{TTS}_2)} &= -0.28 * \mathcal{N}(\text{MUM}) - 0.14 * \mathcal{N}(\text{CMR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnujeta 39% variance v prvem in 14% variance v drugem eksperimentu Čarovnik iz Oza.

V postopku vzratne eliminacije z ASR kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

$$\begin{aligned}\widehat{\mathcal{N}(\text{ASR}_1)} &= 0.31 * \mathcal{N}(\text{Comp}) - 0.50 * \mathcal{N}(\text{NND}) \\ \widehat{\mathcal{N}(\text{ASR}_2)} &= 0.29 * \mathcal{N}(\kappa) - 0.25 * \mathcal{N}(\text{NMR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnujeta 41% variance v prvem in 15% variance v drugem eksperimentu Čarovnik iz Oza.

V postopku vzratne eliminacije s TE kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

$$\begin{aligned}\widehat{\mathcal{N}(\text{TE}_1)} &= 0.25 * \mathcal{N}(\text{Comp}) - 0.30 * \mathcal{N}(\text{NUR}) \\ \widehat{\mathcal{N}(\text{TE}_2)} &= 0.27 * \mathcal{N}(\kappa) + 0.32 * \mathcal{N}(\text{GDR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnujeta 19% variance v prvem in 16% variance v drugem eksperimentu Čarovnik iz Oza.

Vzratno eliminacijo z IP kot odvisno spremenljivko lahko izvajamo le za prvi eksperiment Čarovnik iz Oza, saj v drugem eksperimentu ni bilo statistično značilnih parametrov. Dobimo funkcijo učinkovitosti

$$\widehat{\mathcal{N}(\text{IP}_1)} = 0.31 * \mathcal{N}(\text{URR}) - 0.39 * \mathcal{N}(\text{NND}),$$

ki pojasnjuje 22% variance.

V postopku vzratne eliminacije z UE kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

$$\begin{aligned}\widehat{\mathcal{N}(\text{UE}_1)} &= -0.18 * \mathcal{N}(\text{NMR}) - 0.57 * \mathcal{N}(\text{NND}) \\ \widehat{\mathcal{N}(\text{UE}_2)} &= 0.24 * \mathcal{N}(\text{Comp}) - 0.27 * \mathcal{N}(\text{HMR}) - 0.31 * \mathcal{N}(\text{CMR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnujeta 40% variance v prvem in 23% variance v drugem eksperimentu Čarovnik iz Oza.

V postopku vzratne eliminacije z SR kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

$$\begin{aligned}\widehat{\mathcal{N}(\text{SR}_1)} &= 0.23 * \mathcal{N}(\text{Comp}) - 0.36 * \mathcal{N}(\text{NND}) \\ \widehat{\mathcal{N}(\text{SR}_2)} &= -0.33 * \mathcal{N}(\text{IRR}) - 0.36 * \mathcal{N}(\text{CMR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnjujeta 22% variance v prvem in 30% variance v drugem eksperimentu Čarovnik iz Oza.

V postopku vzvratne eliminacije z EB kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

$$\begin{aligned}\widehat{\mathcal{N}}(\widehat{\text{EB}}_1) &= 0.15 * \mathcal{N}(\text{MUM}) - 0.41 * \mathcal{N}(\text{NND}) \\ \widehat{\mathcal{N}}(\widehat{\text{EB}}_2) &= 0.26 * \mathcal{N}(\text{Comp}) - 0.26 * \mathcal{N}(\text{CMR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnjujeta 24% variance v prvem in 13% variance v drugem eksperimentu Čarovnik iz Oza.

V postopku vzvratne eliminacije z FU kot odvisno spremenljivko dobimo naslednji funkciji učinkovitosti:

$$\begin{aligned}\widehat{\mathcal{N}}(\widehat{\text{FU}}_1) &= 0.13 * \mathcal{N}(\text{MUM}) - 0.22 * \mathcal{N}(\text{NND}) \\ \widehat{\mathcal{N}}(\widehat{\text{FU}}_2) &= 0.26 * \mathcal{N}(\text{NUI}) - 0.30 * \mathcal{N}(\text{ARR})\end{aligned}$$

Dobljeni funkciji učinkovitosti pojasnjujeta 9% variance v prvem in 16% variance v drugem eksperimentu Čarovnik iz Oza.

Spet opazimo, da so skoraj vse funkcije učinkovitosti za prvi sistem Čarovnik iz Oza natančnejše od pripadajočih funkcij učinkovitosti za drugi sistem Čarovnik iz Oza. Izjema sta le funkciji učinkovitosti za SR in FU. Najslabše izmed vseh se data modelirati IP in FU. Hitrost interakcije s sistemom (IP) je bila močno odvisna od čarovnika in zato precej nepredvidljiva. Nesposobnost modeliranja ocene FU pa je v skladu z našo že omenjeno trditvijo, da pozitivne izkušnje uporabnikov v testiranjih sistemov za dialog zelo verjetno niso kazalec njihovih prihodnjih navad.

Kljub pričakovanjem je dejstvo, da je bil parameter **število nepodanih informacij** (NND), ki je bil v prvem eksperimentu statistično najznačilnejši parameter sedmih izmed osmih ocen uporabnikov (tabela 4.10) in hkrati parameter z absolutno največjo (sicer negativno) utežjo sedmih izmed osmih funkcij učinkovitosti, neznačilen za prvih sedem ocen v drugem eksperimentu, nekoliko presenetljivo. Tudi za zadnjo oceno, namreč FU, je bil na meji statistične značilnosti in ne nastopa v njeni funkciji učinkovitosti. Razen tega je večina najznačilnejših parametrov posameznih ocen iz drugega eksperimenta statistično neznačilna za isto oceno iz prvega eksperimenta. To je res tudi, če izločimo parameter **delež preverjanj** (CMR), ki ga v prvem eksperimentu ni bilo. Na te ugotovitve lahko gledamo kot na potrditev neprimernosti posamičnih vprašanj za merjenje uporabnikovega zadovoljstva z učinkovitostjo sistemov za dialog (razdelek 3.3.4). Po drugi strani pa bi to lahko pomenilo, da odgovori na izbrana vprašanja (tabela 3.1) kažejo le velike razlike med obema sistemoma.

V prvem eksperimentu izstopa tudi podobnost funkcij učinkovitosti za EB in FU na eni strani ter ASR, TE in SR na drugi strani. Funkciji učinkovitosti za EB in FU se v drugem eksperimentu močno razlikujeta. Trdimo lahko celo, da se ne bi mogli razlikovati bolj – množici njunih značilnih parametrov sta disjunktni. Drugače pa je zo

ASR, TE in SR. Z analizo korelacij med parametri tudi v drugem eksperimentu opazimo podobnosti med njihovimi funkcijami učinkovitosti. Zanimivo je, da predstavlja ta trojica tričetrť parametra **zadovoljstvo uporabnika z vodenjem dialoga in ravnijo sodelujočega odgovarjanja** (DM). To je najbrž eden od vzrokov za zmanjšanje razlike v natančnosti funkcij učinkovitosti obeh sistemov Čarovnik iz Oza, ki smo jo dosegli, ko smo odvisno spremenljivko US zamenjali z DM.

Tekoče poglavje in v njem opažene razlike med dobljenimi funkcijami učinkovitosti nas pripeljejo do sklepa, da so potrebne nadaljnje raziskave. Potreba po razvoju veljavnega psihometričnega vprašalnika o zadovoljstvu uporabnikov ni edina potreba, ki se pojavi v študiji vrednotenja učinkovitosti sistemov za dialog. Treba je tudi razumeti in biti sposoben pravilno opredeliti razlike med različnimi funkcijami učinkovitosti – pa naj bo to med funkcijami, ki se nanašajo na več različic danega sistema za dialog, alisi med funkcijami, ki se nanašajo na različne sisteme ali celo različne načine merjenja zadovoljstva uporabnikov.

5

Relacijski podatkovni model

5.1 Uvod

5.2 Relacijska algebra

Predstavimo najpopularnejši model podatkovnih zbirk, tj. relacijski podatkovni model, katerega matematična osnova je relacija. Pri tem se osredotočimo na jezik proceduralne relacijske algebre.

Definiramo pet osnovnih operacij relacijske algebre – unijo, razliko, kartezični produkt, projekcijo in izbiro ter opišemo izpeljane operacije relacijske algebre – presek, kvocient, spoj, naravni spoj in semispoj.

Navedemo vrsto algebrajskih zakonov, ki se jim podrejajo definirane operacije relacijske algebre in ki imajo pomembno vlogo pri optimizaciji poizvedb.

5.1 Uvod

Vse podatkovne zbirke temeljijo na matematični strukturi, imenovani *podatkovni model*, ki zaobsega:

1. matematični jezik za opis podatkov in relacij med njimi ter
2. tehniko upravljanja s podatki, ki vključuje odgovarjanje na poizvedbe in preverjanje integritete podatkovne zbirke.

Relacijski podatkovni model, ki ga je leta 1970 vpeljal E. F. Codd [Codd-1970], je postal daleč najpopularnejši model podatkovnih zbirk. Najverjetnejši vzrok za tako premoč nad drugimi podatkovnimi modeli [Tsichritzis-1982, Ullman-1988, Ramakrishnan-2003, Elmasri-2004] je izredno močan, preprost proceduralen jezik za opis operacij nad podatki, ki ga relacijski podatkovni model zajema (razdelek 5.2).

Matematična osnova relacijskega podatkovnega modela je *relacija*, torej podmnožica kartezičnega produkta ene ali več vrednostnih množic. Vrednostna množica je lahko množica naravnih števil, množica realnih števil, množica nizov, množica $\{0, 1\}$ in podobno. Relacija je n -mestna, če so njeni elementi urejene n -terice. Ko imamo opravka s podatkovnimi zbirkami, se omejimo na končne relacije in relacijo predstavimo s tabelo, v kateri vsaki n -terici pripada natanko ena vrstica, istoležne komponente n -teric pa ležijo v istem stolpcu. Codd je v svojem legendarnem članku [Codd-1970] lastnosti tabele, ki predstavlja n -mestno relacijo R , opisal takole:

1. Vsaka vrstica predstavlja eno n -terico relacije R .
2. Vrstni red vrstic ni pomemben.
3. Vrstice so paroma različne.
4. Vrstni red stolpcev je pomemben – ustreza vrstnemu redu vrednostnih množic, na katerih je definirana relacija R .
5. Pomembnost vrstnega reda stolpcev delno izrazimo tako, da jim priredimo imena.

Stolpcem tabel torej pogosto priredimo imena, ki jih imenujemo *atributi*. Če so stolpcem dane relacije prirejeni atributi, vrstni red stolpcev postane nepomemben – ločimo jih na osnovi atributov. Na atribut lahko gledamo kot na preslikavo, ki slika objekte, opisane z n -tericami, v pripadajočo vrednostno množico. Rekli bomo, da so posamezne komponente dane n -terice vrednosti pripadajočih atributov. Množico atributov dane relacije imenujemo *relacijska shema*. Če stolpcem n -mestne relacije R priredimo attribute A_1, A_2, \dots, A_n , njeno relacijsko shemo zapišemo kot $R(A_1, A_2, \dots, A_n)$. Atribut A_i relacije R pogosto imenujemo tudi $R.A_i$.

Za konstante uporabljamo nize malih črk, za attribute in relacije pa nize velikih črk.

Primer 5.1. Tabela na sliki 5.1 prikazuje relacijo INFO, katere atributi so MESTO, REGIJA in PREBIVALSTVO. Relacija INFO vsebuje pet elementov, namreč urejene trojke (ljubljana, osrednjeslovenska regija, 265 881), (maribor, štajerska, 110 668), (celje, štajerska, 48 081), (novo mesto, dolenska, 40 925) in (murska sobota, pomurje, 20 080). Njena relacijska shema je $INFO(MESTO, REGIJA, PREBIVALSTVO)$.

MESTO	REGIJA	PREBIVALSTVO
ljubljana	osrednjeslovenska regija	265 881
maribor	štajerska	110 668
celje	štajerska	48 081
novo mesto	dolenska	40 925
murska sobota	pomurje	20 080

Slika 5.1: Relacija INFO. Podatki o številu prebivalcev so povzeti po rezultatih popisa prebivalstva, ki ga je leta 2002 opravil Statistični urad Republike Slovenije.

Elementi dane n -mestne relacije so paroma različni. Podmnožici atributov, ki zadoščajo za medsebojno razlikovanje n -teric, pravimo *ključ relacijske sheme*. Celoten postopek načrtovanja relacijskih podatkovnih zbirk, vključno z ugotavljanjem funkcionalnih odvisnosti v množici atributov in določanjem ključev, je podrobno opisan v [Mohor-1997, Mata-Toledo-2000].

Operacije, ki jih izvajamo nad relacijami v relacijskem podatkovnem modelu, lahko opišemo na dva precej različna, a po izrazni moči enaka [Ullman-1988] načina:

1. algebrski način, ki porodi proceduralno *relacijsko algebro*, kjer poizvedbe zapišemo kot zaporedje operacij nad relacijami, in
2. logični način, ki porodi deklarativen *relacijski račun*, kjer poizvedbe zapišemo kot logične formule, ki naj jim iskani odgovor zadošča.

V razdelku 5.2 bomo obravnavali relacijsko algebro, ki predstavlja enega izmed jezikov relacijskega podatkovnega modela.

5.2 Relacijska algebra

Relacijska algebra, kakor tudi vsi drugi relacijski podatkovni modeli, zajema po izrazni moči omejeno podmnožico vseh možnih poizvedb z naslednjimi lastnostmi:

- ↪ omogoča zadovoljivo reševanje težavnega optimizacijskega problema [Ullman-1989, Chaudhuri-1998] in
- ↪ ima dovolj bogat jezik, na katerem sloni uporabnost podatkovnih sistemov.

Relacijski jeziki pa na primer niso sposobni izraziti operacije, ki bi dvomestni relaciji priredila njeno tranzitivno zaprtje. Naslednja, a ne tako bistvena, omejitev relacijskih jezikov je končnost relacij. Ta zahteva je potrebna, zato ker neskončnih relacij ni mogoče fizično shraniti. Eno izmed omejitev definicije relacijske algebre in drugih relacijskih jezikov, ki se pojavi zaradi zahteve po končnosti relacij, predstavlja algebrajska operacija komplement. Te operacije ne smemo dovoliti, saj je komplement končne relacije R , definirane na vsaj eni neskončni vrednostni množici, v splošnem lahko neskončna relacija. Namesto komplementa zato uporabljamo *razliko* ali *relativni komplement* (razdelek 5.2.1).

5.2.1 Osnovne operacije relacijske algebre

Operandi relacijske algebre so lahko konstantne relacije ali spremenljivke, ki nastopajo namesto relacij določene dimenzije. Ponavadi definiramo pet osnovnih operacij relacijske algebre – *unijo*, *razliko*, *kartezični produkt*, *projekcijo* in *izbiro*:

1. *Unija* n -mestnih relacij R in S , ki jo označimo z $R \cup S$, je unija množic R in S . To je množica vseh n -teric, ki so vsaj v eni izmed množic R in S . Attribute danih relacij izpustimo, uniji pa lahko priredimo poljubne (nove) attribute.
2. *Razlika* ali *relativni komplement* n -mestnih relacij R in S , ki jo označimo z $R - S$, je razlika množic R in S . To je množica vseh n -teric iz R , ki niso elementi S . Attribute danih relacij spet izpustimo, razliki pa lahko priredimo poljubne (nove) attribute.
3. *Kartezični produkt* r -mestne relacije R in s -mestne relacije S , ki ga označimo z $R \times S$, je kartezični produkt množic R in S . To je množica vseh $(r + s)$ -teric, kjer prvih r komponent tvori r -terico iz R , zadnjih s komponent pa s -terico iz S .
4. *Projekcija* n -mestne relacije R na paroma različne komponente z zaporednimi številkami i_1, i_2, \dots, i_k , tj. naravnimi števili iz množice $\{1, 2, \dots, n\}$, je množica vseh k -teric (a_1, a_2, \dots, a_k) , za katere obstaja n -terica (b_1, b_2, \dots, b_n) v R , da je $a_j = b_{i_j}$ za $j = 1, 2, \dots, k$. Projekcijo relacije R na komponente i_1, i_2, \dots, i_k označimo s $\pi_{i_1, i_2, \dots, i_k}(R)$. Namesto zaporednih števil izbranih komponent lahko pišemo tudi imena njim pripadajočih atributov. Projekcija omogoča permutacijo in/ali izbris izbranih stolpcev dane relacije.
5. *Izbira* temelji na formuli F , ki vsebuje:
 - konstantne relacije in zaporedne številke komponent, kjer je i -ta komponenta zapisana kot $\$i$;
 - aritmetične operatorje primerjav: $=$, $<$, $>$, \neq , \leq in \geq ;
 - logične operatorje: \wedge , \vee in \neg .

V nadaljevanju poglavja bodo vse formule ustrezale naštetim lastnostim. Tako kot pri projekciji namesto zaporednih števil v F nastopajočih komponent lahko pišemo imena njim pripadajočih atributov. *Izbira* n -teric relacije R , ki zadošča

formuli F , je tedaj množica vseh n -teric iz R , katerih komponente zadoščajo formuli F . Izbiro n -teric relacije R , ki zadošča formuli F , označimo s $\sigma_F(R)$.

Primer 5.2. *Vzemimo relaciji R in S , definirani s tabelama:*

A	B	C
a	b	c
d	a	f
c	b	d

Relacija R

D	E	F
b	g	a
d	a	f

Relacija S

Unija $R \cup S$ in razlika $R - S$ relacij R in S sta tedaj podani z naslednjima tabelama:

a	b	c
d	a	f
c	b	d
b	g	a
d	a	f

$R \cup S$

a	b	c
c	b	d

$R - S$

Operaciji unije in razlike lahko uporabimo na relacijah R in S kljub različnim atributom stolpcev obeh relacij. Pogoj je le enaka dimenzija relacij.

Kartezični produkt $R \times S$ relacij R in S prikazuje naslednja tabela:

A	B	C	D	E	F
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f

$R \times S$

Ker sta množici atributov relacij R in S disjunktni, lahko te prenesemo na relacijo $R \times S$. Če pa bi se R in S v katerem atributu ujemale, na primer v G , bi pripadajoča stolpca lahko poimenovali $R.G$ in $S.G$.

Naslednji tabeli podajata še projekcijo $\pi_{A,C}(R)$ in izbiro $\sigma_{B=b}(R)$:

A	C
a	c
d	f
c	d

$$\pi_{A,C}(R)$$

A	B	C
a	b	c
c	b	d

$$\sigma_{B=b}(R)$$

5.2.2 Izpeljane operacije relacijske algebre

Obstaja še vrsta drugih zanimivih operacij nad relacijami, ki jih lahko izrazimo z naštetimi petimi osnovnimi operacijami relacijske algebre. Opisali bomo le nekatere izmed njih – *presek*, *kvocient*, *spoj*, *naravni spoj* in *semispoj*.

Presek

Presek r -mestnih relacij R in S , ki ga označimo z $R \cap S$, je presek množic R in S . To je množica vseh r -teric, skupnih relacijama R in S . Attribute danih relacij izpustimo, preseku pa lahko priredimo poljubne (nove) attribute. Relacija $R \cap S$, je ekvivalentna relaciji

$$R - (R - S),$$

ki jo iz R in S dobimo z uporabo osnovnih operacij relacijske algebre [Ullman-1988]. Velja tudi

$$R - S = R \cap S^c,$$

kar pomeni, da bi za osnovno operacijo relacijske algebre namesto razlike lahko vzeli tudi presek. Tako bo v kategornih modelih, ki nimajo komplementov (poglavje 9).

Primer 5.3. *Vzemimo relaciji R in S iz primera 5.2. Presek $R \cap S$ relacij R in S prikazuje tabela:*

d	a	f
$R \cap S$		

Kvocient

Kvocient r -mestne relacije R in s -mestne relacije S , kjer je $r > s$ in $S \neq \emptyset$, je množica vseh $(r - s)$ -teric (a_1, \dots, a_{r-s}) , da je za vse s -terice (a_{r-s+1}, \dots, a_r) iz S r -terica (a_1, \dots, a_r) v R . Kvocient relacij R in S označimo z $R \div S$, in velja:

$$(a_1, \dots, a_{r-s}) \in R \div S \iff \forall (a_{r-s+1}, \dots, a_r) \in S : (a_1, \dots, a_r) \in R$$

Relacija $R \div S$ je ekvivalentna relaciji

$$\pi_{1,2,\dots,r-s}(R) - \pi_{1,2,\dots,r-s}((\pi_{1,2,\dots,r-s}(R) \times S) - R),$$

ki jo iz R in S dobimo z uporabo osnovnih operacij relacijske algebre [Ullman-1988].

Primer 5.4. Naj bosta relaciji R in S podani s tabelama:

a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

Relacija R

c	d
e	f

Relacija S

Kvocijent $R \div S$ relacij R in S prikazuje naslednja tabela:

a	b
e	d

$R \div S$

Par (a, b) je element relacije $R \div S$, ker sta (a, b, c, d) in (a, b, e, f) v R . Par (e, d) je element $R \div S$, ker sta v R (e, d, c, d) in (e, d, e, f) . Par (b, c) , ki je poleg že omenjenih edini, ki se pojavlja v prvih dveh stolpcih relacije R , pa ni element $R \div S$, ker (b, c, c, d) ni element relacije R .

Spoj

Naj bo θ poljuben aritmetični operator primerjav, torej $=, <, >, \neq, \leq$ ali \geq . θ -spoj relacij R in S na i -tem in j -tem stolpcu, ki ga označimo z

$$R \bowtie_{i\theta j} S,$$

je množica tistih elementov kartezičnega produkta $R \times S$, za katere je i -ta komponenta iz R v relaciji θ z j -to komponento iz S . Zaporedne številke stolpcev lahko nadomestimo z atributi, če so dani.

Relacija $R \bowtie_{i\theta j} S$, kjer je R r -mestna relacija, je ekvivalentna relaciji

$$\sigma_{\theta_{i\theta j}(r+j)}(R \times S),$$

ki jo iz R in S dobimo z uporabo osnovnih operacij relacijske algebre [Ullman-1988].

Primer 5.5. Naj bosta relaciji R in S podani s tabelama:

A	B	C
1	2	3
4	5	6
7	8	9

Relacija R

D	E
3	1
6	2

Relacija S

Tedaj je $R \bowtie_{B<D} S$ oz. $R \bowtie_{2<1} S$ naslednja relacija:

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

$$R \bowtie_{B<D} S$$

Naravni spoj

Naravni spoj relacij R in S , katerih stolpcem so prirejene atributi, dobimo tako, da najprej iz kartezičnega produkta $R \times S$ izberemo tiste elemente, katerih vrednosti skupnih atributov relacij R in S se ujemajo. To je, za vsak atribut A , ki ga vsebujeta relacija R in relacija S , se vrednosti stolpcev (relacije $R \times S$) z imenoma $R.A$ in $S.A$ ujemata. Nato za vsak tak atribut A izločimo stolpec $S.A$ in stolpec $R.A$ poimenujemo A . Naravni spoj relacij R in S označimo z $R \bowtie S$.

Naj bodo A_1, \dots, A_k vsi atributi, skupni relacijama R in S , i_1, \dots, i_m pa urejen seznam vseh atributov relacije $R \times S$ brez $S.A_1, \dots, S.A_k$. Tedaj je relacija $R \bowtie S$ ekvivalentna relaciji

$$\pi_{i_1, \dots, i_m} \sigma_{R.A_1=S.A_1 \wedge \dots \wedge R.A_k=S.A_k} (R \times S),$$

ki jo iz R in S dobimo z uporabo osnovnih operacij relacijske algebre [Ullman-1988].

Primer 5.6. Naj bosta relaciji R in S podani s tabelama:

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

Relacija R

B	C	D
b	c	d
b	c	e
a	d	b

Relacija S

Naravni spoj $R \bowtie S$ relacij R in S prikazuje naslednja tabela:

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

$$R \bowtie S$$

Naravni spoj $R \bowtie S$ dobimo tako, da za vsak element relacije R preverimo, s katerimi elementi iz S se ujema v stolpcih B in C . Ker se trojica (a, b, c) iz R ujema s trojicama (b, c, d) in (b, c, e) iz S , v $R \bowtie S$ dobimo četverki (a, b, c, d) in (a, b, c, e) . Podobno ugotovimo, da so v $R \bowtie S$ tudi (d, b, c, d) in (d, b, c, e) ter (c, a, d, b) . Ker pa se trojica (b, b, f) iz R v stolpcih B in C ne ujema z nobeno trojico iz S , v relaciji $R \bowtie S$ ni elementa, katerega prve tri komponente bi bile (b, b, f) .

Semispoj

Semispoj relacij R in S , ki ga označimo z $R \times S$, je projekcija naravnega spoja R in S na attribute A_1, \dots, A_n relacije R . To je

$$R \times S = \pi_{A_1, \dots, A_n}(R \bowtie S).$$

Pri tem pogosto, namesto da bi našteli vse attribute relacije, zapišemo kar ime relacije:

$$R \times S = \pi_R(R \bowtie S).$$

Relacija $R \times S$ je ekvivalentna relaciji

$$R \bowtie \pi_{R \cap S}(S),$$

ki jo iz R in S dobimo z uporabo osnovnih operacij relacijske algebre [Ullman-1988]. Pri tem smo s $\pi_{R \cap S}(S)$ označili projekcijo relacije S na skupne attribute relacij R in S .

Primer 5.7. *Vzemimo relaciji R in S iz primera 5.6. Semispoj $R \times S$ je projekcija relacije $R \bowtie S$ na attribute A, B in C . To je relacija, podana v naslednji tabeli:*

A	B	C
a	b	c
d	b	c
c	a	d

$$R \times S$$

5.2.3 Algebrajski zakoni

Operacije relacijske algebre se podrejajo vrsti algebrajskih zakonov [Ullman-1988]. Poglejmo najprej, katere izmed njih so asociativne oz. komutativne:

1. Unija je asociativna in komutativna operacija na relacijah:

$$R \cup (S \cup T) = (R \cup S) \cup T$$

$$R \cup S = S \cup R$$

2. Razlika ni niti komutativna niti asociativna operacija na relacijah.
3. Kartezični produkt je asociativna operacija na relacijah:

$$R \times (S \times T) = (R \times S) \times T$$

Kartezični produkt relacij v splošnem ni komutativen. Če pa so stolpcem relacije prirejeni atributi in njihov vrstni red ni pomemben, je kartezični produkt relacij tudi komutativen.

4. Projekcija po definiciji ne more biti komutativna. Za množico $\{A_1, \dots, A_n\}$, ki je podmnožica množice (ne nujno vseh) atributov $\{B_1, \dots, B_m\}$ relacije E , pa velja

$$\pi_{A_1, \dots, A_n}(\pi_{B_1, \dots, B_m}(E)) = \pi_{A_1, \dots, A_n}(E).$$

5. Izbira je komutativna operacija na relacijah:

$$\sigma_{F_1}(\sigma_{F_2}(E)) = \sigma_{F_2}(\sigma_{F_1}(E))$$

Velja tudi:

$$\sigma_{F_1}(\sigma_{F_2}(E)) = \sigma_{F_1 \wedge F_2}(E)$$

6. Presek je asociativna in komutativna operacija na relacijah:

$$R \cap (S \cap T) = (R \cap S) \cap T$$

$$R \cap S = S \cap R$$

7. Kvocient ni niti komutativna niti asociativna operacija na relacijah, saj se dimenziji relacij $R \div (S \div T)$ in $(R \div S) \div T$ ter dimenziji relacij $R \div S$ in $S \div R$ v splošnem ne ujemata.
8. Naj bosta θ_1 in θ_2 poljubna aritmetična operatorja primerjav, R naj bo r -mestna relacija, S s -mestna relacija, T pa poljubna relacija. Tedaj za $i, k, l \in \mathbb{N}$ in naravno število $j \leq s$ velja

$$R \underset{i\theta_1 j}{\bowtie} (S \underset{k\theta_2 l}{\bowtie} T) = (R \underset{i\theta_1 j}{\bowtie} S) \underset{(r+k)\theta_2 l}{\bowtie} T.$$

Ta zakon spominja na zakon o asociativnosti θ -spoja. Edina razlika je v spremembi indeksov stolpcev para relacij v drugem θ -spoju. Iz definicije θ -spoja pa sledi, da θ -spoj v splošnem ni komutativna operacija na relacijah.

9. Naj bodo stolpcem relacij R in S prirejeni atributi. Tedaj je naravni spoj asociativna in komutativna operacija na relacijah:

$$R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$$

$$R \bowtie S = S \bowtie R$$

Posledica je preprosta oblika naravnega spoja končne množice relacij. Relacija $R = R_1 \bowtie \dots \bowtie R_n$ je namreč množica natanko tistih elementov μ , katerih projekcija na attribute R_i je element relacije R_i za $i = 1, \dots, n$. Torej:

$$\mu \in R \iff \forall i = 1, \dots, n : \pi_{R_i}(\mu) \in R_i$$

Vrstni red spajanja relacij R_i pri tem ni pomemben.

10. Semispoj ni komutativna operacija na relacijah, saj sta $R \bowtie S = \pi_R(R \bowtie S)$ in $S \bowtie R = \pi_S(S \bowtie R) = \pi_S(R \bowtie S)$ projekciji elemenov $R \bowtie S$ na v splošnem različno število komponent, ki v splošnem ustrezajo tudi različnemu naboru atributov.

V nadaljevanju bomo navedli nekaj algebrajskih zakonov, ki se nanašajo na kompozicije in/ali kombinacije različnih, v glavnem osnovnih operacij relacijske algebre. Veljajo naslednji zakoni [Ullman-1989]:

1. Za operacijo izbire, ki temelji na formuli $F = F_1 \vee F_2$, velja naslednje:

$$\sigma_{F_1 \vee F_2}(E) = \sigma_{F_1}(E) \cup \sigma_{F_2}(E)$$

2. Naj bo F formula, ki vsebuje le attribute A_1, \dots, A_n . Tedaj velja

$$\pi_{A_1, \dots, A_n}(\sigma_F(E)) = \sigma_F(\pi_{A_1, \dots, A_n}(E)).$$

Če F vsebuje tudi attribute B_1, \dots, B_m , ki so vsi različni od A_1, \dots, A_n , velja

$$\pi_{A_1, \dots, A_n}(\sigma_F(E)) = \pi_{A_1, \dots, A_n}(\sigma_F(\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(E))).$$

3. Naj formula F vsebuje le attribute relacije E_1 . Tedaj velja

$$\sigma_F(E_1 \times E_2) = \sigma_F(E_1) \times E_2.$$

Če je $F = F_1 \wedge F_2$, kjer F_1 vsebuje le attribute relacije E_1 , F_2 pa le attribute relacije E_2 , velja

$$\sigma_F(E_1 \times E_2) = \sigma_{F_1}(E_1) \times \sigma_{F_2}(E_2).$$

Če v primeru $F = F_1 \wedge F_2$, formula F_1 vsebuje le attribute relacije E_1 , formula F_2 pa attribute relacije E_1 in attribute relacije E_2 , velja

$$\sigma_F(E_1 \times E_2) = \sigma_{F_2}(\sigma_{F_1}(E_1) \times E_2).$$

4. Naj se atributi relacije $E = E_1 \cup E_2$ ujemajo z atributi relacij E_1 in E_2 . Tedaj za formulo F velja

$$\sigma_F(E_1 \cup E_2) = \sigma_F(E_1) \cup \sigma_F(E_2).$$

Če se nabor atributov relacij E_1 in E_2 razlikuje od atributov relacije E , je v formulah, ki nastopata na desni strani enakosti, attribute treba ustrezno zamenjati.

5. Naj se atributi relacije $E = E_1 - E_2$ ujemajo z atributi relacij E_1 in E_2 . Tedaj za formulo F velja

$$\sigma_F(E_1 - E_2) = \sigma_F(E_1) - \sigma_F(E_2).$$

Če se atributi relacij E_1 in E_2 razlikujejo od atributov relacije E , je v formulah, ki nastopata na desni strani enakosti, attribute treba ustrezno zamenjati. Omenimo še, da izbira $\sigma_F(E_2)$, ki nastopa na desni strani enakosti, ni potrebna – zamenjamo jo lahko z E_2 . Zaradi razlike v močeh pa je relacijo $\sigma_F(E_2)$ pogosto lažje določiti kot relacijo E_2 .

6. Naj bo A_1, \dots, A_n nabor atributov relacije E_1 , ki ne vsebuje nobenega atributa relacije E_2 . Tedaj velja

$$\pi_{A_1, \dots, A_n}(E_1 \times E_2) = \pi_{A_1, \dots, A_n}(E_1).$$

- Naj bo B_1, \dots, B_m nabor atributov relacije E_2 , ki ne vsebuje nobenega atributa relacije E_1 . Tedaj velja

$$\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(E_1 \times E_2) = \pi_{A_1, \dots, A_n}(E_1) \times \pi_{B_1, \dots, B_m}(E_2).$$

- Če pa nabor B_1, \dots, B_m vsebuje attribute relacije E_1 in attribute relacije E_2 , velja

$$\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(E_1 \times E_2) = \pi_{A_1, \dots, A_n, B_1, \dots, B_m}(\pi_{A_1, \dots, A_n}(E_1) \times E_2).$$

7. Naj se atributi relacije $E = E_1 \cup E_2$ ujemajo z atributi relacij E_1 in E_2 . Tedaj za attribute A_1, \dots, A_n velja

$$\pi_{A_1, \dots, A_n}(E_1 \cup E_2) = \pi_{A_1, \dots, A_n}(E_1) \cup \pi_{A_1, \dots, A_n}(E_2).$$

- Če se atributi relacij E_1 in E_2 razlikujejo od atributov relacije E , je na desni strani enakosti attribute treba ustrezno zamenjati.

8. Naj formula F vsebuje le attribute, ki so skupni relacijama E_1 in E_2 . Tedaj velja

$$\sigma_F(E_1 \bowtie E_2) = \sigma_F(E_1) \bowtie \sigma_F(E_2).$$

9. Naj bodo A_1, \dots, A_n atributi, med katerimi so B_1, \dots, B_m atributi relacije E_1 , ostali C_1, \dots, C_{n-m} pa atributi relacije E_2 . Tedaj velja

$$\pi_{A_1, \dots, A_n}(E_1 \times E_2) = \pi_{B_1, \dots, B_m}(E_1) \times \pi_{C_1, \dots, C_{n-m}}(E_2).$$

Našteti algebrajski zakoni igrajo pomembno vlogo pri algebrajski optimizaciji oz. algebrajski poenostavitvi. Za izraz relacijske algebre, ki ga z uporabo naštetih algebrajskih zakonov in obstoječih principov optimizacije [Ullman-1989] dobimo iz začetne poizvedbe, pa žal ni nobenega zagotovila, da je optimalen izmed vseh enakovrednih izrazov.

6 Deduktivni podatkovni model

6.1 Uvod

6.2 Sintaksa Dataloga

6.3 Semantika Dataloga

6.4 Predpostavka zaprtega sveta

6.5 Prednosti in slabosti Dataloga

Predstavimo deduktivni podatkovni model in najbolj razširjen jezik deduktivnih podatkovnih modelov, imenovan Datalog, katerega formule in pravila sklepanja so zapisani v obliki Hornovih stavkov, ki tvorijo podmnožico izjav logike prvega reda.

Opišemo tri različne načine interpretacije logičnih pravil. To so: interpretacija z dokazi, interpretacija z modeli in interpretacija z algoritmi.

Obravnavamo tri podmnožice Datalog pravil – nerekurzivna Datalog pravila, rekurzivna Datalog pravila in Datalog pravila z negacijo, katerih kompleksnost narašča v podanem vrstnem redu.

Navedemo prednosti in slabosti logike kot teoretične osnove podatkovnih sistemov.

6.1 Uvod

Uporaba logike in deduktivnega sklepanja v okviru podatkovnih zbirk sega v pozna šestdeseta leta prejšnjega stoletja [Levien-1965, Kuhns-1967, Gallaire-1978]. Največji vpliv pa je imel prav razvoj logičnega programiranja [Kowalski-1974, Das-1992], kjer se logika uporablja kot programski jezik. *Logični program* imenujemo končno množico *formul* in *pravil sklepanja*. Formule so izjave o danem svetu kot na primer

“Tomaž je Majin sin”,

pravila sklepanja pa so izjave, z uporabo katerih lahko iz danih formul izpeljemo nove formule kot na primer

“Če je X oče od Y in je Y oče od Z , tedaj je X dedek od Z ”.

Tako formule kot pravila sklepanja predstavljajo znanje o svetu. Logični program brez funkcijskih simbolov imenujemo *deduktivna podatkovna zbirka*. Zgodovinski razvoj deduktivnih podatkovnih zbirk je pregledno podan v [Minker-1996, Ramakrishnan-1995], njihova uporaba pa v [Colomb-1998].

Za relacijske podatkovne zbirke pogosto pravimo, da so predhodnik deduktivnih podatkovnih zbirk. Čeprav logična formalizacija relacijskih podatkovnih zbirk, ki jo je uvedel R. Reiter [Reiter-1984], vsebuje nekatere omejitve, pa je ta imela pomembno vlogo pri razumevanju modelov podatkovnih zbirk ter pri razumevanju pojmov *poizvedbe* in *odgovora*. Reiter je navedel naslednje predpostavke o relacijski podatkovni zbirki, opisani z jezikom prvega reda brez funkcijskih simbolov:

1. PREDPOSTAVKA ZAPRTEGA SVETA: Formule, ki v relacijski podatkovni zbirki niso resnične, so lažne. Ta predpostavka je potrebna zaradi izjavnega veznika \neg .
2. PREDPOSTAVKA ENOLIČNIH IMEN: Vsaka konstanta, zapisana v zbirki, ima edinstveno ime. Dve konstanti z različnima imenoma sta različni.
3. PREDPOSTAVKA ZAPRTIH DOMEN: V podatkovni zbirki so zapisane vse konstante danega jezika.

Na osnovi naštetih predpostavk je nastal logični model relacijskih podatkovnih zbirk, v katerem je *poizvedba* izjava v danem jeziku, *odgovor* pa logična posledica podatkovne zbirke.

Najbolj znan model deduktivnih podatkovnih zbirk je *Datalog* [Ceri-1989]. Čeprav Datalog temelji na relacijskem podatkovnem modelu (predikatni simboli označujejo relacije), se od njega razlikuje. V Datalogu namreč nastopa dvojje relacij:

1. relacije *ekstenzionalne podatkovne zbirke* (EPZ) ustrezajo predikatom, ki so shranjeni v podatkovni zbirki,
2. relacije *intenzionalne podatkovne zbirke* (IPZ) ustrezajo predikatom, ki so definirani z logičnimi pravili.

V relacijskem podatkovnem modelu so vse relacije vrste EPZ, v Datalogu pa je poljuben predikatni simbol ali vrste EPZ ali IPZ. Relacije IPZ so sicer nekoliko podobne t. i. *pogledom* v relacijskem podatkovnem modelu, vendar pa IPZ relacije in pogledi po moči niso primerljivi, kar bomo videli v poglavju 7.

6.2 Sintaksa Dataloga

V Datalogu so formule in pravila sklepanja zapisani v obliki *Hornovih stavkov*, ki tvorijo podmnožico izjav *prvega reda*, njihova definicija pa temelji na pojmih *atomarne formule* in *literala*.

Jezik prvega reda sestavljajo naslednji simboli ($n \geq 1$):

- konstante (1, sandra, drevo ...)
- spremenljivke (X, Y, A, B ...)
- n -mestni predikatni simboli (brat, $>$...)
- n -mestni funkcijski simboli (koren, glavno_mesto, f ...)
- logični vezniki $\wedge \vee \neg \Rightarrow \Leftrightarrow$
- kvantifikatorja $\forall \exists$
- ločila $() , :$

Za konstante, predikatne simbole in funkcijske simbole uporabljamo nize, sestavljene iz malih črk in števil, za spremenljivke pa nize z veliko začetnico. Množico *termov* logike prvega reda definiramo takole:

1. Konstante in spremenljivke so termi.
2. Če je f n -mestni funkcijski simbol in so t_1, t_2, \dots, t_n termi, potem je $f(t_1, t_2, \dots, t_n)$ term.

Naj bo p n -mestni predikat, t_1, \dots, t_n pa termi brez funkcijskih simbolov, tj. spremenljivke in konstante. Tedaj izjavo $p(t_1, \dots, t_n)$ imenujemo *atomarna formula*. Atomarni formuli $p(t_1, \dots, t_n)$ pripada relacija – podmnožica relacije predikata p , za katero velja:

- \rightsquigarrow Konstante, ki nastopajo v atomarni formuli, se ujemajo z istoležnimi komponentami elementov relacije.
- \rightsquigarrow Vse komponente elementov relacije, ki ustrezajo istim spremenljivkam pripadajoče atomarne formule, so enake.

Atomarne formule lahko temeljijo tudi na predikatih aritmetičnih primerjav: $=$, $<$, $>$, \leq , \geq ali \neq . Te predikate bomo imenovali *vgrajeni predikati*, atomarno formulo $\theta(X, Y)$ z vgrajenim predikatом θ pa bomo pisali kot $X\theta Y$. Predikate, ki niso vgrajeni, bomo imenovali *običajni predikati*.

Literal je atomarna formula, tj. *pozitivni literal*, ali negacija atomarne formule, tj. *negativni literal*. Negacijo atomarne formule $p(t_1, \dots, t_n)$ označimo z $\neg p(t_1, \dots, t_n)$. *Izjava* je logična disjunkcija literalov.

Hornov stavek je izjava, ki vsebuje največ en pozitiven literal. To je torej:

1. pozitiven literal, ki ga imenujemo *dejstvo*, ali
2. logična disjunkcija enega ali več negativnih literalov, ki jo imenujemo *integritetna omejitev*, ali
3. logična disjunkcija enega pozitivnega in enega ali več negativnih literalov, ki jo imenujemo *pravilo*.

Zbirko Hornovih stavkov imenujemo *Datalog program*. Datalog ne dovoljuje funkcijskih simbolov. Ko bomo v okviru Dataloga govorili o logiki prvega reda, tega ne bomo vedno poudarjali.

Dejstva določajo EPZ relacije, pravila pa IPZ relacije. Integritetne omejitve bomo obravnavali šele v poglavju 8. Hornove stavke iz 3. točke definicije imenujemo pravila, ker je Hornov stavek oblike

$$\neg p_1 \vee \dots \vee \neg p_n \vee q$$

logično ekvivalenten izjavi $p_1 \wedge \dots \wedge p_n \Rightarrow q$.

Hornove stavke bomo zapisovali s sintakso programskega jezika Prolog [Colomb-1998]:

$$\begin{array}{ll} \text{Dejstvo :} & p \rightsquigarrow p. \\ \text{Integritetna omejitev :} & \neg p_1 \vee \dots \vee \neg p_n \rightsquigarrow :- p_1 \& \dots \& p_n. \\ \text{Pravilo :} & \neg p_1 \vee \dots \vee \neg p_n \vee q \rightsquigarrow q :- p_1 \& \dots \& p_n. \end{array}$$

V pravilu

$$q :- p_1 \& \dots \& p_n.$$

imenujemo atomarno formulo q *glava pravila*, $p_1 \& \dots \& p_n$ pa *telo pravila*. Posamezne atomarne formule p_i imenujemo *delni cilji*. V integritetni omejitvi

$$:- p_1 \& \dots \& p_n.$$

imenujemo $p_1 \& \dots \& p_n$ *telo integritetne omejitve*. V glavi pravil nastopajo le IPZ predikati, v telesih pravil pa lahko nastopajo EPZ in IPZ predikati. Spremenljivke, ki nastopajo le v glavi pravila, so pod vplivom delovanja eksistencialnega kvantifikatorja \exists nad glavo pravila, ostale spremenljivke pa so pod vplivom delovanja univerzalnega kvantifikatorja \forall nad celotnim pravilom. Vse spremenljivke integritetnih omejitev so pod vplivom univerzalnega kvantifikatorja \exists .

Primer 6.1. *Vzemimo pravilo*

$$\text{sestra_brat}(X, Y) : - \text{roditelj}(X, Z) \ \& \ \text{roditelj}(Y, Z) \ \& \ X \neq Y.$$

Preberemo ga takole: Za vse X in Y je X sestra oz. brat od Y , če obstaja Z , ki je roditelj (oče ali mati) X in Y , ter X in Y nista isti osebi. Logično ekvivalentno je pravilo interpretirati takole: Za vse X , Y in Z je X sestra oz. brat od Y , če je Z roditelj X in Y , X in Y pa sta različna.

Poizvedba podatkovnega modela Datalog je izjava, zapisana v jeziku prvega reda, s katero povprašujemo po podatkih deduktivne podatkovne zbirke. Primer poizvedbe je

$$\exists X : \text{sestra_brat}(\text{nina}, X),$$

ki jo bomo pisali kot

$$\text{sestra_brat}(\text{nina}, X)?$$

in sprašuje po sestrah in bratih osebe z imenom Nina. *Odgovor* na poizvedbo je odziv, zapisan v jeziku prvega reda, ki podaja stanje podatkovne zbirke, po katerem sprašujemo. Primer odgovora na zgornjo poizvedbo je:

$$\text{sestra_brat}(\text{nina}, \text{matej})$$

$$\text{sestra_brat}(\text{nina}, \text{maja})$$

Ta odgovor, zapisan s sintakso Prologa, ki pravi, da ima oseba z imenom Nina sestro z imenom Maja in brata z imenom Matej, je videti takole:

$$\text{sestra_brat}(\text{nina}, \text{matej}).$$

$$\text{sestra_brat}(\text{nina}, \text{maja}).$$

Primer 6.2. *Imejmo naslednji relacijski shemi:*

$$\begin{aligned} & \text{STRANKE}(\text{IME_STRANKA}, \text{NASLOV_STRANKA}, \text{BILANCA}) \\ & \text{VSEBINA}(\# \text{NAROČILO}, \text{IME_IZDELEK}, \text{KOLIČINA}) \end{aligned}$$

Atomarna formula

$$\text{stranke}(\text{tomaž}, \text{Naslov}, \text{Bilanca}),$$

kjer je tomaž konstanta, Naslov in Bilanca pa sta spremenljivki, predstavlja relacijo $\sigma_{\$1=\text{tomaž}}(\text{STRANKE})$. Atomarna formula

$$\text{vsebina}(X, \text{Izdelek}, X),$$

kjer sta Izdelek in X spremenljivki, predstavlja relacijo $\sigma_{\$1=\$3}(\text{VSEBINA})$, torej izdelke, katerih zaporedna številka naročila se ujema z naročeno količino.

Izjavni veznik \neg , ki ga uporabljamo v okviru Dataloga, se ponavadi ne ujema z logično negacijo, tj. z izjavnim veznikom \neg logike prvega reda. Njegov pomen je odvisen od deduktivne podatkovne zbirke oz. njenih predpostavk. Mi bomo uporabljali pomen, ki temelji na *predpostavki zaprtega sveta* (razdelek 6.4).

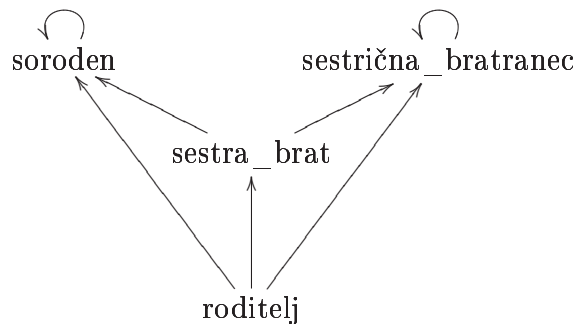
Med predikati Datalog programov pogosto obstaja vrsta odvisnosti, ki jih ponazorimo s strukturo, imenovano *graf odvisnosti*. Množico vozlišč grafa odvisnosti Datalog programa tvorijo običajni predikati. Med predikatoma p in q obstaja povezava, če obstaja pravilo z delnim ciljem, ki vsebuje predikat p , in z glavo, ki vsebuje predikat q . Vgrajenih predikatov ne vključimo v množico vozlišč grafa odvisnosti.

Če v grafu odvisnosti Datalog programa obstaja vsaj en cikel (lahko tudi zanka), program imenujemo *rekurziven*, sicer ga imenujemo *nerekurziven*. Predikate, ki ležijo na katerem od ciklov, imenujemo *rekurzivni predikati*. Vse predikate nerekurzivnih Datalog programov kakor tudi predikate rekurzivnih Datalog programov, ki niso del nobenega cikla, imenujemo *nerekurzivni predikati*.

Primer 6.3. *Vzemimo naslednji Datalog program:*

- (1) $sestra_brat(X, Y)$: – $roditelj(X, Z) \ \& \ roditelj(Y, Z) \ \& \ X \neq Y$.
- (2) $sestrična_bratranec(X, Y)$: – $roditelj(X, X_p) \ \& \ roditelj(Y, Y_p) \ \& \ sestra_brat(X_p, Y_p)$.
- (3) $sestrična_bratranec(X, Y)$: – $roditelj(X, X_p) \ \& \ roditelj(Y, Y_p) \ \& \ sestra_brat(X_p, Y_p)$.
- (4) $soroden(X, Y)$: – $sestra_brat(X, Y)$.
- (5) $soroden(X, Y)$: – $soroden(X, Z) \ \& \ roditelj(Y, Z)$.
- (6) $soroden(X, Y)$: – $soroden(Z, Y) \ \& \ roditelj(X, Z)$.

Njemu pripadajoč graf odvisnosti je podan s sliko 6.1.



Slika 6.1: Graf odvisnosti za Datalog program iz primera 6.3.

Zaradi pravila (1) v grafu odvisnosti obstaja povezava med predikatoma $roditelj$ in $sestra_brat$. Pravilo (2) porodi povezavo med predikatoma $roditelj$ in $sestrična_bratranec$

ter povezavo med predikatoma *sestra_brat* in *sestrična_bratranec*. Povezavo med predikatoma *roditelj* in *sestrična_bratranec* ter zanko na vozlišču *sestrična_bratranec* utemeljuje pravilo (3). Druge povezave sledijo iz pravil (4)-(6).

Graf odvisnosti, podan s sliko 6.1, vsebuje dva cikla, tj. zanki na vozliščih *soroden* in *sestrična_bratranec*. Predikata *soroden* in *sestrična_bratranec* ter pripadajoč Datalog program so zato rekurzivni, predikata *roditelj* in *sestra_brat* pa sta nerekurzivna.

EPZ predikati, za katere vemo, da nastopajo le v telesih pravil, so natanko tisti predikati, katerih vozlišča v grafu odvisnosti nimajo vhodnih povezav, in torej ne morejo biti rekurzivni. Vsi EPZ predikati so zato nerekurzivni.

Če naj Datalog pravila predstavljajo operacije nad končnimi relacijami, iz danih končnih relacij ne smejo ustvariti neskončnih relacij. Zato bomo zahtevali, da vsaka spremenljivka, ki v telesu pravila nastopa v atomarni formuli z vgrajenim predikatom, nastopa v telesu istega pravila še v kakšni atomarni formuli z običajnim predikatom, ki ji predpisuje končno množico možnih vrednosti. Prepovedati pa moramo tudi spremenljivke, ki nastopajo le v glavi pravila. Spremenljivke, ki omenjene zahteve izpolnjujejo, imenujemo *omejene spremenljivke*. Omejene spremenljivke danega pravila lahko opišemo takole:

1. Spremenljivka, ki nastopa v delnem cilju z običajnim predikatom, je omejena.
2. Spremenljivka X , ki nastopa v delnem cilju oblike $X = a$ ali $a = X$, kjer je a konstanta, je omejena.
3. Spremenljivka X , ki nastopa v delnem cilju oblike $X = Y$ ali $Y = X$, kjer je Y spremenljivka, za katero vemo, da je omejena, je omejena.

Pravilo je *varno*, če so vse njegove spremenljivke omejene. Z oznako $\text{Datalog}_{\text{varno}}$ bomo označevali Datalog programe, ki vsebujejo dejstva in varna pravila brez negacije.

Primer 6.4. Pravilo

$$\text{večji}(X, Y) : - X > Y.$$

ni varno, saj nobena njegova spremenljivka ni omejena. Pravilo (1) iz primera 6.3 je varno, saj so spremenljivke X , Y in Z , zaradi nastopa v delnih ciljih s predikatom *roditelj*, omejene. Tudi pravila (2)-(6) so varna. Vzemimo še pravilo

$$p(X, Y) : - q(X, Z) \ \& \ W = a \ \& \ Y = W.$$

Spremenljivki X in Z sta omejeni zaradi 1. točke definicije omejenih spremenljivk, W je omejena zaradi 2. točke, Y pa posledično zaradi 3. točke iste definicije. Ker so vse spremenljivke omejene, je pravilo varno.

6.3 Semantika Dataloga

Logična pravila lahko interpretiramo na tri različne načine:

1. INTERPRETACIJA Z DOKAZI: Znana dejstva vstavimo v desne strani logičnih pravil in izpeljemo nova dejstva, tj. leve strani pravil. Množico vseh dejstev, ki jih lahko tako dokažemo, imenujemo *dokazljiva dejstva*. Izkaže se (razdelek 6.3.3), da se v primeru pravil z negacijo množica dokazljivih dejstev in intuitivni pomen pravil pogosto ne ujemata.
2. INTERPRETACIJA Z MODELI: Posameznim nastopom določenega predikata pripišemo vrednost *resnično* ali *napačno* (oz. *neresnično*). Množico resničnih nastopov imenujemo *interpretacija*. Taka interpretacija je *model* množice pravil, če uresničuje vsa pravila (primer 6.5). Če po odstranitvi poljubnega resničnega dejstva model postane nekonsistenten s podatkovno zbirko, ga imenujemo *minimalni model*. Izkaže se (razdelek 6.3.3), da v primeru pravil z negacijo pogosto obstaja celo več minimalnih modelov in da včasih nobeden izmed njih ne ustreza množici dokazljivih dejstev. Določen nastop danega n -mestnega predikata je resničen natanko tedaj, ko njemu pripadajoča relacija podatkovne zbirke vsebuje to n -terico kot element.
3. INTERPRETACIJA Z ALGORITMI: Z različnimi algoritmi ugotavljamo resničnost morebitnih dejstev. Ta način določanja pomena logičnih pravil uporablja programski jezik Prolog. Na žalost se izkaže [Ullman-1988], da se množica dejstev, ki jih najde Prolog, in množica vseh dokazljivih dejstev ne ujemata vedno. Množica dejstev, ki jih najde Prolog, niti ni nujno model. Ko bomo uporabljali ta način interpretacije, bomo pravila pretvorili v izraze relacijske algebre. Za Datalog pravila brez negacije bomo dokazali (razdelek 6.3.2), da taka interpretacija porodi enoličen minimalni model in (zato) množico dokazljivih dejstev. V primeru negacije se bomo omejili na posebno podmnožico Datalog programov in zapisali algoritem za izračun (ne nujno enoličnega) minimalnega modela.

Primer 6.5. *Oglejmo si primer interpretacije z modeli. Vzemimo naslednji pravili:*

$$\begin{aligned} (1) \quad p(X) &: - \quad q(X). \\ (2) \quad q(X) &: - \quad r(X). \end{aligned}$$

Za domene predikatov p , q in r vzemimo množico naravnih števil. Pravili pravita, da je p resničen za vsa naravna števila, za katera je resničen q , in da je q resničen za vsa naravna števila, za katera je resničen r .

Naj bo $M_1 = \{r(1), q(1), p(1), q(2), p(2), p(3)\}$ interpretacija danih pravil. Zanima nas, ali je M_1 tudi model. Ko v pravilo (1) vstavimo $X = 1$, postaneta obe strani pravila resnični. Enako se zgodi, ko vstavimo $X = 1$ v pravilo (2) ali $X = 2$ v pravilo (1). Te vrednosti torej dani pravili uresničujeta. Ko pa $X = 2$ vstavimo v pravilo (2) ali $X = 3$ v pravilo (1), postane leva stran resnična, desna pa ne. Tudi to je način uresničitve

danih pravil. Če vstavimo $X = 3$ v pravilo (2) ali katero koli naravno število različno od 1, 2 in 3 v katero koli pravilo, postaneta obe strani izbranega pravila neresnični, pravilo pa resnično. Ugotovili smo, da interpretacija M_1 uresničuje vsa pravila, kar pomeni, da je model.

V nadaljevanju bomo obravnavali tri podmnožice Datalog pravil, namreč nerekurzivna Datalog pravila (razdelek 6.3.1), rekurzivna Datalog pravila (razdelek 6.3.2) in Datalog pravila z negacijo (razdelek 6.3.3). Njihova kompleksnost narašča v tem vrstnem redu.

6.3.1 Interpretacija nerekurzivnih Datalog pravil

Za interpretacijo nerekurzivnih Datalog pravil bomo uporabljali interpretacijo z algoritmi. Pokazali bomo, da lahko nerekurzivna Datalog_{varno} pravila pretvorimo v izraze relacijske algebre. Ti izrazi izračunajo relacije IPZ predikatov, ki so hkrati minimalni model pravil in množica IPZ dejstev, izpeljivih iz pravil in podatkovne zbirke.

Ker graf odvisnosti nerekurzivnega Datalog_{varno} programa ne vsebuje ciklov, ga topološko uredimo. Vozlišča p_1, \dots, p_n grafa odvisnosti uredimo, tako da za povezave $p_i p_j$ velja $i < j$. Relacije, ki ustrezajo atomarnim formulam s predikati p_1, \dots, p_n , lahko tedaj računamo v tem vrstnem redu. Ko bomo računali relacijo za p_i , bomo relacije, pripadajoče delnim ciljem za p_i , že poznali.

Prvi korak pri pretvorbi danega pravila v izraz relacijske algebre je najti množico vrednosti, ki jih lahko zavzamejo spremenljivke, da bo glava pravila resnična. V ta namen definirajmo t. i. *relacijo telesa pravila*. Ta relacija s spremenljivkami X_1, \dots, X_n kot atributi vsebuje element (a_1, \dots, a_n) natanko tedaj, ko so vsi delni cilji danega pravila po zamenjavi spremenljivk s pripadajočimi a_i ($1 \leq i \leq n$) resnični. Rekli bomo, da zamenjava delne cilje *uresniči*.

Algoritem 1 opisuje konstrukcijo izraza relacijske algebre, ki ustreza relaciji telesa vhodnega nerekurzivnega Datalog_{varno} pravila.

Algoritem 1. IZRAČUN IZRAZA RELACIJSKE ALGEBRE, KI USTREZA RELACIJI TELESA NEREKURZIVNEGA DATALOG_{varno} PRAVILA.

VHOD: Telo nerekurzivnega Datalog_{varno} pravila r , ki sestoji iz delnih ciljev S_1, \dots, S_n , ki vsebujejo spremenljivke X_1, \dots, X_m . Delnemu cilju $S_i = p_i(A_{i1}, \dots, A_{ik_i})$ z običajnim predikatom pripada relacija R_i , katere argumenti so termini A_{i1}, \dots, A_{ik_i} .

IZHOD: Izraz relacijske algebre, imenovan

$$\text{EVAL-RULE}(r, R_1, \dots, R_n),$$

ki iz relacij R_1, \dots, R_n izračuna relacijo $R(X_1, \dots, X_m)$, katere elementi so natanko tiste m -terice (a_1, \dots, a_m) , da zamenjava X_j z a_j , $1 \leq j \leq m$, delne cilje S_1, \dots, S_n uresniči.

Delni cilji, ki vsebujejo vgrajene predikate, nimajo pripadajočih relacij. Zaradi tega je število argumentov R_j v EVAL-RULE ponavadi manjše od n .

POSTOPEK: Izračun je razdeljen na več korakov:

1. Za vsak delni cilj S_i , ki vsebuje običajni predikat, definiramo izraz

$$Q_i = \pi_{V_i}(\sigma_{F_i}(R_i)),$$

kjer je V_i množica, ki za vsako spremenljivko X iz S_i vsebuje po eno zaporedno številko komponente enake X (ista spremenljivka lahko v danem delnem cilju nastopa večkrat). Formula F_i je logična konjunkcija:

- Če je na k -tem mestu delnega cilja S_i konstanta a , formula F_i vsebuje izraz $\$k = a$.
- Če je na k -tem in l -tem mestu delnega cilja S_i ista spremenljivka, formula F_i vsebuje izraz $\$k = \l . Pri tem ni nujno, da formula F_i vsebuje izraze te oblike za vse take pare k in l . Dodati je treba le toliko parov, da zagotovimo enakost vseh nastopov iste spremenljivke. Če npr. spremenljivka X nastopa na mestih 2, 5 in 6, zadostuje dodati izraza $\$2 = \5 in $\$5 = \6 .

Če je S_i tak, da ne izpolnjuje nobenega izmed zgornjih pogojev, na primer $S_i = p(X, Y)$, tedaj je F_i tautologija in $Q_i = R_i$.

2. Za vsako spremenljivko X , ki ne nastopa v nobenem delnem cilju z običajnim predikatom, izračunamo izraz D_X kot enomestno relacijo, katere elementi so vrednosti, ki jih X zavzame v določilu, ki uresniči vse delne cilje pravila r . Ker je pravilo r varno, je spremenljivka X omejena ali z delnim ciljem $X = a$ oz. $a = X$ ali z verižno izenačitvijo z neko omejeno spremenljivko Y . Ravnamo takole:

- Če je $X = a$ ali $a = X$ delni cilj, definiramo $D_X = \{a\}$.
- Če se Y pojavi kot j -ti argument običajnega delnega cilja S_i , definiramo $D_X = \pi_j(R_i)$.

3. Naj bo E naravni spoj vseh Q_i in D_X , definiranih v 1. in 2. koraku tega algoritma. Pri tem za attribute relacije Q_i vzamemo spremenljivke iz S_i , za edini atribut relacije D_X pa X . Ker relaciji D_X pripadajoča spremenljivka X ni atribut nobenega Q_i , naravni spoj vsebuje kartezični produkt vseh D_X .

4. Definiramo

$$\text{EVAL-RULE}(r, R_1, \dots, R_n) = \sigma_F(E),$$

kjer je F konjunkcija vseh vgrajenih delnih ciljev $X\theta Y$, E pa izraz, definiran v 3. koraku algoritma. Če pravilo r ne vsebuje vgrajenih delnih ciljev, definiramo $\text{EVAL-RULE}(r, R_1, \dots, R_n) = E$.

Primer 6.6. *Vzemimo nerekurzivno Datalog_{varno} pravilo (1) iz primera 6.3:*

$$\text{sestra_brat}(X, Y) \quad :- \quad \text{roditelj}(X, Z) \ \& \ \text{roditelj}(Y, Z) \ \& \ X \neq Y.$$

Pravilo poimenujmo r . S pomočjo algoritma 1 poiščimo izraz relacijske algebre, ki podaja relacijo telesa pravila r . Naj bosta $RODITELJ(X, Z)$ in $RODITELJ(Y, Z)$ relacijski shemi dveh kopij relacije $RODITELJ$, ki pripadata edinima delnima ciljema z običajnim predikatom, tj. delnima ciljema $\text{roditelj}(X, Z)$ in $\text{roditelj}(Y, Z)$.

1. *Formuli F_1 in F_2 , ki pripadata delnima ciljema $\text{roditelj}(X, Z)$ in $\text{roditelj}(Y, Z)$, sta po algoritmu 1 tautologiji, zato definirajmo:*

$$Q_1(X, Z) = RODITELJ(X, Z)$$

$$Q_2(Y, Z) = RODITELJ(Y, Z)$$

2. *Vse spremenljivke pravila r , tj. X , Y in Z , nastopajo v vsaj enem delnem cilju z običajnim predikatom, zato 2. korak algoritma izpustimo.*
3. *Izraz E definirajmo kot naravni spoj relacij Q_1 in Q_2 :*

$$E = RODITELJ(X, Z) \bowtie RODITELJ(Y, Z)$$

V poglavju 5 smo povedali, da je naravni spoj komutativna operacija na relacijah, zato vrstni red spajanja relacij ni pomemben.

4. *Ker pravilo r vsebuje le en delni cilj z vgrajenim predikatom, namreč $X \neq Y$, je*

$$\text{EVAL-RULE}(r, RODITELJ) = \sigma_{X \neq Y}(RODITELJ(X, Z) \bowtie RODITELJ(Y, Z)).$$

Torej, relacija telesa pravila r je podmnožica tistih trojk (x, y, z) iz $RODITELJ(X, Z) \bowtie RODITELJ(Y, Z)$, katerih prvi dve komponenti sta različni ter je $(x, z) \in RODITELJ$ in $(y, z) \in RODITELJ$. To so natanko tiste trojke (x, y, z) , ki vse delne cilje pravila r uresničijo.

Primer 6.7. *Vzemimo sedaj nerekurzivno Datalog_{varno} pravilo:*

$$p(X, Y) \quad :- \quad s(a, X) \ \& \ t(X, Z, X) \ \& \ u(Y, Z).$$

Pravilo poimenujmo r . S pomočjo algoritma 1 poiščimo izraz relacijske algebre, ki podaja relacijo telesa pravila r . Naj bodo S , T in U relacije, ki pripadajo delnim ciljem pravila.

1. *Ker je prva komponenta delnega cilja $s(a, X)$ konstanta, je F_s enak $\$1 = a$ in*

$$Q_s(X) = \pi_2(\sigma_{\$1=a}(S)).$$

Ker je na prvem in tretjem mestu delnega cilja $t(X, Z, X)$ spremenljivka X , je F_t enak $\$1 = \3 in

$$Q_t(X, Z) = \pi_{1,2}(\sigma_{\$1=\$3}(T)).$$

Formula F_u , ki pripada delnemu cilju $u(Y, Z)$, je po algoritmu 1 tautologija in

$$Q_u(Y, Z) = U(Y, Z).$$

2. Vse spremenljivke pravila r , tj. X , Y in Z , nastopajo v vsaj enem delnem cilju z običajnim predikatom, zato 2. korak algoritma izpustimo.
3. Izraz E definirajmo kot naravni spoj relacij Q_s , Q_t in Q_u :

$$E = \pi_2(\sigma_{\$1=a}(S)) \bowtie \pi_{1,2}(\sigma_{\$1=\$3}(T)) \bowtie U(Y, Z)$$

V 5. poglavju smo povedali, da je naravni spoj asociativna in komutativna operacija na relacijah, zato vrstni red spajanja relacij ni pomemben.

4. Ker pravilo r ne vsebuje delnih ciljev z vgrajenimi predikati, definiramo:

$$\text{EVAL-RULE}(r, S, T, U) = \pi_2(\sigma_{\$1=a}(S)) \bowtie \pi_{1,2}(\sigma_{\$1=\$3}(T)) \bowtie U(Y, Z)$$

Relacija P je množica trojk (x, y, z) , za katere velja: $(a, x) \in S$, $(x, z, x) \in T$ in $(y, z) \in U$. To so natanko tiste trojke (x, y, z) , ki vse delne cilje pravila r uresničijo.

Algoritem 1 je pravilen – izhodna relacija $R(X_1, \dots, X_m)$ vsebuje natanko tiste elemente (a_1, \dots, a_m) , ki po zamenjavah X_j z a_j , $1 \leq j \leq m$, vse delne cilje pravila uresničijo [Ullman-1988].

Naslednji korak pri pretvorbi danega pravila v izraz relacijske algebre je relacijo telesa pravila pretvoriti v relacijo IPZ predikata p , ki nastopa v glavi. Osnovna ideja je, da vzamemo vsa pravila s predikatom p v glavi, izračunamo relacije teles teh pravil, jih projiciramo na spremenljivke, ki nastopajo v glavah, in jih z operacijo unije združimo. Težava se pojavi, ko v glavah s predikatom p nastopajo konstante ali podvojene spremenljivke, kot je na primer $p(a, X, X)$.

Množico pravil za predikat p , katerih glave so enake $p(X_1, \dots, X_k)$, kjer so X_1, \dots, X_k paroma različne spremenljivke, bomo imenovali *množica urejenih pravil*. Dane množice neurejenih pravil za predikat p pa ni težko urediti. Naj ima pravilo r glavo oblike $p(t_1, \dots, t_k)$, kjer t_1, \dots, t_k niso nujno paroma različni termini. Glavo pravila r zamenjamo s $p(X_1, \dots, X_k)$, kjer so X_1, \dots, X_k paroma različne, nove spremenljivke, v telo pravila r pa za $1 \leq i \leq k$ dodamo delne cilje $X_i = t_i$. Če je t_i spremenljivka nekega pravila, lahko namesto da bi pravilu dodali delni cilj $X_i = t_i$, vse t_i zamenjamo z X_i . Tako zamenjavo lahko v danem pravilu za vsak t_i naredimo samo enkrat, saj lahko ob večkratni zamenjavi spet dobimo podvojene spremenljivke.

Primer 6.8. *Vzemimo predikat p , definiran z naslednjima praviloma:*

$$\begin{aligned} p(a, X, Y) &: - q(X, Y). \\ p(X, Y, X) &: - q(Y, X). \end{aligned}$$

Pravili uredimo tako, da njuni glavi zamenjamo s $p(U, V, W)$ in telesoma dodamo ustrezne delne cilje:

$$\begin{aligned} p(U, V, W) &: - q(X, Y) \quad \& \quad U = a \quad \& \quad V = X \quad \& \quad W = Y. \\ p(U, V, W) &: - q(Y, X) \quad \& \quad U = X \quad \& \quad V = Y \quad \& \quad W = X. \end{aligned}$$

Namesto delnih ciljev $V = X$ in $W = Y$, ki smo ju dodali prvemu pravilu, lahko vse X zamenjamo z V in vse Y z W . Podobno lahko v drugem pravilu vse X zamenjamo z U in vse Y z V . Dobimo urejeni pravili:

$$\begin{aligned} p(U, V, W) &: - q(V, W) \quad \& \quad U = a. \\ p(U, V, W) &: - q(V, U) \quad \& \quad W = U. \end{aligned}$$

Naj bo r' pravilo, ki ga iz r dobimo po urejanju. Tedaj velja [Ullman-1988]:

- a.) Če je pravilo r varno, je varno tudi pravilo r' .
- b.) Pravili r in r' sta ekvivalentni, kar pomeni, da ob danih relacijah vseh delnih ciljev obeh pravil velja: zamenjava spremenljivk pravila r , ki vse delne cilje uresniči, glavo pa spremeni v $p(a_1, \dots, a_n)$, obstaja natanko tedaj, ko obstaja zamenjava spremenljivk pravila r' , ki vse delne cilje uresniči, glavo pravila r' pa spremeni v $p(a_1, \dots, a_n)$.

Relacije teles urejenih nerekurzivnih $\text{Datalog}_{\text{varno}}$ pravil pretvorimo v relacijo predikata, tako da jih projiciramo na spremenljivke, ki nastopajo v glavah, in jih z operacijo unije združimo. Postopek opisuje algoritem 2.

Algoritem 2. IZRAČUN IZRAZOV RELACIJSKE ALGEBRE, KI USTREZAJO POSAMEZ-
NIM IPZ PREDIKATOM NEREKURZIVNEGA $\text{DATALOG}_{\text{varno}}$ PROGRAMA.

VHOD: Nerekurziven $\text{Datalog}_{\text{varno}}$ program in relacije R_1, \dots, R_m , pripadajoče posameznim delnim ciljem z EPZ predikati.

IZHOD: Za vsak IPZ predikat p po en izraz relacijske algebre, ki iz relacij R_1, \dots, R_m izračuna relacijo predikata p .

POSTOPEK: Konstruiramo graf odvisnosti vhodnega nerekurzivnega $\text{Datalog}_{\text{varno}}$ programa in predikate p_1, \dots, p_n uredimo, tako da za vsako povezavo $p_i p_j$ velja $i < j$. Ker je program nerekurziven, ne vsebuje ciklov, zato tak vrstni red obstaja.

IPZ predikat p_i izračunamo po naslednjem postopku:

1. Za vsako pravilo r , katerega glava vsebuje p_i , dobimo z uporabo algoritma 1 izraz E_r relacijske algebre, ki iz relacij delnih ciljev pravila r izračuna relacijo R_r telesa pravila r .
2. Ker je vhodni Datalog_{varno} program nerekurziven, razpolagamo z vsemi izrazi relacijske algebre, ki z uporabo EPZ relacij izračunajo relacije delnih ciljev pravila r . V izrazu E_r vse IPZ relacije zamenjamo s pripadajočimi izrazi in dobljen izraz poimenujemo F_r .
3. Predpostavimo lahko, da so glave posameznih pravil za p_i vse enake $p_i(X_1, \dots, X_k)$. To dosežemo po ustreznem preimenovanju spremenljivk. Izraz relacijske algebre, ki izračuna relacijo P_i IPZ predikata p_i , je tedaj unija projekcij $\pi_{X_1, \dots, X_k}(F_r)$, pripadajočih posameznim pravilom r za p_i .

Primer 6.9. *Vzemimo naslednji nerekurziven Datalog_{varno} program:*

$$\begin{aligned}
 (1) \quad p(a, Y) & : - \quad t(X, Y). \\
 (2) \quad p(X, Y) & : - \quad s(X, Z) \quad \& \quad t(Z, Y). \\
 (3) \quad q(X, X) & : - \quad p(X, b). \\
 (4) \quad q(X, Y) & : - \quad p(X, Z) \quad \& \quad s(Z, Y).
 \end{aligned}$$

Naj bosta S in T relaciji, ki pripadata EPZ predikatoma s in t . Z uporabo algoritma 2 poiščimo izraza relacijske algebre, ki izračunata relaciji P in Q , pripadajoči IPZ predikatoma p in q .

Dana pravila Datalog_{varno} programa najprej uredimo. Dobimo množico pravil:

$$\begin{aligned}
 (1) \quad p(X, Y) & : - \quad t(Z, Y) \quad \& \quad X = a. \\
 (2) \quad p(X, Y) & : - \quad s(X, Z) \quad \& \quad t(Z, Y). \\
 (3) \quad q(X, Y) & : - \quad p(X, b) \quad \& \quad Y = X. \\
 (4) \quad q(X, Y) & : - \quad p(X, Z) \quad \& \quad s(Z, Y).
 \end{aligned}$$

Ustrezen vrstni red predikatov je naslednji: s , t , p , q . Relaciji P in Q IPZ predikatov p in q računamo v omenjenem vrstnem redu. Sledi izračun relacije P .

1. Z uporabo algoritma 1 izračunamo E_{p1} in E_{p2} , tj. izraza relacijske algebre, ki izračunata relaciji teles pravil (1) in (2), katerih glava vsebuje IPZ predikat p . Dobimo

$$E_{p1} = T(Z, Y) \bowtie D_X(X),$$

kjer je $D_X = \{a\}$, in

$$E_{p2} = S(X, Z) \bowtie T(Z, Y).$$

2. Izraza E_{p1} in E_{p2} ne vsebujeta nobene IPZ relacije, zato 2. korak algoritma izpustimo.

3. Izraz relacijske algebre, ki izračuna relacijo P , je tedaj unija projekcij $\pi_{X,Y}(E_{p1})$ in $\pi_{X,Y}(E_{p2})$, torej

$$P(X, Y) = \pi_{X,Y}(T(Z, Y) \bowtie \{a\}(X)) \cup \pi_{X,Y}(S(X, Z) \bowtie T(Z, Y)).$$

Sledi še izračun relacije Q .

1. Z uporabo algoritma 1 izračunamo E_{q1} in E_{q2} , tj. izraza relacijske algebre, ki izračunata relaciji teles pravil (3) in (4), katerih glava vsebuje IPZ predikat q . Postopek tokrat opišimo nekoliko podrobneje. Za delni cilj $p(X, b)$ pravila (3) dobimo izraz

$$\pi_X(\sigma_{Z=b}(P(X, Z))),$$

kjer je Z poljubno izbrana spremenljivka, ki pri projekciji izgine. Ta izraz podaja relacijo, katere edini atribut je X , zato za končni izračun relacije $Q(X, Y)$ potrebujemo še izraz, ki bo določal vse možne vrednosti spremenljivke Y . Ker sta X in Y v delnem cilju $Y = X$ izenačena, vemo, da je domena Y lahko le podmnožica domene spremenljivke X . Za domeno spremenljivke Y tedaj lahko vzamemo domeno prvega argumenta relacije P , torej $\pi_Y(P(Y, W))$, kjer je W poljubno izbrana spremenljivka. Izraz relacijske algebre, ki izračuna relacijo telesa pravila (3), je tedaj naravni spoj (v tem primeru kar kartezični produkt) izraza za $p(X, b)$ in domene spremenljivke Y , zaradi vgrajenega delnega cilja dodatno okrnjen z izbiro $Y = X$:

$$E_{q1} = \sigma_{Y=X}(\pi_X(\sigma_{Z=b}(P(X, Z))) \times \pi_Y(P(Y, W)))$$

Za pravilo (4) pa dobimo izraz $E_{q2} = P(X, Z) \bowtie S(Z, Y)$.

2. V izrazih E_{q1} in E_{q2} nadomestimo IPZ relacijo P z že izračunanim izrazom $P(X, Y)$, ki vsebuje le EPZ relacije. Dobimo F_{q1} in F_{q2} .
3. Izraz relacijske algebre, ki izračuna relacijo Q , je tedaj unija projekcij $\pi_{X,Y}(F_{q1})$ in $\pi_{X,Y}(F_{q2})$, torej

$$Q(X, Y) = \sigma_{Y=X}(\pi_X(\sigma_{Z=b}(P(X, Z))) \times \pi_Y(P(Y, W))) \cup \pi_{X,Y}(P(X, Z) \bowtie S(Z, Y)),$$

kjer je IPZ relacijo P treba zamenjati z ekvivalentnim izrazom $P(X, Y)$, ki vsebuje le EPZ relacije.

Algoritem 2 pravilno določi izraze relacijske algebre posameznih IPZ predikatov – izhodne relacije porodijo natanko tista dejstva, ki sledijo iz dejstev in pravil dane podatkovne zbirke [Ullman-1988].

6.3.2 Interpretacija rekurzivnih Datalog pravil

Algoritem 2 za rekurzivne Datalog programe ni primeren. To pa zato, ker predikatov rekurzivnega programa ni mogoče urediti tako, da bi za vsako povezavo $p_i p_j$ v grafu odvisnosti veljalo $i < j$. Če pa to ne velja, bi, ko bi želeli izračunati relacijo prvega rekurzivnega IPZ predikata, obstajalo pravilo s še nedosegljivo relacijo za nek njegov delni cilj.

Za interpretacijo rekurzivnih Datalog pravil bomo uporabili interpretacijo z dokazi. Iskanja množice veljavnih dejstev se namreč lahko lotimo tudi tako, da začnemo z EPZ dejstvi in pravili, iz katerih izpeljemo končno množico novih dejstev oblike $p(a_1, \dots, a_k)$, kjer je p IPZ predikat, vsebovan v pravilih, a_1, \dots, a_k pa konstante, zapisane v podatkovni zbirki. V naslednjih korakih z uporabo vedno več novih dejstev izpeljemo še več dejstev. Ker razpolagamo s končno množico dejstev in pravil, se ta postopek konča. Naj bodo torej R_1, \dots, R_k EPZ relacije, P_1, \dots, P_m pa še neizračunane IPZ relacije nekega rekurzivnega Datalog_{varno} programa. Množica dokazljivih dejstev za predikat p_i , pripadajoč IPZ relaciji P_i , je tedaj enaka

$$P_i = \text{EVAL}(p_i, R_1, \dots, R_k, P_1, \dots, P_m),$$

kjer je EVAL unija EVAL-RULE posameznih pravil za p_i (definiranih v algoritmu 1), projiciranih na spremenljivke iz glave pravila p_i . Na začetku so vse relacije P_j , $1 \leq j \leq m$, prazne, postopek pa se konča, ko nobeni relaciji ne dodamo novih dejstev. Enačbe, ki jih iz programa izpeljemo na opisan način, imenujemo *Datalog enačbe*.

Primer 6.10. *Vzemimo pravila iz primera 6.3, ki so del rekurzivnega Datalog_{varno} programa. Naj relacije RODITELJ, SESTRA_BRAT, SESTRICNA_BRATRANEC in SORODEN po vrsti pripadajo predikatom roditelj, sestra_brat, sestrična_bratranec in soroden. Pravila definirajo naslednje Datalog enačbe:*

$$\begin{aligned} \text{SESTRA_BRAT}(X, Y) = \\ \pi_{X,Y}(\sigma_{X \neq Y}(\text{RODITELJ}(X, Z) \bowtie \text{RODITELJ}(Y, Z))) \end{aligned}$$

$$\begin{aligned} \text{SESTRICNA_BRATRANEC} = \\ \pi_{X,Y}(\text{RODITELJ}(X, X_p) \bowtie \text{RODITELJ}(Y, Y_p) \bowtie \text{SESTRA_BRAT}(X_p, Y_p)) \cup \\ \pi_{X,Y}(\text{RODITELJ}(X, X_p) \bowtie \text{RODITELJ}(Y, Y_p) \bowtie \\ \text{SESTRICNA_BRATRANEC}(X_p, Y_p)) \end{aligned}$$

$$\begin{aligned} \text{SORODEN}(X, Y) = \\ \text{SESTRA_BRAT}(X, Y) \cup \\ \pi_{X,Y}(\text{SORODEN}(X, Z) \bowtie \text{RODITELJ}(Y, Z)) \cup \\ \pi_{X,Y}(\text{SORODEN}(Z, Y) \bowtie \text{RODITELJ}(X, Z)) \end{aligned}$$

V splošnem obstaja več rešitev danega sistema Datalog enačb. Rešitev sistema, ki glede na EPZ relacije R_1, \dots, R_k ustreza IPZ predikatom, pa imenujemo *negibna točka*. Za negibno točko P_1, \dots, P_m sistema Datalog enačb glede na EPZ relacije R_1, \dots, R_k velja,

da tvori skupaj z relacijami R_1, \dots, R_k model za dejstva in pravila, na katerih temelji dan sistem enačb [Ullman-1988]. Obratno, da je torej poljuben model za množico $\text{Datalog}_{\text{varno}}$ dejstev in pravil tudi negibna točka pripadajočih Datalog enačb, pa ne velja več. Model namreč lahko, poleg danih dejstev, izpolnjuje še dejstva, ki nastopajo le na levi strani enačb (primer 6.11).

V nadaljevanju se bomo osredotočili na negibne točke in *minimalne modele* $\text{Datalog}_{\text{varno}}$ programov, tj. modele, ki zadoščajo izključno dejstvom in pravilom danega programa. Izkaže se, da ima vsak $\text{Datalog}_{\text{varno}}$ program en sam minimalni model. Ta model vsebuje vse dane EPZ relacije in je glede na te relacije edina negibna točka pripadajočega sistema Datalog enačb. To negibno točko imenujemo *najmanjša negibna točka*.

Naj bodo P_1, \dots, P_m spremenljivke (relacije) sistema Datalog enačb, ki pripadajo IPZ predikatom p_1, \dots, p_m . Naj bodo R_1, \dots, R_k relacije EPZ predikatov r_1, \dots, r_k . Rešitev ali negibna točka glede na EPZ relacije R_1, \dots, R_k priredi spremenljivkam P_1, \dots, P_m relacije $P_1^{(1)}, \dots, P_m^{(1)}$, ki zadoščajo enačbam sistema. Naj bosta $\mathcal{S}_1 = P_1^{(1)}, \dots, P_m^{(1)}$ in $\mathcal{S}_2 = P_1^{(2)}, \dots, P_m^{(2)}$ dve rešitvi sistema enačb. Pravimo, da je $\mathcal{S}_1 \leq \mathcal{S}_2$, če je relacija $P_i^{(1)}$ podmnožica relacije $P_i^{(2)}$ za $1 \leq i \leq m$. Če pa je $\mathcal{S}_1 \leq \mathcal{S}_2$ in $\mathcal{S}_1 \neq \mathcal{S}_2$, pišemo $\mathcal{S}_1 < \mathcal{S}_2$. Negibno točko \mathcal{S}_0 imenujemo *najmanjša negibna točka* danega sistema Datalog enačb glede na EPZ relacije R_1, \dots, R_k , če za vsako rešitev \mathcal{S} velja $\mathcal{S}_0 \leq \mathcal{S}$. Če pa ne obstaja negibna točka \mathcal{S} , za katero bi veljalo $\mathcal{S} \leq \mathcal{S}_0$, negibno točko \mathcal{S}_0 imenujemo *minimalna negibna točka*.

Če obstaja najmanjša negibna točka, je ta točka edina minimalna negibna točka. Obstaja pa lahko več različnih minimalnih negibnih točk, ki niso v relaciji \leq . V tem primeru najmanjša negibna točka ne obstaja.

Primer 6.11. *Oglejmo si problem izračuna tranzitivnega zaprtja usmerjenega grafa. Definirajmo EPZ predikat povezava – dejstvo*

$$\text{povezava}(X, Y)$$

naj bo resnično natanko tedaj, ko obstaja usmerjena povezava iz vozlišča X v vozlišče Y . Sprehode v grafu lahko tedaj najnazorneje opišemo z naslednjima $\text{Datalog}_{\text{varno}}$ praviloma:

- (1) $\text{sprehod}(X, Y) : - \text{povezava}(X, Y)$.
- (2) $\text{sprehod}(X, Y) : - \text{sprehod}(X, Z) \ \& \ \text{sprehod}(Z, Y)$.

Prvo pravilo pravi, da sprehod lahko sestavlja ena sama povezava, drugo pa, da je sklop dveh sprehodov, kjer se končno vozlišče prvega in začetno vozlišče drugega sprehoda ujemata, sprehod, ki se začne v začetnem vozlišču prvega sprehoda in konča v končnem vozlišču drugega sprehoda.

Naj relacija POVEZAVA pripada predikatu povezava. Tedaj lahko pravili (1) in (2) pretvorimo v skupno Datalog enačbo za relacijo SPREHOD predikata sprehod:

$$\text{SPREHOD}(X, Y) = \text{POVEZAVA}(X, Y) \cup \pi_{X, Y}(\text{SPREHOD}(X, Z) \bowtie \text{SPREHOD}(Z, Y))$$

Vzemimo model danega Datalog_{varno} programa, v katerem množica konstant ustreza množici vozlišč $\{1, 2, 3\}$ in relacija POVEZAVA = $\{(1, 2), (2, 3)\}$ paru povezav $1 \rightarrow 2$ in $2 \rightarrow 3$. Poiščimo IPZ relacijo SPREHOD. Pravilo (1) pravi, da sta $(1, 2)$ in $(2, 3)$ elementa relacije SPREHOD, iz pravila (2) pa sledi $(1, 3) \in SPREHOD$. Opazimo, da je $SPREHOD = \{(1, 2), (1, 3), (2, 3)\}$ rešitev Datalog enačbe za relacijo SPREHOD. To je

$$\{(1, 2), (1, 3), (2, 3)\} = \{(1, 2), (2, 3)\} \cup \pi_{X,Y}(\{(1, 2), (1, 3), (2, 3)\} \bowtie \{(1, 2), (1, 3), (2, 3)\}).$$

Pri tem ne smemo pozabiti, da je levi operand naravnega spoja relacija z atributoma X in Z , desni operand pa relacija z atributoma Z in Y . Izraz $\pi_{X,Y}(SPREHOD(X, Z) \bowtie SPREHOD(Z, Y))$ je tedaj enak $\{(1, 3)\}$. Rezultat, ki smo ga dobili, predstavlja množico vseh dejstev, ki jih lahko izpeljemo. Ta model pa je tudi minimalni model danega Datalog_{varno} programa in SPREHOD je najmanjša negibna točka dane Datalog enačbe glede na EPZ relacijo POVEZAVA.

Rešitev $SPREHOD = \{(1, 2), (1, 3), (2, 3)\}$ pa ni edina rešitev dane Datalog enačbe. Recimo, da je tudi $(1, 1) \in SPREHOD$. Ker zaradi elementa $(1, 1)$ ne izpeljemo novih dejstev, velja $POVEZAVA = \{(1, 2), (2, 3)\}$ in $SPREHOD = \{(1, 1), (1, 2), (1, 3), (2, 3)\}$. Sledi še ena rešitev:

$$\{(1, 1), (1, 2), (1, 3), (2, 3)\} = \{(1, 2), (2, 3)\} \cup \pi_{X,Y}(\{(1, 1), (1, 2), (1, 3), (2, 3)\} \bowtie \{(1, 1), (1, 2), (1, 3), (2, 3)\})$$

Tudi vrednost relacije SPREHOD, ki bi vsebovala vseh devet parov (i, j) , $1 \leq i, j \leq 3$, bi zadoščala dani Datalog enačbi. Obstajajo pa vrednosti relacije SPREHOD, ki tej enačbi ne zadoščajo, kot na primer $SPREHOD = \{(1, 2), (1, 3), (2, 3), (3, 1)\}$. Tedaj velja

$$\{(1, 2), (1, 3), (2, 3), (3, 1)\} \neq \{(1, 2), (2, 3)\} \cup \pi_{X,Y}(\{(1, 2), (1, 3), (2, 3), (3, 1)\} \bowtie \{(1, 2), (1, 3), (2, 3), (3, 1)\}),$$

saj je vrednost izraza $\pi_{X,Y}(SPREHOD(X, Z) \bowtie SPREHOD(Z, Y))$ v tem primeru enaka $\{(1, 1), (1, 3), (2, 1), (3, 2), (3, 3)\}$, iz česar sledi neenakost obeh strani.

Vzemimo še model danega Datalog_{varno} programa s $POVEZAVA = \emptyset$ in $SPREHOD = \{(1, 2)\}$. Ker ne obstaja zamenjava, ki bi uresničila telo povezava(X, Y) pravila (1), je pravilo resnično za vsako zamenjavo spremenljivk. Za pravilo (2) ne obstaja konstanta, s katero bi lahko zamenjali spremenljivko Z , da bi uresničili oba delna cilja, $sprehod(X, Z)$ in $sprehod(Z, Y)$. Sledi, tudi pravilo (2) je resnično za vsako zamenjavo spremenljivk. Sklenemo lahko, da je množica dejstev, ki vsebuje le $SPREHOD(1, 2)$, model danega Datalog_{varno} programa. Kljub temu pa relaciji POVEZAVA in SPREHOD naši Datalog enačbi ne zadoščata,

$$\{(1, 2)\} \neq \emptyset \cup \pi_{X,Y}(\{(1, 2)\} \bowtie \{(1, 2)\}),$$

saj je vrednost izraza na desni enaka \emptyset . Sledi, $SPREHOD = \{(1, 2)\}$ ni negibna točka dane Datalog enačbe glede na EPZ relacijo $POVEZAVA = \emptyset$.

Pri reševanju sistema Datalog enačb, kjer so R_1, \dots, R_k relacije vsebovanih EPZ predikatov r_1, \dots, r_k , na začetku predpostavimo, da so vse spremenljivke P_1, \dots, P_m , ki pripadajo IPZ predikatom p_1, \dots, p_m , prazne relacije. Z uporabo operacije EVAL na trenutnih vrednostih IPZ relacij in nespreminjajočih se vrednostih EPZ relacij dobimo nove vrednosti IPZ relacij P_1, \dots, P_m . Ta postopek, opisan z algoritmom 3, nadaljujemo, dokler se nobena od IPZ relacij ne spremeni več.

Algoritem 3. IZRAČUN NAJMANJŠE NEGIBNE TOČKE DATALOG ENAČB.

VHOD: Datalog_{varno} program z EPZ predikati r_1, \dots, r_k , s pripadajočimi relacijami R_1, \dots, R_k in IPZ predikati p_1, \dots, p_m .

IZHOD: Najmanjša negibna točka sistema Datalog enačb glede na R_1, \dots, R_k , ki sledi iz vhodnega programa.

POSTOPEK: Poiščimo sistem Datalog enačb, ki pripada vhodnemu Datalog_{varno} programu. V sistemu enačb nastopajo spremenljivke P_1, \dots, P_m , ki ustrezajo IPZ predikatom p_1, \dots, p_m . Spremenljivki P_i tedaj pripada enačba

$$P_i = \text{EVAL}(p_i, R_1, \dots, R_k, P_1, \dots, P_m),$$

ki jo rešujemo iterativno:

```

za  $i = 1$  do  $m$  naredi
   $P_i := \emptyset$ ;
ponavljaj
  za  $i = 1$  do  $m$  naredi
     $Q_i := P_i$ ;
  za  $i = 1$  do  $m$  naredi
     $P_i := \text{EVAL}(p_i, R_1, \dots, R_k, Q_1, \dots, Q_m)$ ;
dokler  $P_i = Q_i$  za  $1 \leq i \leq m$ ;

```

Primer 6.12. Vzemimo pravila iz primera 6.3 in EPZ relacijo

$$\text{RODITELJ} = \{(c, a), (d, a), (d, b), (e, b), (f, c), (f, e), (g, c), (h, d), (i, d), (i, e), (j, f), (j, h), (k, g), (k, i)\},$$

ki pripada EPZ predikatu *roditelj*. Naj *SESTRA_BRAT*, *SESTRICNA_BRATRANEC* in *SORODEN* pripadajo IPZ predikatom *sestra_brat*, *sestrična_bratranec* in *soroden*.

Pripadajoč sistem Datalog enačb je zapisan v primeru 6.10. Tabela 6.1 prikazuje iterativno dodajanje elementov relacijam *SESTRA_BRAT*, *SESTRIČNA_BRATRANEC* in *SORODEN* v skladu z algoritmom 3.

	S1	S2	S3			
1	<i>cd</i> <i>de</i> <i>fg</i> <i>hi</i> <i>fi</i>					
2		<i>fh</i> <i>fi</i> <i>ii</i> <i>gh</i> <i>gi</i> <i>hi</i> <i>jk</i>	<i>cd</i> <i>de</i> <i>fg</i> <i>hi</i> <i>fi</i>			
3		<i>jj</i> <i>kk</i>	<i>df</i> <i>dg</i> <i>ch</i> <i>di</i> <i>ci</i> <i>eh</i> <i>ei</i> <i>gj</i> <i>fk</i> <i>hk</i> <i>ij</i>			
4			<i>fh</i> <i>dj</i> <i>gh</i> <i>jk</i> <i>gi</i> <i>dk</i> <i>cj</i> <i>ii</i> <i>ck</i> <i>ej</i> <i>ek</i>			
5			<i>fj</i> <i>hj</i> <i>gk</i> <i>ik</i>			
6			<i>jj</i> <i>kk</i>			

Tabela 6.1: Prikaz delovanja algoritma 3 na Datalog programu iz primera 6.3. Pri tem smo uporabili nekatere okrajšave: imena relacij *SESTRA_BRAT*, *SESTRIČNA_BRATRANEC* in *SORODEN* smo zamenjali z *S1*, *S2* in *S3*, urejene pare oblike (f, h) smo zapisali kot *fh*, zaradi simetričnosti vseh IPZ relacij pa smo prikazali le tiste elemente, katerih prva komponenta je po abecednem redu pred drugo komponento.

Na začetku algoritma 3 relacijam *SESTRA_BRAT*, *SESTRIČNA_BRATRANEC* in *SORODEN* priredi vrednost \emptyset . V 1. iteraciji zato dobimo naslednje delne rezultate:

$$\begin{aligned} \text{SESTRA_BRAT}_1(X, Y) &:= \text{EVAL}(\text{sestra_brat}, \text{RODITELJ}, \emptyset, \emptyset, \emptyset) = \\ &\pi_{X,Y}(\sigma_{X \neq Y}(\text{RODITELJ}(X, Z) \bowtie \text{RODITELJ}(Y, Z))) \end{aligned}$$

$$\begin{aligned} \text{SESTRIČNA_BRATRANEC}_1(X, Y) &:= \\ \text{EVAL}(\text{sestrična_bratranec}, \text{RODITELJ}, \emptyset, \emptyset, \emptyset) &= \\ \pi_{X,Y}(\text{RODITELJ}(X, X_p) \bowtie \text{RODITELJ}(Y, Y_p) \bowtie \emptyset(X_p, Y_p)) \cup \\ \pi_{X,Y}(\text{RODITELJ}(X, X_p) \bowtie \text{RODITELJ}(Y, Y_p) \bowtie \emptyset(X_p, Y_p)) &= \emptyset \end{aligned}$$

$$\begin{aligned} \text{SORODEN}_1(X, Y) &:= \text{EVAL}(\text{soroden}, \text{RODITELJ}, \emptyset, \emptyset, \emptyset) = \\ \emptyset(X, Y) \cup \pi_{X,Y}(\emptyset(X, Z) \bowtie \text{RODITELJ}(Y, Z)) \cup \\ \pi_{X,Y}(\emptyset(Z, Y) \bowtie \text{RODITELJ}(X, Z)) &= \emptyset \end{aligned}$$

Le relacija *SESTRA_BRAT*₁ ni prazna – njeni elementi so podani v tabeli 6.1. Ker je *EVAL* izraz za relacijo *SESTRA_BRAT* odvisen le od nespremenljajoče se *EPZ* relacije, sledi *SESTRA_BRAT* = *SESTRA_BRAT*₁.

V 2. iteraciji algoritma dobimo naslednje delne rezultate:

$$\begin{aligned} \text{SESTRICNA_BRATRANEC}_2(X, Y) &:= \\ \text{EVAL}(\text{sestrična_bratranec}, \text{RODITELJ}, \text{SESTRA_BRAT}, \emptyset, \emptyset) &= \\ \pi_{X, Y}(\text{RODITELJ}(X, X_p) \bowtie \text{RODITELJ}(Y, Y_p) \bowtie \text{SESTRA_BRAT}(X_p, Y_p)) \end{aligned}$$

$$\begin{aligned} \text{SORODEN}_2(X, Y) &:= \text{EVAL}(\text{soroden}, \text{RODITELJ}, \text{SESTRA_BRAT}, \emptyset, \emptyset) = \\ \text{SESTRA_BRAT}(X, Y) \end{aligned}$$

Iz Datalog enačbe za $\text{SESTRICNA_BRATRANEC}_2$ in $(c, d) \in \text{SESTRA_BRAT}$ dobimo na primer izjavo, da so vsi otroci od c (tj. f in g) sestrične oz. bratranca otrok od d (tj. h in i). Relacija $\text{SESTRICNA_BRATRANEC}$ torej vsebuje elemente (f, h) , (f, i) , (g, h) in (g, i) . Zaradi simetrije pa iz $(d, c) \in \text{SESTRA_BRAT}$ sledi, da relacija $\text{SESTRICNA_BRATRANEC}$ vsebuje tudi obrate naštetih elementov, namreč (h, f) , (i, f) , (h, g) in (i, g) .

V 3. iteraciji algoritma dobimo presenetljiv, a tehnično pravilen rezultat – da sta namreč j in k sama svoji sestrični oz. svoja bratranca. Za j je to posledica naslednjih elementov: $(f, h) \in \text{SESTRICNA_BRATRANEC}$, $(j, f) \in \text{RODITELJ}$ in $(j, h) \in \text{RODITELJ}$. Postopek se konča, ko se delni rezultat za relacijo SORODEN v 7. iteraciji ne spremeni več.

Algoritem 3 se ustavi in je pravilen, tj. rezultat je edina in zato najmanjša negibna točka sistema Datalog enačb, ki pripada vhodnemu Datalog_{varno} programu glede na EPZ relacije [Ullman-1988]. Ta najmanjša negibna točka predstavlja natanko množico vseh dokazljivih dejstev. Dobljene relacije skupaj z vhodnimi EPZ relacijami tedaj tvorijo minimalni model vhodnega Datalog_{varno} programa.

6.3.3 Interpretacija Datalog pravil z negacijo

Hornove stavke, ki smo jih poimenovali pravila, smo definirali kot logično disjunkcijo enega pozitivnega in enega ali več negativnih literalov, jih zapisali s sintakso programskega jezika Prolog in atomarne formule iz telesa pravila (tj. pozitivne literale) poimenovali delni cilji. Kljub temu da pravila z negativnimi delnimi cilji niso Hornovi stavki, bomo videli, da lahko vrsto idej, ki smo jih razvili za Hornove stavke, uporabimo tudi pri takih posplošenih pravilih. Logične programe, ki poleg pravih Hornovih stavkov vsebujejo še negacijo oz. pravila z negativnimi delnimi cilji, bomo imenovali Datalog^\neg programi.

Intuitivno se zdi, da pomen pravil, ki vsebujejo enega ali več negativnih delnih ciljev, dobimo tako, da relacijo telesa pravila (algoritem 1) izračunamo šele, ko komplementiramo relacije vseh negativnih delnih ciljev. To pa ni tako preprosto, saj komplement glede na neko univerzalno množico v svetu podatkovnih zbirk ni dobro definiran (razdelek 5.2). Zaradi narave običajnih univerzalnih množic (npr. \mathbb{N} ali \mathbb{R}) komplement končne relacije namreč pogosto ni končen, algoritma 1 pa pri pravilu z negativnimi delnimi cilji zato ni mogoče izvajati.

Izkaže se, da bodo pri določanju pomena pravil z negativnimi delnimi cilji ključne tiste spremenljivke, ki nastopajo v negativnih delnih ciljih, ne nastopajo pa v nobenem od

pozitivnih, običajnih (nevgrajenih) delnih ciljev. V primerih 6.13, 6.14 in 6.15 bomo razložili pomen nekaterih Datalog pravil z negacijo in sestavili seznam pogojev, ki jim Datalog pravila z negacijo morajo zadoščati, če naj so smiselna.

Primer 6.13. *Vzemimo Datalog program iz primera 6.3. Naj predikat pravi povezuje tiste pare oseb, ki so povezane s predikatom sestrična_bratranec in niso povezane s predikatom sestra_brat, tj. prave sestrične in bratrance. Pripadajoče Datalog[¬] pravilo se glasi:*

$$\text{pravi}(X, Y) : - \text{sestrična_bratranec}(X, Y) \ \& \ \neg \text{sestra_brat}(X, Y).$$

Predikatu pravi pripadajočo relacijo PRAVI izračuna izraz relacijske algebre

$$\text{SESTRICNA_BRATRANEC} - \text{SESTRA_BRAT},$$

ki je ekvivalenten izrazu

$$\text{SESTRICNA_BRATRANEC}(X, Y) \bowtie \overline{\text{SESTRA_BRAT}}(X, Y),$$

kjer je $\overline{\text{SESTRA_BRAT}}$ komplement relacije SESTRA_BRAT glede na neko univerzalno množico U , ki vsebuje vse elemente relacije SESTRICNA_BRATRANEC. Izraza sta ekvivalentna, ker je v primeru, ko se atributi obeh relacij ujemajo, naravni spoj enak preseku, presek relacije s komplementom druge relacije pa je enak njuni razliki. Za univerzalno množico U bi lahko vzeli množico tistih oseb, katerih imena nastopajo v vsaj enem od elementov relacije RODITELJ. Tedaj bi bila relacija $\overline{\text{SESTRA_BRAT}}$ enaka $U \times U - \text{SESTRA_BRAT}$.

Naslednji primer ponazarja težave, ki se pojavijo, ko obstaja spremenljivka, ki nastopa le v negativnem delnem cilju Datalog[¬] pravila.

Primer 6.14. *Vzemimo Datalog[¬] pravilo:*

$$\text{samski}(X) : - \text{moški}(X) \ \& \ \neg \text{poročen}(X, Y).$$

EPZ predikat moški naj označuje osebe moškega spola, atomarna formula poročen(X, Y) z EPZ predikatom poročen pa naj pomeni, da je X mož od Y .

Ena od smiselnih interpretacij zgornjega pravila je, da je X samski, če je moški in ne obstaja Y , s katerim bi bil X poročen. Naj bodo MOŠKI, POROČEN in SAMSKI relacije predikatov moški, poročen in samski. Če relacijo telesa danega Datalog[¬] pravila izračunamo, tako da relacijo MOŠKI(X) naravno spojimo s komplementom relacije POROČEN(X, Y), tj. množico neporočenih parov (X, Y), dobimo množico parov (X, Y), kjer je X moški in Y ni poročen z X . Ko to množico projiciramo še na X (algoritem 1), ugotovimo, da so samski tisti moški, ki niso hkrati poročeni z vsemi ljudmi tega sveta. To je, samski so tisti moški, za katere obstajajo osebe, ki z njimi niso poročene.

Med intuitivno razlago zgornjega $Datalog^-$ pravila in rezultatom, ki ga dobimo, če negacijo obravnavamo na običajen način (tj. kot komplement relacije), obstaja torej očitna razlika. Tej težavi se izognemo, tako da prepovemo spremenljivke, ki poleg negativnih delnih ciljev ne nastopajo v nobenem običajnem (nevgrajenem) delnem cilju. Ta omejitev pa ni tako močna, kot se zdi na prvi pogled, saj je pravila vedno mogoče preoblikovati tako, da spremenljivk z omenjeno lastnostjo ne vsebujejo. Pri tem negativni delni cilj $\neg q(X_1, \dots, X_n)$ interpretiramo kot množico brez elementov oz. določil spremenljivk X_1, \dots, X_n , katerih komponente, pripadajoče tistim spremenljivkam, ki nastopajo le v negativnih delnih ciljih, delni cilj $q(X_1, \dots, X_n)$ uresničijo. Zgornje $Datalog^-$ pravilo na primer ustrezno preoblikujemo tako, da iz relacije $POROČEN$ s projekcijo odstranimo Y . Rezultat je naslednji par pravil:

$$\begin{aligned} mož(X) & : - \text{poročen}(X, Y). \\ \text{samski}(X) & : - \text{moški}(X) \ \& \ \neg \text{mož}(X). \end{aligned}$$

Pomen nastopajočih predikatov izražajo njim pripadajoče relacije:

$$\begin{aligned} MOŽ(X) & = \pi_X(POROČEN(X, Y)) \\ \text{SAMSKI}(X) & = \text{MOŠKI}(X) - \text{MOŽ}(X) \end{aligned}$$

Oba izraza še združimo:

$$\text{SAMSKI}(X) = \text{MOŠKI}(X) - \pi_X(POROČEN(X, Y))$$

Torej, $Datalog^-$ pravila s spremenljivkami, ki nastopajo le v negativnih delnih ciljih, ne pa tudi v katerem od običajnih delnih ciljev, je treba ustrezno preoblikovati in se takih spremenljivk znebiti. Naslednji primer pa ponazarja težavo, ki se pojavi, če negativni delni cilji vsebujejo 'premalo' spremenljivk.

Primer 6.15. *Vzemimo $Datalog^-$ pravilo:*

$$\text{kupec}(X, Y) : - \text{želi}(X, Y) \ \& \ \neg \text{bankrot}(X).$$

Naj relaciji $\check{Z}ELI$ in $BANKROT$ pripadata EPZ predikatoma $\check{z}eli$ in $bankrot$, relacija $KUPEC$ pa IPZ predikatu $kupec$. Intuitivni pomen zgornjega pravila je, da obstaja verjetost, da bo X kupil Y , če si X predmet Y $\check{z}eli$ in ima dovolj denarja. Če relacijo telesa tega pravila izračunamo, tako da relacijo $\check{Z}ELI$ naravno spojimo s komplementom relacije $BANKROT$, lahko pravilo zapišemo z naslednjo enačbo:

$$\text{KUPEC}(X, Y) = \check{Z}ELI(X, Y) \ \bowtie \ \overline{BANKROT}(X)$$

Pri tem smo z $\overline{BANKROT}$ označili komplement relacije $BANKROT$, tj. vse osebe, ki imajo dovolj denarja. Kot v prejšnjem primeru je tudi ta izraz ekvivalenten razliki dveh končnih relacij:

$$KUPEC(X, Y) = \check{ZELI}(X, Y) - (BANKROT(X) \times \pi_Y(\check{ZELI}(X, Y)))$$

Ker ima relacija *BANKROT* en atribut manj kot relacija *ŽELI*, smo ji v enačbi za relacijo *KUPEC* manjkajoči atribut dodali. Za vrednostno množico spremenljivke *Y* smo vzeli množico $\pi_Y(\check{ZELI}(X, Y))$.

Zgornji izraz relacijske algebre lahko pretvorimo v Datalog[¬] pravila. Eno pravilo bomo rabili zaradi projekcije $\pi_Y(\check{ZELI}(X, Y))$, drugo za združitev obeh predikatov v skupno relacijo $BANKROT(X) \times \pi_Y(\check{ZELI}(X, Y))$, tretje pa za razliko dobljenih relacij. Torej:

$$\begin{aligned} \check{z}eljen(Y) &: - \check{z}eli(X, Y). \\ nemogo\check{c}e(X, Y) &: - bankrot(X) \ \& \ \check{z}eljen(Y). \\ kupec(X, Y) &: - \check{z}eli(X, Y) \ \& \ \neg nemogo\check{c}e(X, Y). \end{aligned}$$

Tudi dodajanje atributov ne reši vseh težav, ki se lahko pojavijo zaradi negativnih delnih ciljev. Izkaže se namreč, da sistem Datalog[¬] enačb, v nasprotju s sistemom Datalog enačb, pogosto nima najmanjše negibne točke. Obstaja lahko več minimalnih negibnih točk. Tako situacijo ponazarja primer 6.16.

Primer 6.16. *Vzemimo naslednji Datalog[¬] pravili:*

$$\begin{aligned} p(X) &: - \ r(X) \ \& \ \neg q(X). \\ q(X) &: - \ r(X) \ \& \ \neg p(X). \end{aligned}$$

Naj bosta *P* in *Q* relacij IPZ predikatov *p* in *q*, $R = \{1\}$ pa relacija EPZ predikata *r*. Tedaj zgornjemu programu pripadata Datalog[¬] enačbi $P = R - Q$ in $Q = R - P$. Z \mathcal{S}_1 označimo rešitev $P = \emptyset$ in $Q = \{1\}$, z \mathcal{S}_2 pa rešitev $P = \{1\}$ in $Q = \emptyset$.

Tedaj velja: $\mathcal{S}_1 \not\leq \mathcal{S}_2$ in $\mathcal{S}_2 \not\leq \mathcal{S}_1$. Še več, rešitev \mathcal{S} , za katero bi veljalo $\mathcal{S} < \mathcal{S}_1$ ali $\mathcal{S} < \mathcal{S}_2$, ne obstaja. Za tak \mathcal{S} bi namreč moralo veljati $P = Q = \emptyset$, kar ni rešitev danega sistema enačb. Torej, tako \mathcal{S}_1 kakor tudi \mathcal{S}_2 sta negibni točki, obe minimalni. Sistem Datalog[¬] enačb zato nima najmanjše negibne točke.

Da bi se izognili problemu obstoja več minimalnih negibnih točk, bomo dovoljevali le t. i. *plastovita pravila*. Datalog[¬] pravilo imenujemo plastovito pravilo, če za predikat *q* iz glave v telesu pravila ne obstaja predikat *p* negativnega delnega cilja, za katerega bi v grafu odvisnosti logičnega programa obstajal sprehod od *q* do *p*. Omejitev na plastovita pravila sicer ne zagotavlja obstoja najmanjše negibne točke (primer 6.17), dovoljuje pa smiselno izbiro minimalne negibne točke, ki jo nato pojmuje kot pomen plastovitega Datalog[¬] programa. Datalog[¬] programe, katerih pravila z negativnimi delnimi cilji so vsa plastovita, bomo označevali z Datalog[¬]_{plast}. Če bodo pravila tudi varna, bomo uporabili oznako Datalog[¬]_{varno,plast}.

Zgradba grafa odvisnosti logičnega programa se z negativnimi delnimi cilji ne spremeni. Če je $\neg q(X_1, \dots, X_n)$ negativni delni cilj, glava pravila pa vsebuje predikat *p*, tedaj v grafu odvisnosti obstaja povezava od vozlišča *q* do vozlišča *p*. Vsa nerekurzivna Datalog[¬] pravila so torej plastovita že po definiciji.

Primer 6.17. *Vzemimo Datalog_{plast}^- program:*

- (1) $p(X) : - r(X).$
- (2) $p(X) : - p(X).$
- (3) $q(X) : - s(X) \quad \& \quad \neg p(X).$

Vsa pravila tega programa so plastovita, saj za edini negativni delni cilj $\neg p(X)$ v grafu odvisnosti ne obstaja sprehod od IPZ predikata q do predikata p .

Naj relaciji $R = \{1\}$ in $S = \{1, 2\}$ pripadata EPZ predikatoma r in s , relaciji P in Q pa IPZ predikatoma p in q . Zgornjemu Datalog_{plast}^- programu tedaj pripadata enačbi $P = P \cup R$ in $Q = S - P$. Ena rešitev tega sistema je $P = \{1\}$ in $Q = \{2\}$. Imenujmo jo \mathcal{S}_1 . Druga rešitev je $P = \{1, 2\}$ in $Q = \emptyset$. Imenujmo jo \mathcal{S}_2 . Obe rešiti, \mathcal{S}_1 in \mathcal{S}_2 , sta minimalni. Sistem Datalog enačb zato nima najmanjše negibne točke. Kljub temu se zdi rešitev \mathcal{S}_1 bolj naravna.

Ker niso vsi Datalog^- programi plastoviti, potrebujemo algoritem, ki bo to lastnost preverjal in v primeru neplastovitega programa poiskal njemu ekvivalenten plastovit program. V ta namen definirajmo *plasti* predikatov kot največjo množico predikatov, za katero velja:

1. Če predikat p nastopa v glavi nekega pravila z negativnim delnim ciljem, ki vsebuje predikat q , tedaj q leži v nižji plasti kot p .
2. Če predikat p nastopa v glavi nekega pravila s pozitivnim delnim ciljem, ki vsebuje predikat q , tedaj p leži v plasti, ki je vsaj tako visoko kot plast predikata q .

Plasti predikatov nakazujejo vrstni red izračuna IPZ relacij, ki omogoča, da relacije negativnih delnih ciljev med izračunavanjem obravnavamo kot EPZ relacije. Za Datalog^- program bomo rekli, da *ima plasti*, če obstajajo plasti njegovih predikatov. Postopek razvrščanja predikatov v plasti opisuje algoritem 4.

Algoritem 4. PREVERJANJE PLASTOVITOSTI IN RAZVRSTITEV PREDIKATOV DATALOG^- PROGRAMA V PLASTI.

VHOD: Množica Datalog^- pravil.

IZHOD: Odločitev, ali so pravila plastovita. Če so, tudi plasti predikatov.

POSTOPEK: Najprej opišimo postopek razvrščanja predikatov v plasti. Na začetku vse predikate priredimo plasti 1. Pravila pregledujemo po vrsti. Naj pravilo s predikatoma p v glavi vsebuje negativni delni cilj s predikatoma q . Predikat p naj pripada plasti i , predikat q pa plasti j . Če je $i \leq j$, predikat p premestimo v plast $j + 1$. Če pa pravilo s predikatoma p v glavi vsebuje pozitivni delni cilj s predikatoma q , ki pripada plasti j , in je $i < j$, tedaj predikat p premestimo v plast j . Ta postopek formaliziramo takole:

za vsak predikat p **naredi**

$plast[p] := 1;$

ponavlja

za vsako pravilo r s predikatom p v glavi **začni**

za vsak negativni delni cilj pravila r s predikatom q **naredi**

$plast[p] := \max(plast[p], 1 + plast[q]);$

za vsak pozitivni delni cilj pravila r s predikatom q **naredi**

$plast[p] := \max(plast[p], plast[q]);$

dokler plasti se ne spremenijo ali zaporedna številka katere od plasti preseže število predikatov;

Če v Datalog⁻ programu ni negativnih delnih ciljev, se algoritem takoj ustavi. Odgovor je da in vsi predikati pripadajo isti plasti. Če pridemo do točke, ko algoritem ne spremeni nobene plasti več, so pravila plastovita. Odgovor je da, izhod iz algoritma pa so tudi zgrajene plasti. Če med postopkom razvrščanja pridemo do točke, ko nek predikat priredimo plasti, katere zaporedna številka presega skupno število vseh predikatov vhodnega Datalog⁻ programa, pravila niso plastovita. Odgovor je ne.

Primer 6.18. *Vzemimo Datalog⁻ pravili iz primera 6.16:*

- (1) $p(X) : - r(X) \ \& \ \neg q(X).$
- (2) $q(X) : - r(X) \ \& \ \neg p(X).$

Z algoritmom 4 poskušajmo predikate p , q in r razvrstiti v plasti. Na začetku vsi predikati pripadajo plasti 1. V prvi iteraciji glavne zanke predikat p zaradi pravila (1) premestimo v plast 2, predikat q pa zaradi pravila (2) premestimo v plast 3. V naslednji iteraciji predikat p zaradi pravila (1) premestimo v plast 4. Ta zaporedna številka plasti pa presega skupno število predikatov, ki je enako tri, zato program ni plastovit. Sklep je pravilen, saj je predikat q , ki nastopa v negativnem delnem cilju pravila (1) s predikatom p v glavi, odvisen od p , tj. v grafu odvisnosti obstaja sprehod od p do q .

Primer 6.19. *Vzemimo še Datalog⁻ pravila iz primera 6.17:*

- (1) $p(X) : - r(X).$
- (2) $p(X) : - p(X).$
- (3) $q(X) : - s(X) \ \& \ \neg p(X).$

Za ta pravila smo že ugotovili, da so plastovita. Z algoritmom 4 zato predikate p , q , r in s razvrstimo v plasti. Na začetku pripadajo vsi predikati plasti 1. V prvi iteraciji glavne zanke predikat q zaradi pravila (3) premestimo v plast 2. V naslednji iteraciji ne pride do nobenih sprememb, zato se algoritem ustavi. Izhod iz algoritma je odgovor da in dve plasti predikatov: p , r in s pripadajo plasti 1, q pa pripada plasti 2. Taki plasti sta smiselni, saj sta r in s EPZ predikata, p pa IPZ predikat, katerega relacijo

algoritem 3 izračuna iz enačb, pripadajočih praviloma (1) in (2). Dobljeno relacijo predikata p lahko tedaj obravnavamo kot EPZ relacijo. Ker q ni rekurziven, lahko z algoritmom 2 izračunamo še relacijo predikata q .

Algoritem 4 je pravilen – odločitev o tem, ali je vhodni Datalog^\neg program plastovit, je pravilna [Ullman-1988]. Velja tudi, da je Datalog^\neg program plastovit natanko tedaj, ko ima plasti.

Da bi lahko izračunali pomen Datalog^\neg pravil, bomo poleg plastovitosti zahtevali še varnost. Definicija varnih pravil se z uvedbo negativnih delnih ciljev ne spremeni, tj. negativni delni cilji na omejenost spremenljivk ne vplivajo.

Postopek izračuna IPZ relacij $\text{Datalog}_{\text{varno,plast}}^\neg$ programa bomo izvajali plast za plastjo. Začeli bomo z najnižjo. Če bo pravilo za IPZ predikat p iz plasti i vsebovalo delni cilj s predikatom q iz nižje plasti, bomo relacijo za q zaradi plastovitosti pravil že poznali. To bo ali EPZ relacija ali relacija, izračunana v koraku, pripadajočem plasti predikata q . Ker pravila za IPZ predikate iz plasti i ne vsebujejo negativnih delnih ciljev s predikati iz plasti i , pripadajoče $\text{Datalog}_{\text{varno,plast}}^\neg$ enačbe rešimo z algoritmom 3.

Edino vprašanje, ki je ostalo odprto, je, kako določiti končno relacijo negativnega delnega cilja $\neg q(X_1, \dots, X_n)$, ki jo bomo lahko obravnavali kot relacijo (pozitivnega) delnega cilja. V ta namen definirajmo DOM kot unijo termov, ki nastopajo v EPZ relacijah in/ali pravilih $\text{Datalog}_{\text{varno,plast}}^\neg$ programa. Ker v določitih, ki uresničijo telesa varnih pravil, nastopajo le termi iz EPZ relacij in pravil, lahko vrednosti posameznih komponent relacije poljubnega negativnega delnega cilja omejimo na množico DOM . Naj bo Q (že izračunana ali EPZ) relacija predikata q . Tedaj relacijo \bar{Q} negativnega delnega cilja $\neg q(X_1, \dots, X_n)$ definiramo kot

$$\underbrace{DOM \times \dots \times DOM}_n - Q.$$

Taka definicija relacij negativnih delnih ciljev pravil iz plasti i in uporaba algoritma 3 za izračun najmanjše negibne točke plasti i dajeta enak rezultat, kot bi ga dobili, če bi nam uspelo sistem enačb obravnavati z neskončno relacijo \bar{Q} vseh elementov, ki niso v Q . Algoritem 3 v IPZ relacije plasti i dodaja namreč le elemente, katerih vrednosti komponent so iz DOM .

Postopek izračuna minimalne negibne točke, pojmovane kot pomen $\text{Datalog}_{\text{varno,plast}}^\neg$ programa, opisuje algoritem 5.

Algoritem 5. IZRAČUN MINIMALNE NEGIBNE TOČKE $\text{DATALOG}_{\text{varno,plast}}^\neg$ ENAČB.

VHOD: $\text{Datalog}_{\text{varno,plast}}^\neg$ program z urejenimi pravili in EPZ relacijami.

IZHOD: IPZ relacije, ki tvorijo minimalno negibno točko $\text{Datalog}_{\text{varno,plast}}^\neg$ enačb, pripadajočih vhodnemu programu.

POSTOPEK: Z algoritmom 4 izračunamo plasti vhodnega $\text{Datalog}_{\text{varno,plast}}^-$ programa. Vrednostno množico DOM dobimo kot unijo konstant, ki nastopajo v pravilih, in projekcij EPZ relacij na posamezne komponente.

Izračun IPZ relacij bo potekal plast za plastjo v naraščajočem vrstnem redu. Ko bomo računali relacije plasti i , bodo IPZ relacije nižjih plasti že izračunane, EPZ relacije pa podane. Izračun IPZ relacij plasti i je razdeljen na več korakov:

1. Če (pozitivni) delni cilj poljubnega pravila plasti i vsebuje EPZ predikat ali IPZ predikat, ki pripada plasti, nižji od plasti i , njemu pripadajočo relacijo že poznamo.
2. Ker po definiciji plasti predikat iz negativnega delnega cilja poljubnega pravila plasti i pripada plasti, ki leži nižje od plasti i , njemu pripadajočo n -mestno relacijo Q že poznamo. Relacija \bar{Q} pripadajočega negativnega delnega cilja je tedaj enaka $DOM \times \cdots \times DOM - Q$, kjer DOM v produktu nastopa n -krat.
3. Relacije IPZ predikatov plasti i izračunamo z algoritmom 3, pri čemer vse relacije, ki smo jih izračunali v 1. in 2. koraku tega postopka, obravnavamo kot EPZ relacije.

Primer 6.20. *Vzemimo $\text{Datalog}_{\text{varno,plast}}^-$ pravila iz primera 6.17, za katera smo v primeru 6.19 poiskali plasti. Ugotovili smo, da predikati p , r in s pripadajo plasti 1, predikat q pa plasti 2. Naj relaciji R in S pripadata EPZ predikatoma r in s . Ker sta R in S enomestni relaciji ter pravila ne vsebujejo konstant, je vrednostna množica $DOM = \pi_1(R) \cup \pi_1(S) = R \cup S$.*

Z algoritmom 5 izračunajmo obe IPZ relaciji. Relacija P predikata p , tj. edinega IPZ predikata plasti 1, je enaka R . Izračunajmo še relacijo Q predikata q , tj. edinega IPZ predikata plasti 2. Ker pravilo (3) vsebuje negativni delni cilj $\neg p(X)$, moramo izračunati relacijo \bar{P} vseh elementov, ki bi lahko bili v Q , niso pa v P . Velja $\bar{P} = DOM - P = R \cup S - P$. Iz $P = R$ sledi $\bar{P} = S - R$. Z algoritmom 3 zdaj dobimo:

$$\begin{aligned}
 Q(X) &= S(X) \bowtie \bar{P}(X) \\
 &= S(X) \cap \bar{P}(X) \\
 &= S(X) \cap (S(X) - R(X)) \\
 &= S(X) - R(X)
 \end{aligned}$$

V primeru 6.17 smo videli, da lahko za dan logični program obstaja več minimalnih negibnih točk. Negibna točka, ki jo izračuna algoritem 5, v omenjenem primeru ustreza negibni točki \mathcal{S}_1 .

Rezultat algoritma 5 niso nujno izrazi relacijske algebre, kot smo jih dobili npr. v primeru 6.20. V splošnem, ko imamo opravka z rekurzivnimi pravili, takšni izrazi ne obstajajo.

Minimalno negibno točko, ki jo izračuna algoritem 5, imenujemo *idealna negibna točka* ali tudi *model Datalog $_{varno,plast}^{\neg}$ pravil*. Idealna negibna točka \mathcal{S} tvori skupaj z EPZ relacijami model Datalog $_{varno,plast}^{\neg}$ programa in ima naslednji lastnosti [Ullman-1988]:

1. Relacija poljubnega predikata p iz \mathcal{S} , ki pripada plasti 1, je podmnožica relacije predikata p iz poljubne negibne točke \mathcal{S}_1 danega sistema enačb, različne od \mathcal{S} .
2. Relacije predikatov iz \mathcal{S} , ki pripadajo plasti $i > 1$, so podmnožice relacij istih predikatov poljubne negibne točke \mathcal{S}_1 , ki se na relacijah vseh predikatov plasti, nižjih od i , ujema z \mathcal{S} .

Iz tega sledi pravilnost algoritma 5, to je, da je idealna negibna točka \mathcal{S} minimalna.

6.4 Predpostavka zaprtega sveta

Pomen izjavnega veznika \neg , ki ga uporabljamo v okviru Dataloga, temelji na naslednji predpostavki:

PREDPOSTAVKA ZAPRTEGA SVETA (PZS): Ko atomarne formule $p(a_1, \dots, a_n)$, kjer so a_1, \dots, a_n konstante, iz dejstev in pravil podatkovne zbirke ni mogoče izpeljati, lahko predpostavimo, da velja $\neg p(a_1, \dots, a_n)$.

Predpostavka zaprtega sveta je močno orodje za izpeljavo novih formul. Izkaže pa se, da vodi PZS v določenih okoliščinah do logičnega protislovja, zaradi česar je ne smemo privzeti. Preden podamo primer takega protislovja (primer 6.21), naštejmo potrebne pogoje, ki jih mora podatkovna zbirka izpolnjevati, da bo PZS logično konsistentna.

Recimo, da atomarne formule sestra_brat(tone, mojca) iz dejstev in pravil podatkovne zbirke ni mogoče izpeljati. Po predpostavki zaprtega sveta tedaj velja

$$\neg \text{sestra_brat}(\text{tone}, \text{mojca}).$$

Če za osebo z imenom mojca v podatkovni zbirki uporabljamo še ime mojca_novak in v podatkovni zbirki velja sestra_brat(tone, mojca_novak), pridemo do protislovja. Potrebna je naslednja predpostavka:

PREDPOSTAVKA ENOLIČNIH IMEN: Vsaka konstanta, zapisana v zbirki, ima edinstveno ime. Dve konstanti z različnima imenoma sta različni.

Atomarna formula $\neg \text{sestra_brat}(\text{tone}, \text{mojca})$ je resnična, če tone in mojca nimata skupnega roditelja. Če obstaja oseba z imenoma tone in mojca skupni roditelj, ki v podatkovni zbirki ni omenjen, pridemo do protislovja. Potrebna je še ena predpostavka:

PREDPOSTAVKA ZAPRTIH DOMEN: V podatkovni zbirki so zapisane vse konstante danega jezika.

Naj veljajo omenjene predpostavke, sicer že navedene v razdelku 6.1. Naj bo \mathcal{R} množica pravil (tj. vrste Hornovih stavkov brez negativnih delnih ciljev), E pa množica EPZ dejstev, pripadajočih EPZ relacijam iz \mathcal{R} . Naj bo I množica IPZ dejstev, izpeljivih z algoritmom 3. To je, $I \cup E$ je množica vseh elementov vseh relacij najmanjše negibne točke sistema Datalog enačb, ki pripada vhodnemu Datalog_{varno} programu. Naj bo J množica atomarnih formul oblike $\neg p(a_1, \dots, a_n)$, kjer je p predikat iz \mathcal{R} , a_1, \dots, a_n so konstante, dejstvo $p(a_1, \dots, a_n)$ pa ni v $I \cup E$. Tedaj je množica $I \cup E \cup J$ logično konsistentna in predpostavka zaprtega sveta je smiselna.

PZS torej v Datalogu brez integritetnih omejitev smemo privzeti. Ker pa logika kot podatkovni model omogoča veliko splošnejše načine sklepanja, včasih želimo obravnavati tudi izjave, ki niso Hornove oblike. Primer take izjave je $p(0) \vee p(1)$. V takem primeru PZS lahko vodi do protislovja, kot ponazarja primer 6.21.

Primer 6.21. *Vzemimo podatkovno zbirko z eno samo enomestno EPZ relacijo R , ki ji pripada predikat r , in praviloma:*

$$p(X) : - r(X).$$

$$p(0) \vee p(1)$$

Zadnje pravilo ni Hornov stavek, saj vsebuje dva pozitivna literala. Kljub temu ga lahko pišemo s sintakso Prologa:

$$p(0) : - \neg p(1).$$

Recimo, da R ne vsebuje elementov 0 in 1. Tedaj dejstev $p(0)$ in $p(1)$ ni mogoče izpeljati. Iz predpostavke zaprtega sveta zato sledi $\neg p(0)$ in $\neg p(1)$. Formule $\neg p(0)$, $\neg p(1)$ in $p(0) \vee p(1)$ pa vodijo do logičnega protislovja.

Pri posplošitvah predpostavke zaprtega sveta na izjave, ki niso Hornove oblike, se pogosto srečamo s težavo, kako učinkovito preveriti, ali izjava $\neg p(a_1, \dots, a_n)$ izpolnjuje pogoj predpostavke ali ne [Minker-1982].

6.5 Prednosti in slabosti Dataloga

Prednosti logike prvega reda kot teoretične osnove podatkovnih sistemov pred drugačnimi pristopi so naslednje [Das-1992]:

- ↪ Logika ima razumljivo semantiko.
- ↪ Logiko lahko uporabljamo kot enoten jezik za izražanje dejstev, pravil, poizvedb in integritetnih omejitev.
- ↪ Dobro razvita logična teorija lahko služi za reševanje vrste težav, s katerimi se srečujemo v podatkovnih zbirkah, vključno z ničelnimi vrednostmi in nedoločnimi podatki.

- ↪ Več dejstev lahko nadomestimo z enim samim pravilom, kar vodi do izrazno močnega in ekonomičnega okolja za modeliranje podatkov.
- ↪ Koncept deduktivne podatkovne zbirke posplošuje koncept relacijskih podatkovnih zbirk.

Datalog je, v nasprotju z relacijsko algebro, ki je proceduralen logični jezik, deklarativen logični jezik. Dejstvo, ali je izbran jezik deklarativen (kot sta npr. Prolog in SQL) ali proceduralen (kot sta npr. C in Pascal), pa pomembno vpliva na zgradbo celotnega podatkovnega sistema. Deklarativni jeziki so uporabnikom prijaznejši, a jih je težje implementirati kot proceduralne jezike, saj učinkovita implementacija deklarativnega jezika zahteva obsežno optimizacijo [Ullman-1989].

Res je tudi, da se Datalog do danes še ni uspel uveljaviti. Slabosti, ki mu jih podatkovni inženirji očitajo, so naslednje [Ceri-1989]:

- ↪ Aplikacij, ki so uspele izkoristiti celotno izrazno moč Dataloga, je zelo malo.
- ↪ Datalog ni zasnovan kot programski jezik, temveč kot teoretični zgled.
- ↪ Datalog ne doseže čiste deklarativne oblike. Pri obdelavi pravil sklepanja včasih zahteva kontrolo programerja. Uporaba Dataloga z negacijo, tj. oblike podatkovnega jezika Datalog, za katerega bomo videli (razdelek 7.2), da je tudi brez integritetnih omejitev močnejši od relacijske algebre, namreč temelji na predpostavki zaprtega sveta. Predpostavka zaprtega sveta dovoljuje izpeljavo negativnih dejstev iz množice Datalog stavkov, ne dovoljuje pa uporabe teh negativnih dejstev pri izpeljavi novih dejstev.
- ↪ Datalog sistemi so realizirani kot zaprti svetovi, ki ne komunicirajo z drugimi sistemi.

V poglavju 7 se bomo zato osredotočili na primerjavo zmogljivosti relacijske algebre in Dataloga.

7 Relacijska algebra in Datalog

7.1 Uvod

7.2 Od relacijske algebre do logike

7.3 Od logike do relacijske algebre

7.4 Primerjava Dataloga in relacijske algebre

Datalog z integritetnimi omejitvami je absolutno močnejši od relacijske algebre.

Ugotovimo, da Datalog brez negacije (in brez integritetnih omejitev) in relacijska algebra po izrazni moči nista primerljiva – obstajajo funkcije, izrazljive v relacijski algebri, ki niso izrazljive v Datalogu, in obratno.

Datalog z negacijo (in brez integritetnih omejitev) je po izrazljivosti močnejši od relacijske algebre. Množica funkcij, izrazljivih v relacijski algebri, je ekvivalentna množici funkcij, izrazljivih z nerekurzivnim Datalog programom, katerega pravila so varna, nerekurzivna in plastovita.

7.1 Uvod

Integritetne omejitve predstavljajo izjemno prednost deduktivnih podatkovnih modelov pred relacijskimi podatkovnimi modeli. Edina zanimiva primerjava je zato primerjava relacijske algebre in Dataloga brez integritetnih omejitev. Nadaljevanje poglavja se torej nanaša na Datalog brez integritetnih omejitev, česar ne bomo vedno znova poudarjali.

Na izraz relacijske algebre lahko gledamo kot na funkcijo, ki množici relacij priredi novo relacijo – vrednost funkcije. Podobno velja za Datalog program, ki z uporabo EPZ relacij dane podatkovne zbirke izračuna IPZ relacije iste zbirke. Pojavi se vprašanje, ali so funkcije, ki jih določata relacijska algebra na eni strani in Datalog program na drugi strani, enake.

Odgovor, ki ga bomo dobili, je, da Datalog brez negacije in relacijska algebra po izrazni moči nista primerljiva. Obstajajo funkcije, izrazljive v relacijski algebri, ki niso izrazljive v Datalogu, in obratno. Če dovolimo negacijo, je Datalog po izrazljivosti močnejši od relacijske algebre. Množica funkcij, izrazljivih v relacijski algebri, pa je ekvivalentna množici funkcij, izrazljivih z nerekurzivnim Datalog^{varno,plast} programom, katerega pravila so varna, nerekurzivna in plastovita. Pri tem spomnimo, da so vsa nerekurzivna pravila plastovita. Pregled izraznih moči različic Dataloga in relacijske algebre je podan v [Ullman-1988, Ceri-1989].

7.2 Od relacijske algebre do logike

Izraze relacijske algebre bomo poskušali pretvoriti v Datalog pravila. To ni težko za naslednje osnovne operacije relacijske algebre: unijo, razliko, kartezični produkt in projekcijo. Izjema je le operacija izbire s kompleksnim pogojem.

Izbiro, katere pogoj vsebuje \wedge , \vee ali \neg , imenujmo *sestavljena izbira*. Izbiro, ki teh operatorjev ne vsebuje, pa imenujmo *enostavna izbira*. Sestavljene izbire bomo razbili na zaporedja unij in enostavnih izbir. Postopek ima dva koraka [Ullman-1988]:

1. Vse negacije formule F z uporabo DeMorganovih zakonov

$$\neg(F_1 \wedge F_2) = (\neg F_1) \vee (\neg F_2) \quad \text{in} \quad \neg(F_1 \vee F_2) = (\neg F_1) \wedge (\neg F_2)$$

pomaknemo v notranjost operatorjev \wedge in \vee . Z zaporedno uporabo takih transformacij od leve proti desni in z zamenjavo $\neg\neg G = G$ sčasoma pridemo do točke, ko vse negacije ustrezajo primerjavam $X\theta Y$, kjer je θ eden od aritmetičnih operatorjev primerjav: $=$, $<$, $>$, \leq , \geq ali \neq . To je posledica dejstva, da lahko izraz $\neg X\theta Y$, kjer je θ aritmetična primerjava, zapišemo kot $X\theta'Y$, kjer je θ' aritmetična primerjava, nasprotna primerjavi θ . Na ta način, tj. z razbitjem formule F , odstranimo vse nastope operatorja \neg . Dobimo izbiro, ki je ekvivalentna začetni izbiri in ne vsebuje operatorja \neg .

2. Poljubnemu izrazu relacijske algebre priredimo izraz, ki izračuna isto relacijo in katerega vse izbire so enostavne, tj. oblike $\sigma_{X\theta Y}$, kjer sta X in Y ali spremenljivki ali konstanti, θ pa operator aritmetičnih primerjav. Postopek opisuje primer 7.1.

Primer 7.1. *Vzemimo izraz*

$$E = \sigma_{\neg(\$1=\$2 \wedge (\$1 < \$3 \vee \$2 \leq \$3))}(R).$$

Za poenostavitev formule, na kateri temelji dana izbira, uporabimo DeMorganova zakona:

$$\begin{aligned} \neg(\$1 = \$2 \wedge (\$1 < \$3 \vee \$2 \leq \$3)) &= \neg(\$1 = \$2) \vee \neg(\$1 < \$3 \vee \$2 \leq \$3) \\ &= \neg(\$1 = \$2) \vee (\neg(\$1 < \$3) \wedge \neg(\$2 \leq \$3)) \end{aligned}$$

Operatorje \neg pomaknimo v notranjost operatorjev \wedge in \vee . Dobimo nasprotno aritmetične primerjave in izraz:

$$E = \sigma_{\$1 \neq \$2 \vee (\$1 \geq \$3 \wedge \$2 > \$3)}(R)$$

Ker je zunanji operator enak \vee , po enem od algebrajskih zakonov iz razdelka 5.2.3 sledi

$$E = \sigma_{\$1 \neq \$2}(R) \cup \sigma_{\$1 \geq \$3 \wedge \$2 > \$3}(R).$$

Prvi argument unije je enostaven, drugega, katerega zunanji operator je \wedge , pa z ustreznim algebrajskim zakonom (razdelek 5.2.3) pretvorimo v kompozicijo dveh izbir. Dobimo rezultat:

$$E = \sigma_{\$1 \neq \$2}(R) \cup \sigma_{\$1 \geq \$3}(\sigma_{\$2 > \$3}(R))$$

Pokažimo, da je vsaka funkcija, ki je izrazljiva v relacijski algebi, izrazljiva tudi kot nerekurzivni $\text{Datalog}_{\text{varno,plast}}^{\neg}$ program. To je, da za poljuben algebrajski izraz obstaja nerekurzivni $\text{Datalog}_{\text{varno,plast}}^{\neg}$ program, katerega izhodna relacija za enega od predikatov ustreza vrednosti danega algebrajskega izraza.

Dokaz bo potekal z indukcijo na dolžino oz. strukturo algebrajskega izraza. Algebrajski izraz, ki ne vsebuje nobene operacije relacijske algebre, je lahko le EPZ relacija R , ki je dosegljiva brez vsakršnih pravil, ali pa množica relacijskih elementov. Če algebrajski izraz podaja množico elementov P , elementu (a_1, \dots, a_n) te množice ustreza $\text{Datalog}_{\text{varno,plast}}^{\neg}$ pravilo oblike

$$p(a_1, \dots, a_n).$$

Vzemimo algebrajski izraz E , ki vsebuje vsaj eno osnovno operacijo relacijske algebre. Pri tem spomnimo, da lahko vse ostale operacije, tj. presek, kvocient, spoj, naravni spoj in semispoj, ki smo jih definirali v poglavju 5, izrazimo z osnovnimi operacijami relacijske algebre. V skladu z zunanjo operacijo zato ločimo pet primerov:

1. $E = E_1 \cup E_2$. Po indukcijski predpostavki obstajata predikata e_1 in e_2 , definirana z nerekurzivnimi $\text{Datalog}_{\text{varno,plast}}^-$ pravili, katerih relaciji se ujemata z relacijama, ki ju določata izraza E_1 in E_2 . Naj bo dimenzija teh relacij n (dimenziji relacij E_1 in E_2 morata biti enaki, saj ju v nasprotnem primeru z operacijo unije ne bi mogli združiti). Izrazu E tedaj ustrezata pravili:

$$\begin{aligned} e(X_1, \dots, X_n) &: - e_1(X_1, \dots, X_n). \\ e(X_1, \dots, X_n) &: - e_2(X_1, \dots, X_n). \end{aligned}$$

Sledi, poljuben element relacije predikata e je ali element relacije predikata e_1 ali predikata e_2 ali obeh. O tem se prepričamo z algoritmom 2.

2. $E = E_1 - E_2$. Kot v 1. primeru predpostavimo, da sta E_1 in E_2 n -mestni relaciji. Po indukcijski predpostavki obstajata predikata e_1 in e_2 , katerih nerekurzivna $\text{Datalog}_{\text{varno,plast}}^-$ pravila določajo isti relaciji kot E_1 in E_2 . Tedaj določa pravilo

$$e(X_1, \dots, X_n) : - e_1(X_1, \dots, X_n) \ \& \ \neg e_2(X_1, \dots, X_n)$$

predikat e , katerega relacija se ujema z relacijo E . O tem se prepričamo z algoritmom 5.

3. $E = E_1 \times E_2$. Naj bosta e_1 in e_2 predikata, ki pripadata relacijama E_1 in E_2 . Predpostavimo, da sta dimenziji teh predikatov n oz. m . Predikat e relacije E tedaj določa naslednje pravilo:

$$e(X_1, \dots, X_{n+m}) : - e_1(X_1, \dots, X_n) \ \& \ e_2(X_{n+1}, \dots, X_{n+m}).$$

4. $E = \pi_{i_1, \dots, i_k}(E_1)$. Pri tem naj bo E_1 n -mestna relacija, $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$, e_1 pa predikat, katerega pravila definirajo relacijo E_1 . Pravilo za predikat e , ki ustreza relaciji E , se glasi:

$$e(X_{i_1}, \dots, X_{i_k}) : - e_1(X_1, \dots, X_n).$$

5. $E = \sigma_F(E_1)$. Predpostavimo lahko, da je F enostavna izbira, recimo $\$i\theta\j . Primer, ko je eden od operandov formule F konstanta, je podoben. Naj bo e_1 n -mestni predikat, katerega relacija se ujema z relacijo E_1 . Pravilo za predikat e se tedaj glasi:

$$e(X_1, \dots, X_n) : - e_1(X_1, \dots, X_n) \ \& \ X_i\theta X_j.$$

Danemu algebrajskemu izrazu je torej mogoče prirediti nerekurzivni $\text{Datalog}_{\text{varno,plast}}^-$ program oz. ekvivalentno relacijo, ki ustreza enemu od predikatov tega programa.

Primer 7.2. *Vzemimo algebrajski izraz*

$$\text{KUPEC}(X, Y) = \check{\text{ZELI}}(X, Y) - (\text{BANKROT}(X) \times \pi_Y(\check{\text{ZELI}}(X, Y)))$$

iz primera 6.15. Zunanja operacija je razlika z levim operandom $\check{ZELI}(X, Y)$ in desnim operandom

$$BANKROT_PAR(X, Y) = BANKROT(X) \times \pi_Y(\check{ZELI}(X, Y)).$$

Predikat levega operanda, ki je EPZ relacija, ne zahteva $Datalog_{varno,plast}^-$ pravil. Zunanja operacija desnega operanda je kartezični produkt z EPZ relacijo $BANKROT(X)$ kot levim operandom in $\pi_Y(\check{ZELI}(X, Y))$ kot desnim operandom. Z uporabo zgornje konstrukcije pretvorimo izraz $\pi_Y(\check{ZELI}(X, Y))$ v naslednje nerekurzivno $Datalog_{varno,plast}^-$ pravilo:

$$\check{z}eljen(Y) : - \check{z}eli(X, Y).$$

Naj predikatu $\check{z}eljen$ pripada relacija, ki se ujema z relacijo $\pi_Y(\check{ZELI}(X, Y))$. Pravilo za predikat $bankrot_par$ se sedaj glasi:

$$bankrot_par(X, Y) : - bankrot(X) \ \& \ \check{z}eljen(Y).$$

Sledi $Datalog_{varno,plast}^-$ pravilo za predikat $kupec$:

$$kupec(X, Y) : - \check{z}eli(X, Y) \ \& \ \neg bankrot_par(X, Y).$$

7.3 Od logike do relacijske algebre

Pokažimo še, da za vsako IPZ relacijo poljubnega nerekurzivnega $Datalog_{varno,plast}^-$ programa obstaja pripadajoč izraz relacijske algebre. To je, vsaka funkcija, izrazljiva z nerekurzivnim $Datalog_{varno,plast}^-$ programom, je izrazljiva tudi v relacijski algebri.

Naj bo \mathcal{R} množica varnih, nerekurzivnih $Datalog$ pravil z negacijo. Ker je \mathcal{R} množica nerekurzivnih pravil, lahko IPZ predikate p_1, \dots, p_n uredimo tako, da za povezave $p_i p_j$ iz grafa odvisnosti velja $i < j$. Če se torej p_i pojavi v delnem cilju pravila za p_j , je v tem urejenem zaporedju predikatov pred p_j . Osnovna ideja je zaporedna uporaba algoritma 2. Tako bomo za vsak IPZ predikat dobili ustrezen izraz relacijske algebre.

Še prej pa relacije R negativnih delnih ciljev (z uporabo algoritma 5) nadomestimo s komplementiranimi relacijami $\bar{R} = DOM^k - R$, kjer je k dimenzija relacije R , DOM pa množica vseh konstant, ki nastopajo v množici pravil \mathcal{R} in EPZ relacijah. Množica DOM je izrazljiva v relacijski algebri. To je unija množice konstant in projekcij EPZ relacij na posamezne komponente. Tudi konstrukcija v algoritmu 2 uporablja le osnovne operacije relacijske algebre. Ker lahko osnovne algebrajske operacije komponiramo, z indukcijo na dolžino urejenega zaporedja predikatov dokažemo, da posameznemu predikatu pripada relacija, podana z nekim izrazom relacijske algebre.

Primer 7.3. *Vzemimo naslednji $Datalog_{varno,plast}^-$ pravili:*

$$\begin{aligned} p(X) & : - \ r(X, Y) \ \& \ \neg s(Y). \\ q(Z) & : - \ s(Z) \ \& \ \neg p(Z). \end{aligned}$$

Naj bosta r in s EPZ predikata s pripadajočima relacijama R in S . Izpeljali bomo algebrajska izraza za relaciji P in Q , ki pripadata IPZ predikatoma p in q . Algebrajski izraz za DOM je unija enomestne relacije S in projekcij relacije R na njeno prvo in drugo komponento:

$$DOM = \pi_1(R) \cup \pi_2(R) \cup S$$

Ustrezen vrstni red IPZ predikatov je p, q .

Predikat p je definiran s prvim pravilom. Za prvi delni cilj vzamemo EPZ relacijo $R(X, Y)$, za drugi delni cilj pa enomestno relacijo $[DOM - S](Y)$. Izraz relacijske algebre, ki podaja relacijo predikata p , tedaj konstruiramo tako, da naravni spoj omenjenih relacij projiciramo na atribut X , ki nastopa v glavi pravila (algoritem 2). Dobimo:

$$P(X) = \pi_X(R(X, Y) \bowtie [DOM - S](Y))$$

Predikat q je definiran z drugim Datalog^{varno,plast} pravilom. Za prvi delni cilj vzamemo relacijo $S(Z)$, za drugega pa $[DOM - P](Z)$. Relacija Q je določena z izrazom $S(Z) \bowtie [DOM - P](Z)$. Ker je naravni spoj v tem primeru enak preseku, velja

$$Q(Z) = (S \cap [DOM - P])(Z).$$

Ker je S podmnožica DOM , je $Q(Z) = S(Z) - P(Z)$. Če vstavimo še izraz za $P(Z)$, dobimo

$$Q(Z) = S(Z) - \pi_Z(R(Z, Y) \bowtie [DOM - S](Y)).$$

Ker k naravnemu spoju relacij $R(Z, Y)$ in $[DOM - S](Z)$ prispevajo le tisti elementi DOM , ki so dobljeni iz druge komponente relacije R , lahko DOM v zgornjem izrazu zamenjamo še s $\pi_2(R)$.

7.4 Primerjava Dataloga in relacijske algebre

Naj bodo argumenti in rezultat funkcije $f(P_1, \dots, P_m)$ relacije,

$$\mathcal{S}_1 = P_1^{(1)}, \dots, P_m^{(1)} \quad \text{in} \quad \mathcal{S}_2 = P_1^{(2)}, \dots, P_m^{(2)}$$

pa dve določili spremenljivk funkcije f , za kateri velja $\mathcal{S}_1 \leq \mathcal{S}_2$. Torej, relacija $P_i^{(1)}$ je podmnožica relacije $P_i^{(2)}$ za $1 \leq i \leq m$. Če za poljubni določili \mathcal{S}_1 in \mathcal{S}_2 z omenjeno lastnostjo velja $f(\mathcal{S}_1) \subseteq f(\mathcal{S}_2)$, pravimo, da je f *monotona funkcija*.

Monotone funkcije se v teoriji relacijskih podatkovnih zbirk zelo pogosto pojavljajo, kar je posledica monotonosti štirih izmed petih osnovnih operacij relacijske algebre – unije, kartezičnega produkta, projekcije in izbire. Monotone osnovne operacije relacijske algebre, tj. vse osnovne operacije razen razlike, tedaj tvorijo *monotono podmnožico* relacijske algebre. V monotono podmnožico dodamo tudi operacije, izpeljive iz monotoni osnovnih operacij kot npr. naravni spoj.

Velja, da je množica funkcij, katerih domena in kodomena sta monotoni podmnožici relacijske algebre, enaka množici funkcij, izrazljivih z nerekurzivnimi Datalog programi

brez negacije. Datalog programi brez negacije (in dovoljeno rekurzijo) pa so monotoni [Ullman-1988].

Posledica je obstoj izrazov relacijske algebre kot na primer $R - S$, ki niso ekvivalentni nobenemu Datalog programu brez negacije. Relacija $R - S$ namreč ni monotona, saj dodajanje elementov, ki pripadajo relaciji R , v relacijo S zmanjša množico elementov relacije $R - S$. Podobno obstajajo funkcije, izrazljive v rekurzivnem Datalogu ali v rekurzivnem Datalogu brez negacije, ki v relacijski algebri niso izrazljive. Primer je tranzitivno zaprtje relacije kot npr. predikat *sprehod* iz primera 6.11.

8

Sodelujoči podatkovni sistemi

8.1 Uvod

8.2 Pomen integritetnih omejitev

8.3 Sodelujoče odgovarjanje

8.4 Tehnike sodelujočega odgovarjanja, ki jih podpirajo Datalog in dialog
mIPC Kripkejevi modeli

8.5 Razširitve relacijskega podatkovnega modela

Obravnavamo logične podatkovne modele z integritetnimi omejitvami, katerih namen je preprečiti vnos neustreznih podatkov in preveriti konsistentnost ekstenzionalne in intenzionalne podatkovne zbirke.

Predstavimo sodelujoče podatkovne sisteme, ki se na poizvedbo odzivajo z dodatnimi informacijami, katerih namen je odpraviti nejasnosti in razčistiti dvoumnosti ter pozitivno vplivati na doseganje uporabnikovih ciljev in namer.

Opišemo vseh pet uveljavljenih tehnik sodelujočega odgovarjanja: upoštevanje uporabnikovih prepričanj, želja in namer; vrednotenje domnev, vsebovanih v poizvedbah; odkrivanje in odpravljanje nesporazumov; oblikovanje intenzionalnih odgovorov ter posploševanje poizvedb in odgovorov. Rezultat zadnje strategije, imenovane tudi relaksacija poizvedb, je odgovor, ki zajema večjo količino sorodnih, relevantnih in/ali smiselnih podatkov.

8.1 Uvod

V poglavjih 6 in 7 smo obravnavali Datalog programe brez *integritetnih omejitev*, tj. Hornovih stavkov, ki so disjunkcija enega ali več negativnih literalov. Datalog programi so bili sestavljeni le iz Hornovih stavkov, imenovanih dejstva in pravila. Tretja oblika Hornovih stavkov, to so integritetne omejitve, pa v okviru podatkovnih zbirk ne igrajo nič manj pomembne vloge [Minker-1996, Godfrey-1998].

Integritetne omejitve (IO) vsebujejo znanje o svetu oz. semantično informacijo o podatkih podatkovne zbirke. Njihov namen je preprečiti vnos neustreznih podatkov in preveriti konsistentnost ekstenzionalne in intenzionalne podatkovne zbirke. Integritetna omejitev mora biti resnična po vsaki transakciji podatkovne zbirke, h katerim štejemo:

↪ *dodajanje izjave*,

↪ *brisanje obstoječe izjave* in

↪ *spreminjanje obstoječe izjave*, ki zajema brisanje obstoječe izjave in dodajanje spremenjene izjave.

Čeprav integritetne omejitve ne vplivajo na rezultat poizvedbe [Reiter-1978], lahko vplivajo na učinkovitost izračuna oz. iskanja odgovora. Prva, ki sta integritetne omejitve uporabila za učinkovitejše iskanje odgovora na poizvedbo, tj. za semantično optimizacijo poizvedbe, sta bila McSkimin in Minker [McSkimin-1977].

Pojem, tesno povezan s semantično optimizacijo poizvedbe, je *sodelujoče odgovarjanje* [Gaasterland-1992a]. Namen sodelujočega podatkovnega sistema je uporabnika seznaniti z vzrokom uspeha oz. neuspeha njegove poizvedbe. Ko namreč uporabnikova poizvedba ni uspešna, tj. sistem ne najde odgovora, uporabnik v splošnem ne more vedeti, zakaj se je to zgodilo. Možna sta dva razloga:

- podatkovna zbirka zahtevane informacije trenutno ne vsebuje ali
- podatkovna zbirka zahtevane informacije ne bo nikoli vsebovala.

Sodelujoče odgovarjanje lahko delno temelji na *relaksaciji poizvedb* [Gaasterland-1992b] kot imenujemo postopek spreminjanja oz. nadomeščanja neuspešne poizvedbe z njej sorodno, relevantno poizvedbo (razdelek 8.3).

Pojem integritetnih omejitev zaradi večjih predstavitvenih zmožnosti včasih razširimo na poljubne formule prvega reda [Godfrey-1998]. Posledica je manjša učinkovitost implementacije podatkovne zbirke. V nadaljevanju tega poglavja bodo integritetne omejitve poljubne formule prvega reda, pod pojmom *osnovne integritetne omejitve* pa bomo razumeli integritetne omejitve, kot smo jih definirali v poglavju 6, tj. disjunkcije enega ali več negativnih literalov.

8.2 Pomen integritetnih omejitev

Oglejmo si primer integritetne omejitve (primer 8.1) in primer uporabe integritetnih omejitev pri vnosu podatkov v podatkovno zbirko (primer 8.2).

Primer 8.1. *Vzemimo 5-mestno relacijo z relacijsko shemo*

USLUŽBENEC(IME, NASLOV, PLAČA, ODDELEK, STAROST)

s pripadajočim predikatom uslužbenec. Osnovno integritetno omejitev, ki pravi, da so vsi uslužbenci polnoletni oz. starejši od 17 let, zapišemo kot

$$\neg \text{uslužbenec}(I, N, P, O, S) \vee \neg(S \leq 17),$$

kjer so vse spremenljivke pod vplivom univerzalnega kvantifikatorja. Zapišimo jo še s sintakso Prologa:

$$: - \text{uslužbenec}(I, N, P, O, S) \ \& \ S \leq 17.$$

Primer 8.2. *Naj bo podatkovna zbirka sestavljena iz naslednjih EPZ dejstev in IPZ pravil:*

EPZ : študent(*andrej*).
 študent(*tina*).
 dodiplomski_študent(*andrej*).
 mentor(*robert, tina*).

IPZ : raziskovalec(*X*) : - študent(*X*) & ¬dodiplomski_študent(*X*).

Naj ima dana podatkovna zbirka eno samo integritetno omejitev, namreč:

$$IO : \forall X : (\text{raziskovalec}(X) \Rightarrow \exists Y : \text{mentor}(Y, X))$$

Ta integritetna omejitev pravi, da mora vsak študent raziskovalnega študija imeti mentorja. Preverimo konsistentnost podatkovne zbirke. Ker andrej ni študent raziskovalnega študija, ne potrebuje mentorja. Nasprotno, zaradi predpostavke zaprtega sveta (razdelek 6.4) je tina študentka raziskovalnega študija in njen mentor je robert. Torej, podatkovna zbirka je konsistentna oz. integritetna omejitev je resnična.

*Recimo, da želimo izvesti transakcijo vnosa dejstva študent(*mojca*). Pri vnosu tega dejstva zaradi pravila, ki ga dana podatkovna zbirka vsebuje, implicitno vnesemo tudi dejstvo raziskovalec(*mojca*). Ker pa *mojca* nima mentorja, tako spremenjena podatkovna zbirka ne ustreza integritetni omejitvi.*

Preverjanje resničnosti integritetnih omejitev je časovno zahtevna in kompleksna operacija [Das-1992]. Pri tem so odločilnega pomena sestava podatkovne zbirke, oblika

integritetnih omejitev in metoda, s katero integritetne omejitve vrednotimo. Pogosto je ugodneje integritetne omejitve obravnavati posebej ali pa jih celo odstraniti. Vsekakor pa mora taka transformacija podatkovne zbirke poroditi semantično ekvivalentno podatkovno zbirko, tj. odgovor začetne podatkovne zbirke na poljubno poizvedbo se mora ujemati z odgovorom transformirane podatkovne zbirke na isto poizvedbo. Najbolj znane tehnike vrednotenja integritetnih omejitev so opisane v [Das-1992, Godfrey-1998].

Poleg osnovnih integritetnih omejitev obstajajo še nekatere značilne oblike integritetnih omejitev:

↪ *Enoličnostna integritetna omejitev*

$$\forall X : (p(X, Z) \wedge (p(X, Y) \Rightarrow Y = Z))$$

izraža, da je poljuben element X neke vrednostne množice preko predikata p povezan z največ enim elementom Z neke druge vrednostne množice.

↪ *Obveznostna integritetna omejitev*

$$\forall X \exists Y : (p(X) \Rightarrow q(X, Y))$$

pravi, da je poljuben element X neke vrednostne množice preko predikata q povezan z vsaj enim elementom Y neke druge vrednostne množice. Primer take omejitve je IO iz primera 8.2.

↪ *Podtipska integritetna omejitev*

$$\forall X : (p(X) \Rightarrow q(X))$$

pravi, da je poljuben element X dane vrednostne množice tudi element neke druge vrednostne množice.

Integritetne omejitve so v deduktivnih podatkovnih zbirkah zapisane deklarativno in v istem logičnem jeziku kot dejstva in pravila. Vključitev takih omejitev v druge oblike podatkovnih zbirk, kot je na primer relacijska podatkovna zbirka, ponavadi temelji na (manj zaželenih) proceduralnih postopkih. Ker pa deduktivne podatkovne zbirke poleg EPZ relacij vsebujejo tudi IPZ relacije, katerih posledica je vrsta implicitnih transakcij (primer 8.2), je preverjanje resničnosti integritetnih omejitev v deduktivnih podatkovnih zbirkah časovno zahtevnejša in kompleksnejša operacija kot v relacijskih podatkovnih zbirkah.

Integritetne omejitve poleg svoje prvotne vloge preprečevanja vnosa neustreznih podatkov in preverjanja konsistentnosti podatkovne zbirke predstavljajo tudi osnovo *sodelujočih podatkovnih sistemov*, katerih cilj je sodelujoče odgovarjanje oz. sodelujoče odzivanje na poizvedbe.

8.3 Sodelujoče odgovarjanje

Sodelujoči podatkovni sistemi so informacijski sistemi, ki se na poizvedbo ne odzivajo le z odgovorom, ki poizvedbi strogo zadošča, temveč tudi z dodatnimi informacijami, katerih namen je odpraviti nejasnosti in razčistiti dvomnosti ter pozitivno vplivati na doseganje uporabnikovih ciljev in namer.

V deduktivnih podatkovnih zbirkah ločimo *podatke* in *znanje* – podatki so vsebovani v ekstenzionalni podatkovni zbirki, znanje pa v intenzionalni podatkovni zbirki in integritetnih omejitvah. Medtem ko odgovor običajnega podatkovnega sistema vsebuje le podatke, odgovor sodelujočega podatkovnega sistema vsebuje tudi znanje.

Uveljavljene tehnike sodelujočega odgovarjanja [Gaasterland-1992a, Godfrey-1998] lahko glede na njihove sposobnosti razvrstimo v pet kategorij:

1. *upoštevanje uporabnikovih prepričanj, želja in namer,*
2. *vrednotenje domnev, vsebovanih v poizvedbah,*
3. *odkrivanje in odpravljanje nesporazumov,*
4. *oblikovanje intenzionalnih odgovorov,*
5. *posploševanje poizvedb in odgovorov.*

V nadaljevanju bomo prve štiri tehnike sodelujočega odgovarjanja opisali le na kratko, zadnjo, tj. posploševanje poizvedb in odgovorov, pa si bomo ogledali nekoliko podrobneje.

8.3.1 Upoštevanje uporabnikovih prepričanj, želja in namer

Za izvorno delo sodelujočega odgovarjanja velja knjiga *Elements of Discourse Understanding*, ki so jo leta 1981 napisali Joshi, Webber in Sag [Joshi-1981]. Knjiga temelji na člankih, ki so bili leta 1978 predstavljeni na delavnici *Computational Aspects of Linguistic Structure and Discourse Setting*. Problemi, s katerimi se ukvarjajo, se v glavnem nanašajo na sposobnost podatkovnih sistemov upoštevati uporabnikova prepričanja, želje in namere. Delimo jih v tri kategorije:

1. *Razumevanje vprašanj in odgovorov:* Vprašanje, ki ga uporabnik zastavi v naravnem jeziku, ima lahko več različnih pomenov. Razumevanje in pravilna interpretacija zastavljenih vprašanj sta zato ključna za uspešen dialog med uporabnikom in podatkovnim sistemom [Allen-1994].
2. *Sledenje dialogu:* To področje preučuje modeliranje poteka dialoga med uporabnikom in sistemom. Podatkovni sistem naj bi bil sposoben slediti spremembam uporabnikovih ciljev in se prilagajati njegovim nameram [Kowtko-1992, Carletta-1996a, Stolcke-2000].

3. *Modeliranje uporabnikovih prepričanj*: Podatkovni sistem naj bi bil sposoben napovedovati uporabnikova prepričanja in z uporabo dodatnih informacij preprečevati zmote [Bratman-1988]. Glej primer 8.3.

Primer 8.3. *Vzemimo vprašanje:*

“Ali je Matej profesor?”

Glede na prepričanje uporabnikov, da je večina profesorjev zaposlenih, je v primeru, da Matej ni zaposlen, v določenem kontekstu smiselno in sodelujoče to informacijo dodati. Torej je odgovor

“Da, vendar ni zaposlen.”

primernejši od jedrnatega odgovora

“Da.”

Taka oblika sodelujočega odgovarjanja temelji na modeliranju uporabnikovih prepričanj.

8.3.2 Vrednotenje domnev, vsebovanih v poizvedbah

Cilj tehnike vrednotenja domnev, vsebovanih v poizvedbah, je iz poizvedbe izluščiti domneve in v odgovor vključiti razlago, ki morebitne napačne domneve odpravi ter prepreči nastanek nadaljnjih napačnih domnev. Pri tem na domneve, vsebovane v poizvedbi, gledamo kot na izjave, katerih resničnost je pogoj za obstoj odgovora. Če je katera koli domneva napačna, poizvedba ni smiselna. Primer take nesmiselne poizvedbe podaja primer 8.4.

Primer 8.4. *Vzemimo poizvedbo*

“Kdo je leta 2004 opravil izpit iz Teorije množic?”

Če leta 2004 predavanj iz Teorije množic ni bilo, je poizvedba neuspešna in se jedrnat odgovor sistema glasi

“Nihče.”

Ker je bila domneva o izvajanju predmeta Teorija množic v letu 2004 napačna, poizvedba ni bila smiselna in bi si ob danem jedrnatem odgovoru uporabnik lahko ustvaril nadaljnje neutemeljene in/ali napačne domneve, kot npr. da je Teorija množic zelo težak predmet, tako da izpita ne opravi skoraj nihče. Sodelujoče bi bilo zato dodati razlog neuspeha prve poizvedbe, namreč neizvajanje predmeta Teorija množic v letu 2004.

8.3.3 Odkrivanje in odpravljanje nesporazumov

Na poizvedbo ponavadi pričakujemo odgovor, prazen odgovor na neuspešno poizvedbo pa pogosto porodi dvoumnost. Poizvedbe, katerih neuspeh je zaradi nekonsistentnosti s semantiko podatkovne zbirke neizogiben, kažejo na nesporazum uporabnika in podatkovnega sistema. Tako poizvedbo, ki ob dani podatkovni zbirki in nespremenjeni semantiki ne bo nikoli uspešna, pogosto imenujemo kar *nesporazum*.

Primer 8.5. *Vzemimo podatkovno zbirko, v kateri so shranjena imena učiteljev, šolskih predmetov in razredov, v katerih učitelji poučujejo. Poizvedba*

“Kateri učitelji poslušajo predmet Matematika?”

oziroma

učitelj(X) & posluša(matematika, X)?

bo neuspešna, saj učitelji predmete ponavadi le poučujejo, ne pa tudi poslušajo. To je, podatkovna zbirka vsebuje naslednjo osnovno integritetno omejitev:

$\neg \text{učitelj}(X) \vee \neg \text{posluša}(Y, X)$

Podatkovni sistem bo bolj informativen in sodelujoč, če bo namesto praznega odgovora

“Nobeden.”

podal razlog za tak nesporazum. Na primer:

“Učitelji predmete poučujejo, učenci jih poslušajo.”

Tehniki odkrivanja in odpravljanja nesporazumov sta se podrobno posvetila Gal in Minker [Gal-1987]. Razvila sta postopek testiranja nesporazumov, temelječ na semantični optimizaciji, ki preverja, ali kateri od pogojev dane poizvedbe ruši integriteto podatkovne zbirke. Taki pogoji so razlog za neuspeh poizvedbe. Odgovor, ki poda razlog neuspeha poizvedbe, je vsekakor bolj informativen in sodelujoč kot prazen odgovor.

8.3.4 Intenzionalni odgovori

Primer sodelujočega odgovora je tudi t. i. *intenzionalni odgovor*, katerega prednost je velika informativnost [Cholvy-1986, Cholvy-1990]. Intenzionalni odgovori pogosto ponazarjajo zgradbo podatkovne zbirke ter sestavo nastopajočih vrednostnih množic in zato nemalokrat vodijo do razrešitve nesporazumov. Velja, da je intenzionalni odgovor

primeren odgovor na poizvedbo, če uporabniku zadostno in razumljivo razloži vsebino eksaktnega odgovora, tj. jedrnatega odgovora na začetno poizvedbo.

Zelo pomembna lastnost nekaterih intenzionalnih odgovorov, ki pride do izraza predvsem pri obsežnih podatkovnih zbirkah, kjer so uporabniki pogosto preobremenjeni z obsegom eksaktnih odgovorov, je njihova jedrnatost. Ta jedrnatost temelji na integritetnih omejitvah, ki zagotavljajo brezpogojno resničnost nekaterih poizvedb. Glej primer 8.6.

Primer 8.6. *Vzemimo podatkovno zbirko iz primera 8.5. Rezultat poizvedbe*

“Kateri učitelji poučujejo?”

oziroma

učitelj(X) & poučuje(Y, X)?

bo seznam vseh učiteljev, vsebovanih v podatkovni zbirki, in njim pripadajočih šolskih predmetov. Podatkovna zbirka naj vsebuje naslednjo obveznostno integritetno omejitev:

$$\forall X \exists Y : (\text{učitelj}(X) \Rightarrow \text{poučuje}(Y, X))$$

Podatkovni sistem bo bolj jedrnat in sodelujoč, če bo namesto naštevanja vseh učiteljev v podatkovni zbirki podal intenzionalni odgovor

učitelj(X).

v obliki razlage brezpogojne resničnosti poizvedbe. Na primer:

“Vsi učitelji poučujejo.”

Če se sistem ne odzove na ta način, bo uporabnika zavedel, saj ta zelo verjetno ne bo opazil, da so v odgovoru naštetni vsi učitelji.

Oglejmo si še en primer intenzionalnega odgovora.

Primer 8.7. *Naj bo podatkovna zbirka sestavljena iz naslednjih EPZ dejstev in IPZ pravil o dodatkih k plačam uslužbencev nekega podjetja:*

*EPZ : zaposlen(andrej, 1990).
zaposlen(tina, 1997).*

zaposlen(matej, 2001).
pohvala(robert).
pohvala(maja).

IPZ : dodatek(X) : – zaposlen(X, Y) & danes(D) &
trajanje(D, Y, Z) & Z ≥ 10.
dodatek(X) : – pohvala(X).

Vzemimo poizvedbo

“Kateri uslužbenci dobijo dodatek k plači?”

oziroma

dodatek(X)?

Leta 2004 se eksakten odgovor glasi takole:

dodatek(andrej).
dodatek(robert).
dodatek(maja).

Našteti uslužbenci pa dodatka k plači ne dobivajo iz istega razloga. Uslužbenec Andrej, ki se je zaposlil leta 1990, dodatek dobiva zaradi več kot desetletne zaposlitve, Robert in Maja pa zaradi pohval, ki sta si ju zaslužila za svoje delo. Primer intenzionalnega odgovora, ki parafrazira obe IPZ pravili, se glasi takole:

“Vsi, ki so v podjetju zaposleni vsaj 10 let.”
“Vsi, ki so bili za svoje delo pohvaljeni.”

Tak odgovor je sodelujoč in bolj infomativen kot eksakten odgovor, saj namesto naštevanja zaposlenih raje poda vzrok izplačila dodatka k plači.

8.3.5 Posploševanje poizvedb in odgovorov

Naslednja strategija oblikovanja sodelujočega odgovora temelji na posploševanju oz. razširjanju obsega poizvedb. Namen te strategije, ki ne temelji na integritetnih omejitvah, je v odgovor zajeti večjo količino sorodnih, relevantnih in/ali smiselnih podatkov. Uporabljamo jo, ko je poizvedba neuspešna ali ko uporabnik v množici odgovorov ne najde sprejemljivega podatka.

Cuppens in Demolombe [Cuppens-1988] poizvedbo razčlenita na:

- *predikat poizvedbe*, tj. predikat, ki naj bo v odgovoru vsebovan,

- *pogoj poizvedbe*, tj. pogoj, ki naj ga parametri predikata poizvedbe izpolnjujejo, in
- *atributi poizvedbe*, tj. množica atributov predikata poizvedbe, ki naj jih odgovor zajema.

Na osnovi te razčlenbe ločimo več oblik transformacije poizvedb:

1. *transformacija predikata* – predikat poizvedbe zamenjamo z njemu sorodnim predikatom, npr.

$$\text{oče}(X, \text{darja})? \longmapsto \text{roditelj}(X, \text{darja})?$$

2. *transformacija pogoja* – spremenimo vrednost katerega od atributov, npr.

$$\text{sestra_brat}(\text{matej}, \text{darja})? \longmapsto \text{sestra_brat}(\text{matej}, X)?$$

$$\text{polet}(X, \text{ljubljana}, \text{london}, 7:00)? \longmapsto \text{polet}(X, \text{ljubljana}, \text{london}, \text{zjutraj})?$$

3. *transformacija izbora atributov* – v množico atributov poizvedbe dodamo ali iz nje izbrišemo kakšen atribut, npr.

$$\text{polet}(X, \text{ljubljana}, \text{london}, Y)? \longmapsto \text{polet}(X, \text{ljubljana}, \text{london}, Y, Z)?$$

Če želimo sodelujoč odgovor, začetno poizvedbo na izbrane načine transformiramo, da dobimo množico novih, bližnjih poizvedb. Razširjen odgovor tedaj poleg eksaktnega odgovora zajema tudi odgovore na izbrane bližnje poizvedbe.

Transformacija poizvedb, ki jo imenujemo tudi *relaksacija poizvedb* [Gaasterland-1992b], temelji na posebni urejenosti predikatov in konstant dane podatkovne zbirke. Najpogosteje uporabljena je relacija sosednosti, imenovana *ISA relacija*, s katero predikate in konstante delno uredimo (refleksivnost, antisimetričnost in tranzitivnost). Ime relacije se nanaša na angleško besedno zvezo *is a*. Relacija ISA je disjunktna unija relacije ISA_P nad predikati in relacije ISA_K nad konstantami podatkovne zbirke. Chu, Chen in Lee [Chu-1990] so urejenostno strukturo ISA relacije poimenovali *hierarhija abstrakcije tipov*.

Urejenost enomestnih predikatov, s katerimi opisujemo tipe konstant podatkovne zbirke, lahko izrazimo s Hornovimi stavki. Da sta predikat p in predikat q v relaciji ISA_P , lahko s sintakso programskega jezika Prolog zapišemo takole:

$$q(X) : - p(X).$$

Na enak način izrazimo urejenost večmestnih predikatov (primer 8.8). Izjave logike prvega reda, ki opisujejo z relacijo ISA delno urejeno množico predikatov in konstant podatkovne zbirke in vsebujejo natanko en *ISA predikat* (tj. predikat iz telesa izjave, ki je s predikatom iz glave v relaciji ISA), imenujemo *klasifikacijske izjave*. ISA

predikat klasifikacijske izjave imenujemo *ključni predikat* in ga označimo z indeksom $*$, ostale predikate telesa pa imenujemo *neključni predikati*. Klasifikacijska izjava izraža ISA relacijo med predikatom glave in ključnim predikatom izjave. Neključni predikati izražajo odnose med spremenljivkami izjave.

Klasifikacijske izjave, ki opisujejo urejenost enomestnih predikatov, imenujemo *ISA izjave*. Telo ISA izjave vsebuje natanko en literal. Pripadajoč predikat je ključni predikat.

Za predikat p , ki pripada relaciji predikata q , projicirani na podmnožico X_1, \dots, X_n atributov Y_1, \dots, Y_m , velja $p \text{ ISA}_P q$. To pa pomeni, da obstaja klasifikacijska izjava

$$q(Y_1, \dots, Y_m) : - p(X_1, \dots, X_n)^*.$$

Primer 8.8. *Recimo, da želimo opisati načine sporazumevanja med dvema oseba. V ta namen definirajmo tromestni predikat dialog in njemu podrejene predikate stik, agent, pošta, e_pošta in telefon, ki po vrsti opisujejo sporazumevanje iz oči v oči, sporazumevanje prek tretje osebe – agenta, sporazumevanje po navadni pošti, sporazumevanje po elektronski pošti in sporazumevanje po telefonu. Podrejenost predikatov, ki opisujejo način sporazumevanja, naj pomeni relacijo ISA_P s predikatom dialog:*

$$\text{ISA}_P = \{(stik, dialog), (agent, dialog), (pošta, dialog), (e_pošta, dialog), (telefon, dialog)\}$$

Pripadajoče klasifikacijske izjave so naslednje:

$$\begin{aligned} dialog(O_1, O_2, X) & : - stik(O_1, O_2, X)^*. \\ dialog(O_1, O_2, X) & : - agent(O_1, O_2, X)^*. \\ dialog(O_1, O_2, X) & : - bivališče(Naslov, O_2) \& pošta(O_1, Naslov, X)^*. \\ dialog(O_1, O_2, X) & : - e_naslov(ENaslov, O_2) \& e_pošta(O_1, ENaslov, X)^*. \\ dialog(O_1, O_2, X) & : - tel_številka(TelŠt, O_2) \& telefon(O_1, TelŠt, X)^*. \end{aligned}$$

Čeprav so vsi nastopajoči ISA predikati tromestni, imajo različne domene – drugi parameter predikatov stik in agent je oseba, drugi parameter predikatov pošta in e_pošta je naslov, drugi parameter predikata telefon pa je telefonska številka.

V postopku relaksacije bomo dejstvo

$$p(c).$$

zapisovali kot

$$p(X) : - c \text{ enak } X.$$

kjer bomo literal c enak X imenovali *izenačitveni literal*. Dvomestni predikat enak, ki je logično ekvivalenten predikatu $=$, uporabljamo zaradi enostavnejšega razločevanja

izenačitvenih literalov in literalov, vsebovanih v začetni poizvedbi. Tudi literal oblike X enak Y , kjer sta X in Y spremenljivki, bomo imenovali *izenačitveni literal*.

V postopku relaksacije poizvedbo posplošimo, tako da predikate in konstante zamenjamo s predikati in konstantami, ki v Hassejevem diagramu ISA relacije (tj. diagramu delne urejenosti) ležijo nad njimi. Torej, literal, katerega predikat nastopa v vlogi ključnega predikata katere od klasifikacijskih izjav, zamenjamo z glavo pripadajoče klasifikacijske izjave. Preden pa poizvedbo podvržemo postopku relaksacije, jo z naslednjim postopkom zamenjave spremenljivk preoblikujemo, tako da omogoča neposredno primerjavo literalov iz poizvedbe in ključnih literalov iz klasifikacijskih izjav:

1. Vsako konstanto, ki nastopa v poizvedbi, zamenjamo z novo spremenljivko in poizvedbi dodamo ustrezen izenačitveni literal. To je, če je c konstanta iz poizvedbe, ki jo zamenjamo s spremenljivko X , poizvedbi dodamo izenačitveni literal

c enak X .

2. Za vsako spremenljivko, ki v poizvedbi nastopa večkrat, ohranimo le po eno kopijo, ostale pa zamenjamo z novimi, paroma različnimi spremenljivkami, ki jih z uporabo izenačitvenih literalov izenačimo z originalom. To je, če eno kopijo spremenljivke X zamenjamo s spremenljivko Y , poizvedbi dodamo izenačitveni literal

X enak Y .

3. Odvečne literale, tj. podvojene literale ali literale, ki ne nosijo nove informacije, odstranimo.

Rezultat take zamenjave spremenljivk je poizvedba, v kateri edino izenačitveni literali vsebujejo konstante in izražajo medsebojne odvisnosti spremenljivk. Postopek ponazarja primer 8.9.

Primer 8.9. *Vzemimo poizvedbo*

$brat(miha, X) \ \& \ oče(X, Y)?$

ki sprašuje po bratih X ter nečakah in nečakinjah Y osebe z imenom miha. V postopku zamenjave spremenljivk, opisanem zgoraj, dobimo naslednjo preoblikovano poizvedbo:

$brat(A, X) \ \& \ oče(B, Y) \ \& \ miha \ enak \ A \ \& \ X \ enak \ B?$

Relaksacija poizvedbe Q , dobljene v postopku zamenjave spremenljivk, poteka tako, da za vsak predikat iz Q poiščemo klasifikacijske izjave, v katerih ta nastopa kot ključni predikat. V naslednjem koraku izbran literal zamenjamo z glavo pripadajoče klasifikacijske izjave (primer 8.11) in vse neključne literale iz telesa klasifikacijske izjave dodamo novi poizvedbi Q' . Ti neključni literali opisujejo odnose med spremenljivkami.

Če želimo relaksacijo implementirati na proceduralen način, v podatkovni model vključimo tudi t. i. *obratne klasifikacijske izjave*. Obratna klasifikacijska izjava dane klasifikacijske izjave \mathcal{C} je izjava \mathcal{C}^\top , ki jo iz \mathcal{C} dobimo po zamenjavi ključnega literala in glave (primer 8.10). Glavo obratne klasifikacijske izjave \mathcal{C}^\top pri tem označimo z *relax*.

Primer 8.10. *Vzemimo klasifikacijsko izjavo*

$$\text{dialog}(O_1, O_2, X) : - \text{bivališče}(\text{Naslov}, O_2) \ \& \ \text{pošta}(O_1, \text{Naslov}, X)^*.$$

iz primera 8.8. Njej pripadajoča obratna klasifikacijska izjava je tedaj

$$\text{relax}(\text{pošta}(O_1, \text{Naslov}, X)) : - \text{bivališče}(\text{Naslov}, O_2) \ \& \ \text{dialog}(O_1, O_2, X).$$

Relaksacija z obratnimi klasifikacijskimi izjavami poizvedbe Q , dobljene v postopku zamenjave spremenljivk, poteka podobno kot relaksacija s klasifikacijskimi izjavami. Za vsak literal iz Q poiščemo obratne klasifikacijske izjave, v katerih ta nastopa v glavi, nato pa izbran literal zamenjamo s telesom pripadajoče obratne klasifikacijske izjave (primer 8.11).

Primer 8.11. *Vzemimo poizvedbo*

$$\text{polet}(\text{brnik}, \text{orly})?$$

ki sprašuje po letalskih povezavah letališča Brnik z letališčem Orly v Parizu. V Parizu poleg letališča Orly obstaja še letališče Charles De Gaulle. S postopkom zamenjave spremenljivk dobimo poizvedbo

$$\text{polet}(X, Y) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y?$$

Naj podatkovna zbirka vsebuje naslednje klasifikacijske izjave

$$\begin{aligned} \mathcal{C}_1 : \text{pariz_let}(X) & : - \text{orly enak } X^*. \\ \mathcal{C}_2 : \text{pariz_let}(X) & : - \text{de_gaulle enak } X^*. \\ \mathcal{D}_1 : \text{pot}(A, B) & : - \text{letališče}(X, A) \ \& \ \text{letališče}(Y, B) \ \& \ \text{polet}(X, Y)^*. \\ \mathcal{D}_2 : \text{pot}(A, B) & : - \text{žpostaja}(X, A) \ \& \ \text{žpostaja}(Y, B) \ \& \ \text{vlak}(X, Y)^*. \\ \mathcal{D}_3 : \text{pot}(A, B) & : - \text{apostaja}(X, A) \ \& \ \text{apostaja}(Y, B) \ \& \ \text{avtobus}(X, Y)^*. \end{aligned}$$

in njim pripadajoče obratne klasifikacijske izjave

$$\begin{aligned} \mathcal{C}_1^\top : \text{relax}(\text{orly enak } X) & : - \text{pariz_let}(X). \\ \mathcal{C}_2^\top : \text{relax}(\text{de_gaulle enak } X) & : - \text{pariz_let}(X). \\ \mathcal{D}_1^\top : \text{relax}(\text{polet}(X, Y)) & : - \text{letališče}(X, A) \ \& \ \text{letališče}(Y, B) \ \& \ \text{pot}(A, B). \\ \mathcal{D}_2^\top : \text{relax}(\text{vlak}(X, Y)) & : - \text{žpostaja}(X, A) \ \& \ \text{žpostaja}(Y, B) \ \& \ \text{pot}(A, B). \\ \mathcal{D}_3^\top : \text{relax}(\text{avtobus}(X, Y)) & : - \text{apostaja}(X, A) \ \& \ \text{apostaja}(Y, B) \ \& \ \text{pot}(A, B). \end{aligned}$$

Te ponazarjajo ISA relacijo nad konstantami, ki označujejo letališča v Parizu, in predikati, ki navajajo možne načine potovanja med dvema krajema.

Če podatkovna zbirka ne vsebuje odgovora na začetno poizvedbo, tj. poizvedba je neuspešna, lahko uporabimo postopek relaksacije. Najprej pokažimo primer transformacije predikata polet. Z uporabo obratne klasifikacijske izjave \mathcal{D}_1^\top dobimo poizvedbo

$$\text{pot}(A, B) \ \& \ \text{letališče}(X, A) \ \& \ \text{letališče}(Y, B) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y?$$

Šele ponovna zamenjava spremenljivk dobljene poizvedbe omogoča poljuben nadaljnji korak v postopku relaksacije:

$$\text{pot}(A, B) \ \& \ \text{letališče}(X, A_1) \ \& \ \text{letališče}(Y, B_1) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y \ \& \ A \ \text{enak } A_1 \ \& \ B \ \text{enak } B_1?$$

Zgornjo poizvedbo z uporabo klasifikacijske izjave \mathcal{D}_2 preoblikujemo, tako da prvi literal $\text{pot}(A, B)$ nadomestimo s telesom klasifikacijske izjave \mathcal{D}_2 . Dobimo poizvedbo, katere osrednji literal vlak je bolj specifičen od izmenjanega literala pot:

$$\text{vlak}(X_1, Y_1) \ \& \ \text{žpostaja}(X_1, A) \ \& \ \text{žpostaja}(Y_1, B) \ \& \ \text{letališče}(X, A_1) \ \& \ \text{letališče}(Y, B_1) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y \ \& \ A \ \text{enak } A_1 \ \& \ B \ \text{enak } B_1?$$

Ker je $A = \text{ljubljana}$ in $B = \text{pariz}$, je ta poizvedba logično ekvivalentna poizvedbi

$$\text{vlak}(X_1, Y_1) \ \& \ \text{žpostaja}(X_1, \text{ljubljana}) \ \& \ \text{žpostaja}(Y_1, \text{pariz})?$$

ki sprašuje po železniških povezavah med Ljubljano in Parizom. Pri tem X_1 označuje železniške postaje v Ljubljani, X_2 pa železniške postaje v Parizu.

Pokažimo še primer transformacije pogoja začetne poizvedbe. Če začetno poizvedbo relaksiramo z obratno klasifikacijsko izjavo \mathcal{C}_1^\top , dobimo poizvedbo

$$\text{polet}(X, Y) \ \& \ \text{brnik enak } X \ \& \ \text{pariz_let}(Y)?$$

ki sprašuje po letalskih povezavah med Brnikom in poljubnim letališčem v Parizu. Posplošitev se nanaša na izbiro letališča.

Tretja oblika relaksacije poizvedbe temelji na transformaciji množice parametrov. Povedali smo že, da je poljuben predikat p v relaciji ISA_P s predikatom, ki ga dobimo z razširitvijo množice atributov predikata p . Za dodatni atribut *Odhod*, ki označuje čas odhoda letala, torej velja klasifikacijska izjava

$$\text{polet}(X, Y, \text{Odhod}) \ : \ - \ \text{polet}(X, Y)^*.$$

in zato tudi obratna klasifikacijska izjava

$$\text{relax}(\text{polet}(X, Y)) : - \text{polet}(X, Y, \text{Odhod}).$$

Začetno poizvedbo je zato mogoče relaksirati v obliko

$$\text{polet}(X, Y, \text{Odhod}) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y?$$

ki sprašuje po časih odhodov letal z Brnika na pariško letališče Orly.

Nespretna transformacija pogoja poizvedbe pogosto zahteva številčno množico dodatnih enomestnih predikatov, kot je na primer *pariz_let* iz primera 8.11. Ker se želimo dodatnim predikatom izogniti, definiramo posebno vrsto transformacije, imenovano *odsev*. Odsev je zaporedje relaksacije z obratno klasifikacijsko izjavo \mathcal{C}^\top in preoblikovanja s pripadajočo klasifikacijsko izjavo \mathcal{C} , pri čemer izbran literal iz poizvedbe zamenjamo s telesom klasifikacijske izjave \mathcal{C} . Primer odseva zajema tudi primer 8.11.

Na prvi pogled se odsev zdi nesmiselna operacija, ki pridela odgovor, že dobljen z začetno, nerelaksirano poizvedbo, kar pa ni popolnoma res. Če namreč poizvedbo transformiramo z odsevom, poleg že znanih podatkov pogosto pridelaymo tudi nove, še neznane podatke, kar ponazarja primer 8.12. Pri tem ne potrebujemo dodatnih enomestnih predikatov.

Primer 8.12. *Vzemimo poizvedbo*

$$\text{polet}(\text{brnik}, \text{orly})?$$

oziroma

$$\text{polet}(X, Y) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y?$$

ter klasifikacijsko izjavo \mathcal{D}_1 in obratno klasifikacijsko izjavo \mathcal{D}_1^\top iz primera 8.11. Po relaksaciji poizvedbe z \mathcal{D}_1^\top dobimo

$$\text{pot}(A, B) \ \& \ \text{letališče}(X, A) \ \& \ \text{letališče}(Y, B) \ \& \ \text{brnik enak } X \ \& \ \text{orly enak } Y?$$

Ko (po zamenjavi spremenljivk) to poizvedbo preoblikujemo še z \mathcal{D}_1 , dobimo (strnjeno zapisano) poizvedbo

$$\text{polet}(X, Y) \ \& \ \text{letališče}(X, A) \ \& \ \text{letališče}(Y, B) \ \& \ \text{letališče}(\text{brnik}, A) \ \& \ \text{letališče}(\text{orly}, B)?$$

Ko bo $X = \text{brnik}$ in $Y = \text{orly}$, bo odgovor enak odgovoru na začetno poizvedbo $\text{polet}(\text{brnik}, \text{orly})$? Ko bo $Y \neq \text{orly}$, dobimo nove odgovore. Ker je $A = \text{ljubljana}$ in $B = \text{pariz}$, je atomarna formula $\text{letališče}(X, A)$ resnična pri $X = \text{brnik}$, atomarna formula $\text{letališče}(Y, B)$ pa, ko Y označuje katero koli letališče v Parizu, torej Orly ($Y = \text{orly}$) ali Charles de Gaulle ($Y = \text{de_gaulle}$). Sledi, tudi odgovor na to poizvedbo,

ki ne vsebuje dodatnega predikata *pariz_let*, vsebuje vse letalske povezave med Brnikom in poljubnim letališčem v Parizu.

Namen relaksacije je torej poizvedbo preoblikovati, tako da bo odgovor poleg odgovora na začetno poizvedbo vseboval še dodatne, sosedne podatke. To je podatke, ki začetni poizvedbi sicer ne bodo ustrezali, bodo pa z njo v nekem smiselnem, relevantnem odnosu. Poglejmo, kdaj bo odgovor na začetno poizvedbo podmnožica odgovora na relaksirano poizvedbo.

Prvo, kar opazimo je, da se množici spremenljivk začetne poizvedbe Q in njej prirejene relaksirane poizvedbe Q' lahko razlikujeta, kar je posledica v postopku relaksacije dovoljenega dodajanja in odvzemanja spremenljivk. To sicer ne bo vedno držalo – včasih se bosta množici spremenljivk poizvedb Q in Q' tudi ujemali. Kakor koli že, nas zanimajo odgovori, ki bodo vsebovali tudi odgovor na začetno, netransformirano poizvedbo, in zato oblika relaksacije, ki množico spremenljivk poizvedbe pri transformaciji poveča. Tako obliko relaksacije imenujemo *generalizacija*.

Poglejmo torej, kateri pogoj mora relaksacija poizvedbe Q v Q' izpolnjevati, da bo generalizacija. Ob dani klasifikacijski izjavi izvedemo korak relaksacije na osrednjem literalu poizvedbe tako, da ga nadomestimo z glavo ustrezne klasifikacijske izjave ter poizvedbi dodamo vse neključne literale. Poizvedba Q' bo generalizacija poizvedbe Q , če bodo neključni literali resnični pri vseh določenih spremenljivk, pri katerih bo resničen ključni literal. Naj bo torej

$$\mathcal{C} : \quad h : - a_1 \ \& \ \dots \ \& \ a_n \ \& \ k^*.$$

poljubna klasifikacijska izjava, kjer je h glava, k ključni literal, a_1, \dots, a_n pa neključni literali. Naj bodo X_1, \dots, X_m spremenljivke literala k , Y_1, \dots, Y_p pa tiste spremenljivke literalov a_1, \dots, a_n , ki ne nastopajo v k . Rekli bomo, da je obratna klasifikacijska izjava

$$\mathcal{C}^\top : \quad \text{relax}(k) : - a_1 \ \& \ \dots \ \& \ a_n \ \& \ h.$$

ohranljiva, če k ne vsebuje konstant in velja

$$\forall X_1, \dots, X_m \ \exists Y_1, \dots, Y_p : (k \Rightarrow a_1 \wedge \dots \wedge a_n).$$

Glej primer 8.13. Pogoj ohranljivosti obratne klasifikacijske izjave lahko delno izpolnimo z integritetnimi omejitvami – z integritetnimi omejitvami lahko zagotovimo obstoj spremenljivk Y_1, \dots, Y_p , ne pa tudi njihove medsebojne povezanosti in/ali pripadnosti spremenljivkam $\forall X_1, \dots, X_m$.

Primer 8.13. *Obratna klasifikacijska izjava \mathcal{D}_1^\top iz primera 8.11 je ohranljiva le, če velja*

$$\forall X, Y \ \exists A, B : (\text{polet}(X, Y) \Rightarrow \text{letališče}(X, A) \wedge \text{letališče}(Y, B)).$$

To je, \mathcal{D}_1^\top je ohranljiva, če za vsako letališče vemo, kateremu mestu pripada.

Naj bo Q' poizvedba, rezultat relaksacije poizvedbe Q z ohranljivimi obratnimi klasifikacijskimi izjavami. Tedaj je Q' generalizacija poizvedbe Q [Gaasterland-1992b]. To pa pomeni, da odgovor na relaksirano poizvedbo gotovo zajema odgovor na začetno poizvedbo, če so vse obratne klasifikacijske izjave, uporabljene v postopku relaksacije, ohranljive.

Ker z relaksacijo raziskujemo soseščino začetne poizvedbe, so najzanimivejše tiste relaksirane poizvedbe, ki so dobljene v najmanj korakih. Bolj se od začetne poizvedbe oddaljimo, manj verjetno je, da bo odgovor uporabnika zanimal. Za zelo uporabno se zato izkaže strategija [Gaasterland-1992b], ki najprej izbere le relaksacije, dobljene v enem koraku, nato pa pusti izbiro uporabniku. Ta se odloči oz. pokaže smer, v katero naj se interakcija nadaljuje. Pri tem moramo biti pozorni tudi na obvladljivost števila relaksacij, ki jih uporabniku ponudimo.

8.4 Tehnike sodelujočega odgovarjanja, ki jih podpirajo Datalog in dialog mIPC Kripkejevi modeli

Podatkovna zbirka deduktivnega podatkovnega sistema, ki temelji na podatkovnem jeziku Datalog (poglavje 6), je sestavljena iz ekstenzionalne podatkovne zbirke (EPZ), intenzionalne podatkovne zbirke (IPZ) in integritetnih omejitev (IO). Datalog zato podpira naslednje tehnike sodelujočega odgovarjanja:

- ↪ vrednotenje domnev, vsebovanih v poizvedbah,
- ↪ odkrivanje in odpravljanje nesporazumov ter
- ↪ oblikovanje intenzionalnih odgovorov.

Povedali smo že, da lahko urejenost predikatov in konstant izrazimo s Hornovimi stavki. Posledično lahko s Hornovimi stavki izrazimo tudi klasifikacijske izjave in obratne klasifikacijske izjave, težavo povzroča le oznaka * ključnega literala klasifikacijske izjave oz. oznaka relax glave obratne klasifikacijske izjave, brez katerih ne moremo vedeti, da gre za ta tipa izjav. Datalog torej, v svoji osnovni obliki, ne podpira relaksacije oz. tehnike posploševanja poizvedb in odgovorov.

Dialog mIPC Kripkejevi modeli, na katerih temelji podatkovni model sodelujoče podatkovne zbirke, uporabljene v drugem eksperimentu Čarovnik iz Oza, uporabljajo intuicionistično modalno logiko, ki je veliko zmogljivejša od logike prvega reda. V razdelku 2.5 smo videli, da ta podatkovni model (poleg vseh tehnik sodelujočega odgovarjanja, ki jih podpira Datalog) podpira tudi relaksacijo. Pri tem imajo ključno vlogo modalni operatorji. Večja zmogljivost intuicionistične modalne logike žal vodi tudi do večje kompleksnosti podatkovnih modelov. Podatkovni modeli, ki so ali veliko kompleksnejši od relacijske algebre ali popolnoma drugače zasnovani, pa s priljubljeno relacijsko algebro le težko tekmujejo.

Relacijska algebra in ostali modeli relacijskih podatkovnih zbirk ne podpirajo nobene izmed tehnik sodelujočega odgovarjanja. To pa pomeni, da jih, v njihovi osnovni

obliki, ne moremo šteti za sodelujoče podatkovne modele. V nadaljevanju se bomo zato posvetili različnim razširitvam relacijskega podatkovnega modela, tudi razširitvam, ki podpirajo nekatere tehnike sodelujočega odgovarjanja.

8.5 Razširitve relacijskega podatkovnega modela

Omenili smo že (poglavje 5), da imajo relacijski jeziki vrsto pomanjkljivosti. Te se v glavnem nanašajo na tranzitivno zaprtje, komplement in kompleksnost podatkov [Elmasri-2004]. Relacijski podatkovni modeli so se npr. izkazali za premalo zmogljive pri urejanju in shranjevanju satelitskih slik, genskih informacij, arhitekturnih načrtov, časovnih podatkov o borznih tečajih in prodajah delnic, prostorskih in drugih geografskih podatkov, podatkov o onesnaženju zraka in vode ter prometnih podatkov. Relacijski podatkovni modeli pa ne podpirajo tudi nobene izmed naštetih tehnik sodelujočega odgovarjanja in jih torej ne moremo šteti za sodelujoče podatkovne modele.

Ravno zaradi izjemne priljubljenosti relacijskih podatkovnih zbirk pa se, hkrati z odkrivanjem pomanjkljivosti, pojavlja tudi vedno več razširitev relacijskega podatkovnega modela. Tako med drugim obstajajo razširitve, ki v relacijske podatkovne zbirke vpelejo:

- ↪ gnezdene relacije [Schek-1986, Thomas-1986, Paredaens-1992, Elmasri-2004],
- ↪ urejenost relacij [Buneman-1991, Raymond-1996, Ng-2001, Hafenrichter-2005],
- ↪ kompleksne tipe podatkov, kot so zaporedja, tabele, množice, vreče [Guting-1989, Kim-1995, Elmasri-2004].
- ↪ nekatere vrste sodelujočega odgovarjanja [Motro-1988, Motro-1990, Godfrey-1994, Chu-1994, Chu-1996, Hafenrichter-2005],

V nadaljevanju bomo predstavili nekatere izmed predlaganih razširitev relacijskega podatkovnega modela.

8.5.1 Gnezdeni relacijski podatkovni modeli

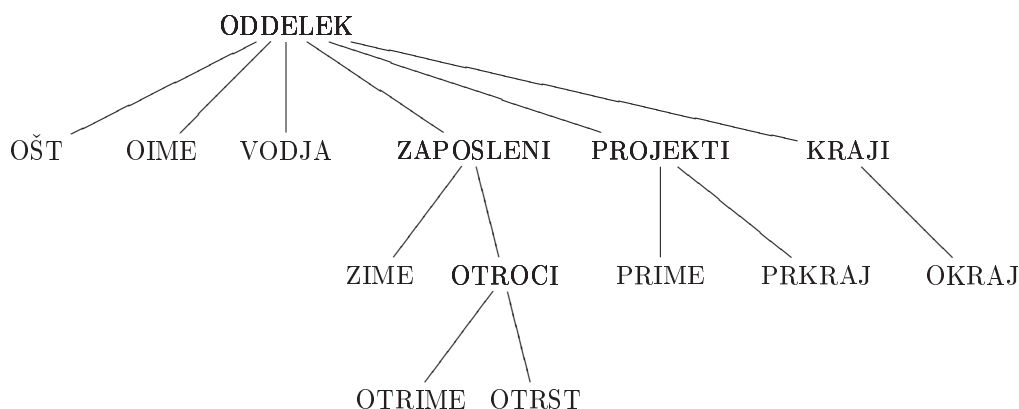
Relacijski podatkovni model, ki dovoljuje gnezdenje relacij, imenujemo *gnezden relacijski podatkovni model* [Schek-1986]. Ta model, v nasprotju z osnovnim relacijskim podatkovnim modelom [Codd-1970], predvideva sestavljene in večvrednostne attribute, ki vodijo do kompleksnih vrstic s hierarhično strukturo. Gnezden relacijski podatkovni model je zato primeren za predstavitev objektov, ki so že po naravi hierarhično strukturirani.

Relacijska shema gnezdene relacijske podatkovne zbirke je sestavljena iz vrste *zunanjih relacijskih shem* – vsaki relaciji najvišje ravni gnezdenja pripada natanko ena zunanja relacijska shema. Attribute zunanje relacijske sheme, ki ustrezajo kateri od v njej

ugnezdenih relacijskih struktur, imenujemo *notranje relacijske sheme*. Attribute najnižje ravni relacijske hierarhične strukture imenujemo *enostavni atributi*. Relacijsko shemo najnazorneje predstavimo z drevesno strukturo, katere koren predstavlja dana zunanja relacijska shema, listi so enostavni atributi, notranja vozlišča pa ustrezajo notranjim relacijskim shemam. Primer relacijske sheme gnezdene relacijske podatkovne zbirke prikazuje slika 8.1. Pripadajoča drevesna predstavitev je podana s sliko 8.2.

OŠT	OIME	VODJA	ZAPOSLENI		PROJEKTI		KRAJI	
			ZIME	OTROCI		PRIME	PRKRAJ	OKRAJ
				OTRIME	OTRST			

Slika 8.1: Primer relacijske sheme relacije ODDELEK gnezdene relacijske podatkovne zbirke.



Slika 8.2: Drevesna predstavitev relacijske sheme relacije ODDELEK.

Edina zunanja relacijska shema na sliki 8.2 je shema relacije ODDELEK. Vse ostale relacijske sheme, tj. sheme relacij ZAPOSLENI, PROJEKTI, KRAJI in OTROCI, so notranje relacijske sheme. Enostavni atributi so OŠT, OIME, VODJA, ZIME, OTRIME, OTRST, PRIME, PRKRAJ in OKRAJ.

Razširitev relacijskih podatkovnih zbirk z gnezdenimi relacijami zahteva razširitev relacijskega podatkovnega modela [Thomas-1986]. Dobljena *gnezdena relacijska algebra* vsebuje dve dodatni operaciji, in sicer operacijo gnezdenja in gnezdenju obratno operacijo. Izkaže pa se, da je osnovna relacijska algebra po izrazljivosti enako močna kot gnezdena relacijska algebra, tj. iz osnovnih relacijskih podatkovnih zbirk je z relacijsko algebro mogoče pridobiti enake podatke kot z gnezdeno relacijsko algebro [Paredaens-1992]. Sledi, tudi z gnezdeno relacijsko algebro ni mogoče izraziti tranzitivnega zaprtja. Zaradi omenjenega so številni raziskovalci predlagali gradnjo podatkovnih sistemov, ki podpirajo gnezdeno relacijski podatkovni model, njihov uporabniški vmesnik pa temelji na osnovni relacijski algebri.

8.5.2 Urejeni relacijski podatkovni modeli

Urejenost je kot osnovna lastnost skoraj vseh podatkovnih tipov naravno prisotna v številnih podatkovnih zbirkah [Maier-1993, Raymond-1996]. Relacijske podatkovne zbirke pa podpirajo le naslednje standardne urejenosti podatkovnih tipov:

- abecedna urejenost črkovnih nizov,
- numerična urejenost števil,
- kronološka urejenost datumov.

Ker omejena podpora urejenosti v relacijskih podatkovnih zbirkah vodi do izgube pomembne semantične informacije, številni raziskovalci poskušajo relacijski podatkovni model ustrezno razširiti [Buneman-1991, Raymond-1996, Ng-2001, Hafenrichter-2005]. Oglejmo si le razširitev, imenovano *delno urejen relacijski podatkovni model*, ki jo je predlagal Ng [Ng-2001].

Naj bo vrednostna množica D_i atributa A_i delno urejena z relacijo \sqsubseteq_{D_i} , tj. (D_i, \sqsubseteq_{D_i}) je *delno urejena domena* atributa A_i za $1 \leq i \leq n$. Relacijo R z relacijsko shemo $R(A_1, \dots, A_n)$ tedaj imenujemo *delno urejena relacija*. Končno množico delno urejenih relacij imenujemo *delno urejena relacijska podatkovna zbirka*. Osnovna relacijska podatkovna zbirka, kjer so vse vrednostne množice atributov popolnoma neurejene, kar je ekvivalentno delni urejenosti z relacijo $=$, je poseben primer delno urejenih relacijskih podatkovnih zbirk. V delno urejenih relacijskih podatkovnih zbirkah se ohranita dve pomembni lastnosti običajnih (neurejenih) relacijskih podatkovnih zbirk: vsi elementi vseh vrednostnih množic so enostavni in vrstice so paroma različne. Pomembno je biti pozoren tudi na nekatere ključne razlike med omenjenima vrstama relacijskih podatkovnih zbirk:

1. Elementi delno urejene relacije z relacijsko shemo $R(A_1, \dots, A_n)$, kjer je (D_i, \sqsubseteq_{D_i}) delno urejena domena atributa A_i ($1 \leq i \leq n$), so delno urejeni z naslednjo relacijo:

$$(x_1, \dots, x_n) \sqsubseteq (y_1, \dots, y_n) \iff x_i \sqsubseteq_{D_i} y_i \text{ za } 1 \leq i \leq n$$

Elementi običajne relacije niso urejeni.

2. Atributi relacijske sheme delno urejene relacije so linearno urejeni.

Delno urejena razširitev relacijskih podatkovnih zbirk zahteva tudi razširitev relacijskega podatkovnega modela. Dobljena *delno urejena relacijska algebra* (PORA) vsebuje dodatni *predikat urejenosti* \sqsubseteq , ki lahko nastopa v formuli F osnovne operacije izbire σ_F . Pri tem je

$$\sigma_{A \sqsubseteq B}(R)$$

množica vseh elementov relacije R , ki zadoščajo formuli $A \sqsubseteq B$. Delno urejena relacijska algebra dovoljuje v formulah, ki določajo izbiro, aritmetične operatorje primerjav $=, <, >, \neq, \leq$ in \geq , ter predikata urejenosti \sqsubseteq in $\not\sqsubseteq$. Da bi preprečila neskladnost

v urejenosti unije dveh urejenih relacij, delno urejena relacijska algebra dovoljuje le unijo dveh relacij, ki imata isto relacijsko shemo in enako urejene vrednostne množice pripadajočih atributov.

PORA je po izrazljivosti močnejša od osnovne relacijske algebre. Obstajajo pa še nekatera odprta vprašanja:

- Kako integrirati delne urejenosti v podatkovne sisteme? Avtor [Ng-2001] predlaga integracijo v jedro podatkovnega sistema v obliki urejenega B-drevesa [Lynn-1982].
- Kako spreminjati delne urejenosti vrednostnih množic?

Zavedati se moramo, da je urejenost vrednostnih množic lahko odvisna od različnih dejavnikov, npr. časa, kraja in uporabnika. To je tudi razlog, zakaj mora biti urejenost integrirana tako, da je do nje mogoče brez težav dostopati in jo po potrebi spreminjati oz. prilagajati uporabnikom.

8.5.3 Objektno-relacijski podatkovni modeli

Razširitve relacijskega podatkovnega modela na nekatere kompleksne tipe podatkov imenujemo *objektno-relacijski podatkovni modeli*. V objektno-relacijskih podatkovnih zbirkah so podatki še vedno urejeni v obliki tabel, le vrednostne množice posameznih atributov (stolpcev) imajo bogatejšo podatkovno strukturo. Vsebujejo lahko npr. zaporedja, tabele, množice, vreče ali slike. Podatkovne tipe, ki jih dobimo kot kombinacijo osnovnih abecednih in številskih podatkovnih tipov, imenujemo *abstraktni podatkovni tipi*. Zanimiv primer abstraktnega podatkovnega tipa je *časovni niz*, tj. tabela časovno urejenih vrednosti.

Kim [Kim-1995] je objektno-relacijske podatkovne zbirke definiriral kot nadmnožico relacijskih podatkovnih zbirk, ki je v primeru neuporabe objektov (tj. abstraktnih podatkovnih tipov) enaka relacijskim podatkovnim zbirkam. Njihovi največji novosti in zanimivosti sta sposobnost definicije zelenih podatkovnih tipov in uporabniških funkcij za izvajanje operacij nad podatki. Različne objektne razširitve relacijskih podatkovnih zbirk zahtevajo različne razširitve relacijskega podatkovnega modela. Dobljeni *objektno-relacijski podatkovni modeli* vsebujejo operacije in funkcije za upravljanje s kompleksnimi tipi podatkov, kot so npr. časovni nizi.

Objektno-relacijske podatkovne zbirke so dosegle največji uspeh pri shranjevanju in urejanju zvočnih, slikovnih in časovno odvisnih podatkov [Grimes-1998].

8.5.4 Sodelujoči relacijski podatkovni modeli

Razširitve relacijskega podatkovnega modela, ki se nanašajo na različne tehnike sodelujočega odgovarjanja, imenujemo *sodelujoči relacijski podatkovni modeli*. Chu in Chen [Chu-1994] z uporabo *hierarhijske abstrakcije tipov* (TAH), ki jo opisujeta kot učinkovito

in urejeno ogrodje za predstavitev povezav med podatki in znanjem, dosežeta npr. relaksacijo.

Hierarhija abstrakcije tipov zajema pojem vsebovanja in pojem sestave, imenovana IS-A in PART-OF. Pri tem avtorja [Chu-1994] ločita dve vrsti *tipov* – atomarne tipe T , ki smo jih v okviru relacijskih podatkovnih zbirk poimenovali atributi, in sestavljene tipe (T_1, \dots, T_n) , ki so kartezični produkti več atributov. Pravimo, da je tip T' *abstrakcija* tipa T , kar pišemo kot $T \preceq T'$, če velja naslednje:

- Poljubni vrednosti atomarnega tipa T' pripada vsaj ena vrednost atomarnega tipa T .
- Za sestavljena tipa $T = (T_1, \dots, T_n)$ in $T' = (T'_1, \dots, T'_n)$ ter $1 \leq i \leq n$ velja $T_i \preceq T'_i$.

Primer abstrakcije numeričnega tipa je interval vrednosti, primer abstrakcije nenumeričnega tipa Ljubljana pa Slovenija. Tip Slovenija lahko npr. predstavlja množico mest: Ljubljana, Maribor, Celje ...

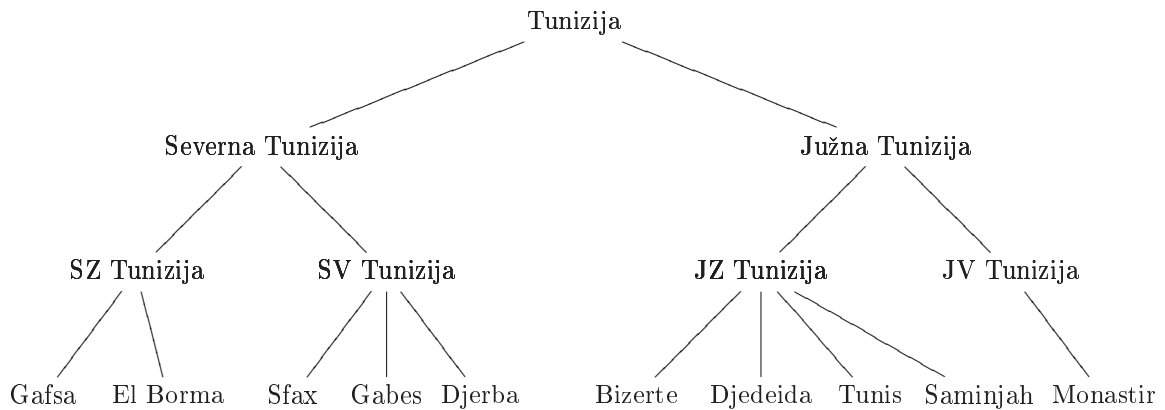
TAH je tedaj drevesna struktura, zgrajena iz množice *osnovnih tipov*. Listi danega TAH drevesa so vrednosti osnovnih tipov, vsa ostala vozlišča pa vrednosti abstrakcij osnovnih tipov. Če osnovni tip označimo s T , njegove abstrakcije pa s $T', T'', \dots, T^{(k)}$, je $T^{(k)}$ najsplošnejši tip in velja:

$$T \preceq T' \preceq T'' \preceq \dots \preceq T^{(k)}$$

Listi TAH drevesa tvorijo najnižjo raven, vrednosti tipa T^i ($1 \leq i \leq k$) pa so natanko i ravni nad njimi. Če gradimo TAH drevo atomarnega tipa, je tudi ta osnovni tip. V nasprotnem primeru, ko gradimo TAH drevo tipa, sestavljenega iz več atributov ene ali več relacij, je osnovni tip sestavljen natanko iz teh atributov. Premik po TAH drevesu navzgor pomeni preslikavo vrednosti tipa $T^{(i)}$ ($1 \leq i \leq k-1$) v vrednost tipa $T^{(i+1)}$, ki je splošnejša. Premik po TAH drevesu navzdol pomeni preslikavo vrednosti splošnejšega tipa $T^{(i+1)}$ ($1 \leq i \leq k-1$) v množico bolj specifičnih vrednosti tipa $T^{(i)}$. Primer TAH drevesa prikazuje slika 8.3.

Relaksacija izbrane osnovne vrednosti x je tedaj zaporedje preslikave te vrednosti v splošnejšo vrednost x' in (rekurzivno definirane) preslikave tako dobljene vrednosti x' nazaj v pripadajočo množico vrednosti osnovnega tipa. Za primer letališča Gabes s slike 8.3 to najprej pomeni preslikavo v splošnejšo vrednost SV Tunizija in nato preslikavo v množico {Sfax, Gabes, Djerba}.

Razširitev relacijskih podatkovnih zbirk s hierarhijo abstrakcije tipov zahteva tudi razširitev relacijskega podatkovnega modela. *Sodelujoča relacijska algebra*, na kateri temelji sodelujoč podatkovni sistem CoBase [Chu-1996], vsebuje številne dodatne *sodelujoče operacije*, ki so jih avtorji razdelili v štiri kategorije:



Slika 8.3: Primer TAH drevesa letališč v Tuniziji.

1. KONTEKSTNO PROSTE OPERACIJE

- *Approximate* v priredi vrednosti v interval vrednosti, ki ga uporabnik definira vnaprej. Na primer *Approximate* $9 = (8, 10)$.
- *Between* (v_1, v_2) priredi nastopajočima vrednostma v_1 in v_2 interval vrednosti, ki ga bodisi uporabnik bodisi upravitelj podatkovne zbirke definira vnaprej. Na primer *Between* $(7, \text{Approximate } 9) = (7, 10)$.
- *Within* $(x_1, y_1, \dots, x_n, y_n)$ določa področje dane slike, omejeno s poligonom $x_1y_1 \dots x_ny_n$.

2. KONTEKSTNO ODVISNE OPERACIJE

- *Near-to* x opisuje (prostorsko) okolico objekta x . Ta mera bližine je odvisna od konteksta in uporabnika. Definiramo jo lahko s TAH drevesi ali pa njeno definicijo prepustimo uporabniku.
- *Similar-to* x based-on $((A_1 w_1)(A_2 w_2) \dots (A_n w_n))$ opisuje množico objektov, ki so pomensko blizu ciljnega objekta x . Pri tem imajo ključno vlogo atributi A_1, \dots, A_n , ki jih izbere uporabnik, in uteži w_1, \dots, w_n , ki kažejo relativno pomembnost posameznih atributov. Množico podobnih objektov lahko uredimo glede na dobljeno mero podobnosti, tj. uteženo srednjo kvadratno napako. Velikost množice podobnih elementov omejimo z vnaprej določenim pragom.

3. KONTROLNE OPERACIJE

- *Relaxation-order* (A_1, \dots, A_n) določa vrstni red (A_1, \dots, A_n) atributov v postopku relaksacije. Če ne obstaja eksakten odgovor na poizvedbo, začnemo z relaksacijo po prvem atributu A_1 . Če na dobljeno relaksirano poizvedbo še vedno ni odgovora, nadaljujemo z relaksacijo po drugem atributu A_2 itd. V primeru, ko uporabnik ne poda vrstnega reda relaksacije, poteka relaksacija v skladu z vnaprej določeno strategijo.

- *Not-relaxable* (A_1, \dots, A_n) določa attribute (A_1, \dots, A_n) , ki jih ne želimo relaksirati.
- *Preference-list* (v_1, \dots, v_n) določa prednostni seznam vrednosti (v_1, \dots, v_n) danega atributa. Pri tem ima v_i prednost pred v_{i+1} . Z uporabo te operacije dosežemo relaksacijo po danem atributu, ki sledi prednostnemu seznamu.
- *Unacceptable-list* (v_1, \dots, v_n) uporabniku dovoljuje, da sistemu prepove določene odgovore. To lahko dosežemo tako, da določene dele TAH drevesa ne pregledujemo. Taka operacija ne vpliva le na večje zadovoljstvo uporabnika, temveč skrajša tudi čas iskanja odgovora.
- *Alternative-TAH* (*TAH-ime*) uporabnikom dovoljuje uporabo izbranih TAH dreves. Tako zadovoljimo raznolike potrebe različnih uporabnikov. Kot primer vzemimo popotnika, ki je na počitnicah in ga zanimajo predvsem cenejše letalske povezave, ter popotnika, ki je na poslovnem potovanju in ga zanimajo časovno ugodnejše letalske povezave. Da bi zadovoljili raznoliki potrebi uporabnikov, ustvarimo več različnih TAH dreves, ki poudarjajo različne attribute.
- *Relaxation-level* (v) določa največjo dovoljeno vrednost relaksacije po danem atributu, tj. vse vrednosti iz intervala $[0, v]$.
- *Answer-set* (s) določa najmanjše število odgovorov, ki jih zahteva uporabnik. CoBase izvaja relaksacijo tako dolgo, dokler ne dobi dovolj, to je $\geq s$, bližnjih odgovorov.

4. INTERAKTIVNE OPERACIJE

- *Nearer* in *Further* omogočata interaktivno kontrolo nad *Near-to* operacijo. Medtem ko operacija *Nearer* zmanjša razdaljo za vnaprej določen odstotek, jo operacija *Further* poveča.

TAH drevesa, ki predstavljajo znanje o medsebojnih odnosih podatkov podatkovne zbirke, lahko shranimo v dodatnih relacijskih tabelah. Kljub temu je predstavljena sodelujoča razširitev relacijskih podatkovnih zbirk zelo kompleksna in poleg omenjenih sodelujočih operacij zajema tudi spremembe v načinu vrednotenja poizvedb (logično sklepanje, spreminjanje, hevristično iskanje).

9

Kategorni model relacijske algebre

9.1 Uvod v teorijo kategorij

9.2 Kategorna posplošitev relacijske algebre

9.3 Kategorija podobnosti

9.4 Zmogljivost kategornega modela relacijskih podatkovnih zbirk

Podamo kategorno posplošitev relacijske algebre – relacijam in nad njimi definiranim operacijam relacijske algebre priredimo ustrezne kategorne konstrukte. Ugotovimo, da za posplošitev relacijske algebre potrebujemo natančen začetni objekt, končni objekt, končne kategorne produkte in slike. Vsi razredi regularnih podobjektov morajo vsebovati relativne komplemente, poljubna dva regularna podobjekta pa morata imeti najmanjšo zgornjo in največjo spodnjo mejo.

Predstavimo kategorijo podobnosti, ki je ustrezno okolje za kategorno predstavitev relacijske algebre. Izpostavimo dve zanimivi podkategoriji kategorije podobnosti, imenovani kategorija urejenih množic in kategorija kompaktnih metričnih prostorov.

Opišemo implementacijo in zmogljivost dobljenega kategornega modela relacijskih podatkovnih zbirk.

9.1 Uvod v teorijo kategorij

Relacijske podatkovne zbirke bomo pogledali z drugega zornega kota, namreč z zornega kota *teorije kategorij*, s katero dosežemo manjšo kompleksnost modelov zmogljivih, tudi sodelujočih podatkovnih zbirk. Pojem kategorije sta leta 1945 uvedla S. Eilenberg in S. MacLane [Eilenberg-1945], osnova teorije kategorij pa je pregledno predstavljena v [Bucur-1968, Barr-1999, Borceux-1994a, Awodey-2005].

Kategorija \mathcal{C} je struktura, sestavljena iz:

1. razreda *objektov* \mathcal{C}_0 in
2. razreda *morfizmov* \mathcal{C}_1 ,

za katere velja naslednje:

- Vsak $f \in \mathcal{C}_1$ ima enolično določeni domeno $\text{dom} f$ in kodomeno $\text{cod} f$, ki sta objekta kategorije \mathcal{C} . Razred morfizmov z domeno A in kodomeno B označimo s $\text{Hom}(A, B)$. Za $f \in \text{Hom}(A, B)$ pišemo $f : A \rightarrow B$.
- Nad morfizmi obstaja binarna operacija, imenovana *kompozicija* ($A, B, C \in \mathcal{C}_0$):

$$\begin{array}{ccc} \circ : \text{Hom}(A, B) \times \text{Hom}(B, C) & \longrightarrow & \text{Hom}(A, C) \\ (f, g) & \longmapsto & g \circ f \end{array}$$

- Kompozicija je asociativna, tj. za poljubne $A, B, C, D \in \mathcal{C}_0$, $f \in \text{Hom}(A, B)$, $g \in \text{Hom}(B, C)$ in $h \in \text{Hom}(C, D)$ velja

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

- Obstaja enota za kompozicijo, tj. za vsak $A \in \mathcal{C}_0$ je $1_A : A \rightarrow A$ tak morfizem, da za $B, C \in \mathcal{C}_0$, $f : A \rightarrow B$ in $g : C \rightarrow A$ velja

$$f \circ 1_A = f \quad \text{in} \quad 1_A \circ g = g.$$

Podkategorija \mathcal{C}' kategorije \mathcal{C} je podstruktura z razredom objektov $\mathcal{C}'_0 \subseteq \mathcal{C}_0$ in razredom morfizmov $\mathcal{C}'_1 \subseteq \mathcal{C}_1$, katere kompozicija in enota za kompozicijo se ujemata s kompozicijo in enoto za kompozicijo v kategoriji \mathcal{C} .

Primer 9.1. Objekti kategorije množic **Set** so množice, morfizmi so preslikave množic, kompozicija je kompozicija preslikav, enota za kompozicijo pa identična preslikava množice. Relacije oz. množice, na katerih je zasnovana relacijska algebra, so torej objekti, operacije nad relacijami pa morfizmi kategorije množic. Trdimo lahko, da je relacijska algebra kot struktura, sestavljena iz relacij in nad njimi definiranih operacij, podstruktura kategorije množic.

Primer 9.2. Objekti kategorije grafov **Graph** so usmerjeni grafi, morfizmi so homomorfizmi grafov, kompozicija je kompozicija homomorfizmov, enota za kompozicijo pa identični homomorfizem grafa.

Primer 9.3. Objekti kategorije mrež [Borceux-1994b] so mreže, morfizmi so homomorfizmi mrež, kompozicija je kompozicija homomorfizmov mrež, enota za kompozicijo pa identični homomorfizem mrež. Pri tem povejmo, da je mreža $L_M = (M; \wedge, \vee)$ neprazna množica M z binarnima operacijama \wedge (kapa) in \vee (kupa), ki imata naslednje lastnosti:

1. KOMUTATIVNOST:

$$\forall a, b \in M : a \wedge b = b \wedge a \text{ in } a \vee b = b \vee a$$

2. ASOCIATIVNOST:

$$\forall a, b, c \in M : (a \wedge b) \wedge c = a \wedge (b \wedge c) \text{ in } (a \vee b) \vee c = a \vee (b \vee c)$$

3. IDEMPOTENTNOST:

$$\forall a \in M : a \wedge a = a \text{ in } a \vee a = a$$

4. ABSORPCIJA:

$$\forall a, b \in M : (a \wedge b) \vee a = a \text{ in } (a \vee b) \wedge a = a$$

Podmreža mreže $L_M = (M; \wedge, \vee)$ je vsaka podmnožica množice M , zaprta za binarni operaciji \wedge in \vee . V mrežo vpeljemo relacijo

$$a \leq b \iff a \wedge b = a \iff a \vee b = b,$$

ki mrežo delno ureja (refleksivnost, antisimetričnost, tranzitivnost). Tudi vsaka delno urejena množica (M, \leq) , v kateri imata poljubna dva elementa enolično določeni največjo spodnjo (inf) in najmanjšo zgornjo mejo (sup), je za operaciji

$$a \wedge b = \inf\{a, b\}$$

$$a \vee b = \sup\{a, b\}$$

mreža [Vidav-2003], katere inducirana relacija \leq in relacija \preceq se ujemata.

Homomorfizem mrež $L_A = (A; \wedge_A, \vee_A)$ in $L_B = (B; \wedge_B, \vee_B)$ je preslikava $f : A \rightarrow B$, za katero za poljubna $a, b \in A$ velja:

$$f(a \wedge_A b) = f(a) \wedge_B f(b)$$

$$f(a \vee_A b) = f(a) \vee_B f(b)$$

Homomorfizmi mrež spoštujejo urejenost \leq :

$$x \leq y \implies f(x) \leq f(y)$$

V disertaciji nas bodo zanimala polne mreže [Grätzer-1998], tj. mreže, v katerih ima vsaka podmnožica največjo spodnjo in najmanjšo zgornjo mejo. Neprazna polna mreža

vsebuje največji element 1, tj. najmanjšo zgornjo mejo vseh elementov mreže, in najmanjši element 0, tj. največjo spodnjo mejo vseh elementov mreže. Podmreža $L_A = (A; \wedge, \vee)$ polne mreže $L_M = (M; \wedge, \vee)$ je vsaka podmnožica množice M , ki vsebuje najmanjši in največji element mreže L_M ter največjo spodnjo in najmanjšo zgornjo mejo vsake svoje podmnožice. Podmreže polnih mrež so polne.

Relativni komplement elementa x v polni mreži (ali delno urejeni množici z največjim elementom 1 in najmanjšim elementom 0) L_M je tak element y iz L_M , za katerega velja: $x \wedge y = 0$ in $x \vee y = 1$.

Homomorfizem polnih mrež $L_A = (A; \wedge_A, \vee_A)$ in $L_B = (B; \wedge_B, \vee_B)$ je preslikava množic $f : A \rightarrow B$, za katero za poljubno družino $\{a_i\}_{i \in I}$ elementov iz A velja:

$$\bigwedge_{i \in I} f(a_i) = f\left(\bigwedge_{i \in I} a_i\right)$$

$$\bigvee_{i \in I} f(a_i) = f\left(\bigvee_{i \in I} a_i\right)$$

Homomorfizmi polnih mrež ohranjajo elementa 0 in 1. Polne mreže in homomorfizmi polnih mrež tvorijo kategorijo polnih mrež **CLat**.

Tako kot v običajni kategoriji množic, kjer ločimo injektivne, surjektivne in bijektivne preslikave množic, v teoriji kategorij ločimo *monomorfizme*, *epimorfizme* in *izomorfizme*. Morfizem

$$f : B \rightarrow C$$

imenujemo *monomorfizem*, če za poljubna morfizma

$$g_1 : A \rightarrow B$$

$$g_2 : A \rightarrow B$$

iz $f \circ g_1 = f \circ g_2$ sledi $g_1 = g_2$. Tedaj pišemo $f : B \rightarrow C$. Nekateri monomorfizmi so še posebej zanimivi – predstavili bomo monomorfizme, imenovane *regularni monomorfizmi* ali *zožki*. Zožek $e : E \rightarrow B$ para morfizmov $f, g : B \rightarrow C$ je morfizem, za katerega velja

$$f \circ e = g \circ e$$

in za poljuben morfizem $e' : E' \rightarrow B$, ki izpolnjuje pogoj $f \circ e' = g \circ e'$, obstaja natanko en morfizem $k : E' \rightarrow E$, da velja $e \circ k = e'$. Glej sliko 9.1.

Izkaže se, da so vsi zožki monomorfizmi in da so določeni do izomorfizma natanko [Awodey-2005]. Zožke imenujemo tudi *regularni monomorfizmi*.

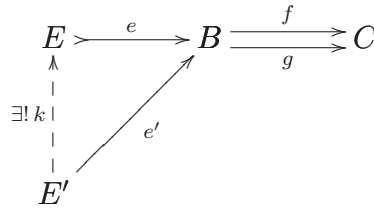
Morfizem

$$f : B \rightarrow C$$

imenujemo *epimorfizem*, če za poljubna morfizma

$$h_1 : C \rightarrow D$$

$$h_2 : C \rightarrow D$$



Slika 9.1: Zožek ali regularni monomorfizem.

iz $h_1 \circ f = h_2 \circ f$ sledi $h_1 = h_2$. Tedaj pišemo $f : B \rightarrow C$. Morfizem f imenujemo *izomorfizem*, če obstaja morfizem

$$f^{-1} : C \rightarrow B,$$

da velja

$$f^{-1} \circ f = 1_B \quad \text{in} \quad f \circ f^{-1} = 1_C.$$

Tedaj f^{-1} imenujemo *inverz* morfizma f in pišemo $f : B \rightsquigarrow C$. Morfizmi 1_A iz definicije kategorije so izomorfizmi. Imenujemo jih *identični izomorfizmi*. Če med objektoma B in C obstaja izomorfizem, pravimo, da sta *izomorfna*. Oznaka \rightsquigarrow je nekoliko zavarajoča. Res je, da so vsi izomorfizmi tudi monomorfizmi in epimorfizmi, obratno pa ni res [Awodey-2005]. Obstajajo namreč kategorije in morfizmi, ki so monomorfizmi in epimorfizmi, ne pa tudi izomorfizmi. Primer je kategorija z dvema objektoma in edinim morfizmom med tema objektoma.

Primer 9.4. *Monomorfizmom kategorije množic Set (primer 9.1) ustrezajo injektivne preslikave, epimorfizmom ustrezajo surjektivne preslikave, izomorfizmom pa bijektivne preslikave. Vsi monomorfizmi kategorije množic so regularni [Barr-1999].*

Primer 9.5. *Monomorfizmom kategorije grafov Graph (primer 9.2) ustrezajo monomorfizmi grafov, epimorfizmom ustrezajo epimorfizmi grafov, izomorfizmom pa izomorfizmi grafov [Diestel-2000]. Izkazuje se, da so tudi vsi monomorfizmi kategorije grafov regularni [Borceux-1994c].*

Primer 9.6. *V kategoriji polnih mrež CLat (primer 9.3) so monomorfizmi injektivni homomorfizmi polnih mrež, izomorfizmi pa homomorfizmi polnih mrež, ki so injektivni in surjektivni.*

Obstajata dva popolnoma različna načina uporabe teorije kategorij pri modeliranju podatkovnih zbirk:

- \rightsquigarrow MODULARIZACIJA: Notranjo strukturo osnovnih podatkovnih objektov, kot so npr. tabele, prezremo v korist zunanjih odnosov med njimi. Pri modularizaciji nas zanimajo podatki kot *objekti*, katerim priredimo *morfizme*, da dobimo kategorijo. Dobljena kategorija tedaj služi za abstrakcijo oz. posploševanje podatkovnih objektov in operacij, ki se nad njimi izvajajo.

↪ SINTETIČNO SKLEPANJE: Notranjo strukturo posplošenih osnovnih podatkovnih objektov poskušamo ponovno vzpostaviti. Namen sintetičnega sklepanja je kategorije približati inženirjem in uporabi v praksi, nanj pa lahko gledamo tudi kot na inverzno operacijo modularizacije. Modularizacija pa nikakor ni predpogoj za sintetično sklepanje – pogosto poznamo abstrakcijo notranje strukture podatkovne zbirke, ne pa tudi zunanjih odnosov med podatkovnimi objekti.

Eden od čarov teorije kategorij je gotovo njena sposobnost združevanja shematičnega formalizma in algebraičnega simbolizma.

Kategorno modeliranje podatkovnih zbirk je precej razpršeno in nepregledno znanstveno področje. Potencial, ki ga teorija kategorij nudi, je v zadnjih desetletjih vodil do številnih kategornih modelov različnih podatkovnih zbirk [Dampney-1992, Rossiter-1994, Nelson-1995, Islam-1994, Johnson-1996, Johnson-2002a, Johnson-2002b]. Mi bomo poiskali kategorni model relacijskih podatkovnih zbirk, ki bo temeljil na modularizaciji in bo zmogljivost relacijske algebre ustrezno razširil. Relacijsko algebro kot najbolj znan model relacijskih podatkovnih zbirk bomo posplošili na teorijo kategorij, tako da bomo relacijam in nad njimi definiranim operacijam priredili ustrezne kategorne konstrukte. Ta postopek nas bo vodil do kategorije in družine njenih podkategorij, ki bo združevala različne kategorne modele relacijske algebre. Eden izmed modelov bo tudi klasična relacijska algebra, na kateri bo posplošitev temeljila in ki je definirana v posebnem primeru kategorije, imenovanem kategorija množic **Set**.

9.2 Kategorna posplošitev relacijske algebre

Kategorije, ki bodo primerno okolje za modeliranje relacijskih podatkovnih zbirk, morajo vsebovati posebne objekte, imenovane *regularni podobjekti*, ki so posplošitev pojma podmnožice oz. podtabele iz kategorije množic **Set**.

Naj bo torej A poljuben objekt kategorije \mathcal{C} , $Mono(A)$ pa razred vseh monomorfizmov s kodomeno A . Naj bosta U in V objekta kategorije \mathcal{C} s pripadajočima monomorfizmoma $u, v \in Mono(A)$:

$$u : U \rightarrow A$$

$$v : V \rightarrow A$$

V razredu $Mono(A)$ definiramo relacijo \leq :

$$u \leq v \iff (\exists u' : U \rightarrow V, \text{ tako da je } u = v \circ u')$$

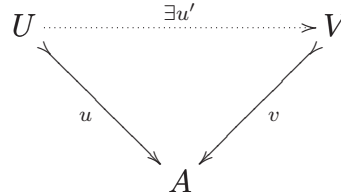
Glej sliko 9.2. Če je

$$u \leq v \text{ in } v \leq u,$$

pišemo $u \equiv v$, in \equiv je ekvivalenčna relacija (refleksivnost, simetričnost, tranzitivnost) v razredu $Mono(A)$. Razred *podobjektov* objekta A , ki ga označimo s $Sub(A)$, je razred ekvivalenčnih razredov relacije \equiv :

$$Sub(A) = Mono(A)/\equiv$$

Razred $\text{Sub}(A)$ podobjektov objekta A je z relacijo \leq delno urejen (refleksivnost, antisimetričnost, tranzitivnost).



Slika 9.2: Relacija \leq v razredu $\text{Mono}(A)$.

Regularnim monomorfizmom pripadajoče podobjekte imenujemo *regularni podobjekti*, razred regularnih podobjektov objekta A pa označimo z $\text{RegSub}(A)$.

Tudi za objekt U s pripadajočim monomorfizmom $u : U \rightarrow A$, ki je predstavnik nekega podobjekta objekta A , pravimo, da je *podobjekt* objekta A in pišemo $U \subseteq A$. Če za $v : V \rightarrow A$ velja $u \leq v$, pišemo $U \leq V$, in če sta u in v predstavnika istega podobjekta objekta A , pišemo $U = V$.

Primer 9.7. Objekti kategorije množic **Set** so množice (primer 9.1), monomorfizmi so injektivne preslikave (primer 9.4), podobjekti pa ekvivalenčni razredi injektivnih preslikav. Če naj bosta dve injektivni preslikavi

$$u : U \rightarrow A$$

$$v : V \rightarrow A$$

elementa istega podobjekta množice A , torej ekvivalentni glede na relacijo \equiv , morata biti množici U in V enako močni. Tedaj obstajata preslikavi

$$g : U \rightarrow V$$

$$h : V \rightarrow U$$

za kateri velja $u = v \circ g$ in $v = u \circ h$. Ker sta u in v injektivni preslikavi, sta g in h enolično določeni in injektivni. Zaradi

$$u = v \circ g = (u \circ h) \circ g = u \circ (h \circ g)$$

$$v = u \circ h = (v \circ g) \circ h = v \circ (g \circ h)$$

pa sledi še $1_U = h \circ g$ in $1_V = g \circ h$. Torej, g in h sta inverzna izomorfizma. Sledi, preslikavi u in v imata isto zalogo vrednosti $I \subseteq A$. Vložitev

$$\text{id}_I : I \hookrightarrow A$$

je element istega podobjekta kot u in v . Pokažimo, da je to edina vložitev v tem podobjektu množice A . Naj bo vložitev

$$id_J : J \hookrightarrow A$$

element istega podobjekta kot id_I . Po zgornji ugotovitvi sta zalogi vrednosti vložitev id_I in id_J enaki, tj. $I = J$ in $id_I = id_J$. Torej, vsak podobjekt kategorije množic vsebuje natanko eno vložitev. Ker so podobjekti ekvivalenčni razredi, ki so paroma disjunktni, med razredom podobjektov dane množice in razredom vseh vložitev v to množico obstaja povratno-enolična preslikava. Sledi, razred $\text{Sub}(A)$ podobjektov objekta A v kategoriji množic **Set** je izomorfen razredu vseh podmnožic množice A . Ta razred je seveda množica.

Ker so vsi monomorfizmi (in s tem vse vložitve) kategorije množic **Set** regularni (primer 9.4), so regularni tudi vsi podobjekti. Za poljubno množico A torej velja: $\text{Sub}(A) = \text{RegSub}(A)$. Iz regularnosti monomorfizmov sledi tudi, da domeno E poljubne vložitve $i : E \hookrightarrow A$ podmnožice E v množico A dobimo kot zožek dveh funkcij $f_1, f_2 : A \rightarrow B$ oz. kot množico rešitev enačbe $f_1(x) = f_2(x)$. Na pojem zožka oz. regularnega podobjekta torej lahko gledamo kot na posplošitev pojma podmnožice ali posplošitev pojma rešitve sistema enačb.

Primer 9.8. Razred $\text{RegSub}(G)$ regularnih podobjektov objekta G v kategoriji grafov **Graph** je izomorfen razredu vseh podgrafov grafa G , o čemer se prepričamo na podoben način kot v primeru 9.7.

Primer 9.9. Razred $\text{RegSub}(L_A)$ regularnih podobjektov polne mreže L_A v kategoriji **CLat** je izomorfen razredu vseh podmrež polne mreže L_A .

Podmnožici, ki imata v relacijski algebri posebno vlogo, sta prazna in polna podmnožica. Poglejmo, katerima podobjektoma dane kategorije pripadata.

Prazno množico lahko na en sam način vložimo v množico A , in sicer s prazno preslikavo, ki je injektivna. Kategorna posplošitev pojma prazne podmnožice oz. prazne preslikave je zato ekvivalenčni razred začetnega monomorfizma, tj.

$$[0_A : \mathbf{0} \rightarrow A],$$

kjer je $\mathbf{0}$ natančen začetni objekt. Začetni objekt kategorije \mathcal{C} je tak objekt $\mathbf{0} \in \mathcal{C}_0$, da za vsak $A \in \mathcal{C}_0$ obstaja natanko en morfizem $0_A \in \text{Hom}(\mathbf{0}, A)$. Morfizem 0_A imenujemo začetni morfizem. Če je domena poljubnega morfizma, katerega kodomena je začetni objekt $\mathbf{0}$, lahko le začetni objekt, $\mathbf{0}$ imenujemo natančen začetni objekt. Pripadajoči začetni morfizmi so tedaj monomorfizmi, ki jih imenujemo začetni monomorfizmi. Glej sliko 9.3.

Velja, da je začetni objekt določen do izomorfizma natanko [Awodey-2005]. Če je začetni objekt natančen, je najmanjši element vsakega delno urejenega razreda podobjektov.

$$\mathbf{0} \xrightarrow{\exists! 0_A} A$$

Slika 9.3: Natančen začetni objekt.

Primer 9.10. Natančen začetni objekt kategorije množic **Set** je prazna množica \emptyset . Edina preslikava iz \emptyset v poljubno množico A je prazna preslikava, ki je injektivna in regularna. Edina preslikava v \emptyset je spet prazna preslikava, katere domena je lahko le \emptyset .

Primer 9.11. Natančen začetni objekt kategorije grafov **Graph** je prazen graf, tj. graf brez vozlišč in brez povezav. Ker so vsi monomorfizmi regularni (primer 9.5), je regularen tudi začetni monomorfizem.

Primer 9.12. Začetni objekt kategorije polnih mrež **CLat** je mreža $L_2 = (2; \wedge, \vee)$, kjer je $2 = \{0, 1\}$ ter velja $0 \wedge 1 = 0$ in $0 \vee 1 = 1$. Edini homomorfizem polnih mrež iz L_2 v poljubno polno mrežo L_A , je namreč monomorfizem polnih mrež, ki slika 0 v 0 in 1 v 1. Polna mreža L_2 ni natančen začetni objekt kategorije **CLat**.

Množico A vložimo vase z identično preslikavo množic, ki je bijektivna. Kategorna posplošitev pojma polne podmnožice je zato ekvivalenčni razred identičnega izomorfizma, tj.

$$[1_A : A \rightarrow A],$$

ki je regularni podobjekt danega objekta A . Dobimo ga kot zožek para identičnih izomorfizmov.

Nujna je tudi kategorna posplošitev podmnožice z enim samim elementom, ki ustreza ekvivalenčnemu razredu

$$[m : \mathbf{1} \rightarrow A],$$

kjer je $\mathbf{1}$ končni objekt, podan ekvivalenčni razred pa podobjekt danega objekta A . Končni objekt kategorije \mathcal{C} je tak objekt $\mathbf{1} \in \mathcal{C}_0$, da za vsak $A \in \mathcal{C}_0$ obstaja natanko en morfizem $!_A \in \text{Hom}(A, \mathbf{1})$. Tak morfizem imenujemo *končni morfizem*. Glej sliko 9.4.

$$A \xrightarrow{\exists! !_A} \mathbf{1}$$

Slika 9.4: Končni objekt.

Velja, da je končni objekt določen do izomorfizma natanko [Awodey-2005].

Primer 9.13. Končni objekt kategorije množic **Set** je množica $\{*\}$ z enim elementom. Vsi monomorfizmi z domeno $\{*\}$ so regularni.

Primer 9.14. *Končni objekt kategorije grafov **Graph** je graf z zanko na edinem vozlišču. Vsi monomorfizmi iz končnega objekta v poljuben graf so regularni.*

Primer 9.15. *Končni objekt kategorije polnih mrež **CLat** je polna mreža $L_1 = (1; \wedge, \vee)$, kjer je $1 = \{1\}$ množica z enim samim elementom, ki je hkrati element 0 in 1. Vsi morfizmi z domeno L_1 so monomorfizmi.*

V razdelku 5.1 smo povedali, da je matematična osnova relacijskega podatkovnega modela relacija, tj. podmnožica kartezičnega produkta množic, ki je ena od petih osnovnih operacij relacijske algebre, definiranih v razdelku 5.2.1. Sledijo kategorne posplošitve vseh definiranih osnovnih operacij, in sicer v naslednjem vrstnem redu:

- \rightsquigarrow kartezični produkt,
- \rightsquigarrow projekcija,
- \rightsquigarrow izbira, unija in razlika.

Vsaki izmed njih bomo priredili ustrezne konstrukte teorije kategorij. Rezultat bo kategorni model relacijskih podatkovnih zbirk, ki bo posplošitev relacijske algebre, definirane v kategoriji množic **Set**.

9.2.1 Posplošitev kartezičnega produkta

Posplošitev kartezičnega produkta na kategorije imenujemo *kategorni produkt*.

Kategorni produkt

Kategorni produkt objektov X_1, \dots, X_n kategorije \mathcal{C} je objekt

$$P = X_1 \times \cdots \times X_n$$

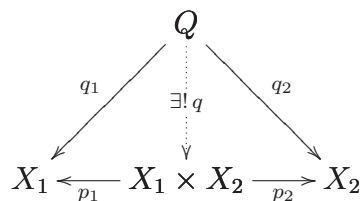
skupaj z množico morfizmov

$$\{p_i : P \rightarrow X_i\}_{1 \leq i \leq n},$$

imenovanih *kategorne projekcije*, pri čemer za vsak objekt $Q \in \mathcal{C}_0$ in poljubno množico morfizmov $\{q_i : Q \rightarrow X_i\}_{1 \leq i \leq n}$ obstaja natanko en morfizem $q : Q \rightarrow P$, da za $1 \leq i \leq n$ velja $p_i \circ q = q_i$. Glej sliko 9.5.

Velja, da je kategorni produkt določen do izomorfizma natanko [Awodey-2005]. Pravimo, da ima kategorija \mathcal{C} končne produkte, če vsebuje kategorni produkt poljubnega končnega števila objektov. To je ekvivalentno obstoju produkta poljubnih dveh objektov in končnega objekta, ki je produkt 0 objektov.

Primer 9.16. *Kategornemu produktu kategorije množic **Set** ustreza kartezični produkt skupaj s projekcijami množic.*

Slika 9.5: Kategorni produkt objektov X_1 in X_2 .

Primer 9.17. Kategornemu produktu kategorije grafov **Graph** ustreza direktni produkt grafov skupaj s projekcijami grafov. Direktni produkt grafov G_1, \dots, G_n je graf $G_1 \times \dots \times G_n$ [Imrich-2000], katerega množica vozlišč je kartezični produkt $V(G_1) \times \dots \times V(G_n)$ in sta vozlišči (u_1, \dots, u_n) in (v_1, \dots, v_n) sosednji natanko tedaj, ko za $1 \leq i \leq n$ velja $u_i v_i \in E(G_i)$. Direktni produkt grafov je eden izmed štirih klasičnih produktov grafov, pojavlja pa se tudi pod drugimi imeni, npr. kardinalni produkt, Kroneckerjev produkt in kategorni produkt grafov.

Primer 9.18. Kategorni produkt $L_A \times L_B$ polnih mrež $L_A = (A; \wedge_A, \vee_A)$ in $L_B = (B; \wedge_B, \vee_B)$ v kategoriji polnih mrež **CLat** je kartezični produkt $A \times B$ z naslednjima binarnima operacijama:

$$(a_1, b_1) \wedge_{A \times B} (a_2, b_2) = (a_1 \wedge_A a_2, b_1 \wedge_B b_2)$$

$$(a_1, b_1) \vee_{A \times B} (a_2, b_2) = (a_1 \vee_A a_2, b_1 \vee_B b_2)$$

Če sta 0_A in 0_B najmanjša elementa polnih mrež L_A in L_B , je $(0_A, 0_B)$ najmanjši element polne mreže $L_A \times L_B$. Podobno, če sta 1_A in 1_B največja elementa polnih mrež L_A in L_B , je $(1_A, 1_B)$ največji element polne mreže $L_A \times L_B$. Velja še:

$$(a_1, b_1) \leq_{A \times B} (a_2, b_2) \iff a_1 \leq_A a_2 \text{ in } b_1 \leq_B b_2$$

Kategorni produkt bo posplošitev kartezičnega produkta na relacijah, če bo kategorni produkt regularnih podobjektov poljubnih objektov A in B kategorije s končnimi kategornimi produkti regularni podobjekt kategornega produkta $A \times B$:

$$\times : \text{RegSub}(A) \times \text{RegSub}(B) \rightarrow \text{RegSub}(A \times B)$$

Pokažimo, da to velja. Naj bosta A in B poljubna regularna objekta kategorije \mathcal{C} , ki ima končne kategorne produkte, A' in B' oz. ekvivalenčna razreda $[m_1 : A' \twoheadrightarrow A]$ in $[m_2 : B' \twoheadrightarrow B]$ pa naj bosta njuna regularna podobjekta. Naj bo objekt $A \times B$ skupaj s kategornim projekcijama

$$p_1 : A \times B \rightarrow A$$

$$p_2 : A \times B \rightarrow B$$

kategorni produkt objektov A in B , objekt $A' \times B'$ skupaj s kategoranima projekcijama

$$p'_1 : A' \times B' \rightarrow A'$$

$$p'_2 : A' \times B' \rightarrow B'$$

pa kategorni produkt objektov A' in B' . Glej sliko 9.6.

$$\begin{array}{ccccc}
 A' & \xleftarrow{p'_1} & A' \times B' & \xrightarrow{p'_2} & B' \\
 \downarrow m_1 & & \downarrow h & & \downarrow m_2 \\
 A & \xleftarrow{p_1} & A \times B & \xrightarrow{p_2} & B
 \end{array}$$

Slika 9.6: Kategorni produkt podobjektov $A' \subseteq A$ in $B' \subseteq B$.

Zaradi lastnosti kategornega produkta obstaja natanko en morfizem

$$h : A' \times B' \rightarrow A \times B,$$

da diagram kategornega produkta $A \times B$ za $Q = A' \times B'$ ter morfizma $m_1 \circ p'_1$ in $m_2 \circ p'_2$ komutira. Dokažimo, da je h monomorfizem. Naj bosta

$$f_1 : P \rightarrow A' \times B'$$

$$f_2 : P \rightarrow A' \times B'$$

poljubna morfizma, za katera velja $h \circ f_1 = h \circ f_2$. Zaradi lastnosti kategornega produkta med objektoma P in $A' \times B'$ obstaja natanko en morfizem, da diagram kategornega produkta $A' \times B'$ za objekt P ter morfizma $p'_1 \circ f_1$ in $p'_2 \circ f_1$ komutira. Sledi, $f_1 = f_2$ in h je monomorfizem. Monomorfizem h je tudi regularen – dobimo ga kot zožek dveh urejenih parov morfizmov, ki nastopajo v zožkih monomorfizmov m_1 in m_2 . Objekt $A' \times B'$ oz. ekvivalenčni razred

$$[h : A' \times B' \rightarrow A \times B],$$

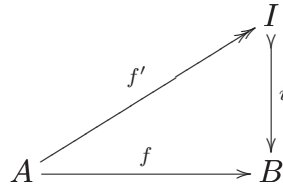
pripadajoč podobjektoma $[m_1 : A' \rightarrow A]$ in $[m_2 : B' \rightarrow B]$, je tedaj regularni podobjekt objekta $A \times B$.

9.2.2 Posplošitev projekcije

Za posplošitev projekcije rabimo kategorno projekcijo, ki nastopa v definiciji kategornega produkta, in *slike*.

Slike

Naj bo $f : A \rightarrow B$ morfizem kategorije \mathcal{C} . Naj obstaja *faktorizacija* morfizma f , tj. epimorfizem $f' : A \rightarrow I$ in monomorfizem $i : I \rightarrow B$, da velja $f = i \circ f'$ (slika 9.7). Najmanjši podobjekt I objekta B , ki izpolnjuje dane pogoje, imenujemo *slika* morfizma f in ga označimo z $im(f)$.



Slika 9.7: Faktorizacija morfizma f .

Slika je očitno določena do izomorfizma natanko. Pravimo, da ima kategorija \mathcal{C} slike, če vsebuje sliko poljubnega morfizma.

Primer 9.19. Kategorija množic **Set** ima slike. Slika preslikave $f : A \rightarrow B$ je množica $im(f) = f[A] = \{y \in B : \exists x \in A : f(x) = y\}$, ki ustreza faktorizaciji s surjektivno preslikavo $f' : A \rightarrow im(f)$, ki se od preslikave f razlikuje le v kodomeni, in vložitvijo $id_{im(f)} : im(f) \hookrightarrow B$.

Primer 9.20. Kategorija grafov **Graph** ima slike. Slika homomorfizma $f : G \rightarrow H$ je podgraf $im(f) = f[G] = (V', E')$ grafa H z množico vozlišč

$$V' = \{v \in V(H) : \exists u \in V(G) : f(u) = v\}$$

in množico povezav

$$E' = \{v_1 v_2 \in E(H) : \exists u_1 u_2 \in E(G) : f(u_1) = v_1 \text{ in } f(u_2) = v_2\}.$$

Slika ustreza faktorizaciji z epimorfizmom $f' : G \rightarrow im(f)$, ki se od preslikave f razlikuje le v kodomeni, in vložitvijo $id_{im(f)} : im(f) \hookrightarrow H$.

Primer 9.21. Kategorija polnih mrež **CLat** ima slike. Slika morfizma

$$f : (A; \wedge_A, \vee_A) \rightarrow (B; \wedge_B, \vee_B)$$

je polna podmreža $im(f) = (B'; \wedge_B, \vee_B)$ mreže $(B; \wedge_B, \vee_B)$, kjer je $B' = \{y \in B : \exists x \in A : f(x) = y\}$. Slika ustreza faktorizaciji z epimorfizmom $f' : (A; \wedge_A, \vee_A) \rightarrow im(f)$, ki se od preslikave f razlikuje le v kodomeni, in vložitvijo $id_{im(f)} : im(f) \hookrightarrow (B; \wedge_B, \vee_B)$.

Kategorna projekcija

Primer 9.22. Kategorni projekciji kategorije množic **Set** ustreza projekcija množic. To je, i -ta kategorna projekcija kartezičnega produkta $A_1 \times \cdots \times A_n$ je množica A_i , pri čemer je i -ta kategorna projekcija elementa $(a_1, \dots, a_n) \in A_1 \times \cdots \times A_n$ element $a_i \in A_i$.

Primer 9.23. Kategorni projekciji kategorije grafov **Graph** ustreza projekcija grafov. To je, i -ta kategorna projekcija grafa $G_1 \times \cdots \times G_n$ je graf G_i .

Primer 9.24. Kategorni projekciji kategorije polnih mrež **CLat** ustreza projekcija polnih mrež. To je, i -ta kategorna projekcija polne mreže $L_1 \times \cdots \times L_n$ je polna mreža L_i .

Kategorna projekcija bo posplošitev projekcije na relacijah, če bosta obe kategorni projekciji poljubnega regularnega podobjekta danega kategornega produkta $A \times B$ kategorije s končnimi kategornimi produkti in slikami regularna podobjekta objekta A oz. B :

$$p_1 : \text{RegSub}(A \times B) \rightarrow \text{RegSub}(A)$$

$$p_2 : \text{RegSub}(A \times B) \rightarrow \text{RegSub}(B)$$

Naj bosta A in B poljubna objekta kategorije \mathcal{C} , ki ima končne kategorne produkte in slike, objekt C oz. ekvivalenčni razred $[m : C \rightarrow A \times B]$ pa regularni podobjekt objekta $A \times B$. Naj bo

$$p_1 : A \times B \rightarrow A$$

kategorna projekcija $A \times B$ na prvo komponento, katere obstoj pogojuje obstoj kategornega produkta, $im(p_1 \circ m)$ pa slika objekta C z morfizmom $p_1 \circ m$. Sliki naj pripada faktorizacija morfizma $p_1 \circ m$ z epimorfizmom $e : C \rightarrow im(p_1 \circ m)$ in monomorfizmom $m_1 : im(p_1 \circ m) \rightarrow A$. Glej sliko 9.8.

$$\begin{array}{ccc} im(p_1 \circ m) & \xleftarrow{e} & C \\ \downarrow m_1 & & \downarrow m \\ A & \xleftarrow{p_1} & A \times B \end{array}$$

Slika 9.8: Kategorna projekcija podobjekta $C \subseteq A \times B$.

Ekvivalenčni razred

$$[m_1 : im(p_1 \circ m) \rightarrow A],$$

pripadajoč podobjektu $[m : C \rightarrow A \times B]$, je tedaj regularni podobjekt objekta A . Dobimo ga kot zožek prvih komponent morfizmov iz zožka $m : C \rightarrow A \times B$. Za p_2 je dokaz enak.

9.2.3 Posplošitev izbire, unije in razlike

Ostala nam je še kategorna posplošitev treh osnovnih operacij relacijske algebre – izbire, unije in razlike. Spomnimo, da je razred regularnih podobjektov $\text{RegSub}(A)$ objekta A delno urejen z relacijo \leq . V kategoriji množic **Set** so regularni podobjekti množice A vse njene podmnožice, razred regularnih podobjektov množice A pa potenčna množica $\mathcal{P}(A)$, ki je delno urejena z relacijo vsebovanosti \subseteq . Videli bomo, da bo operacije izbira (tj. največja spodnja meja), unija (tj. najmanjša zgornja meja) in razlika (tj. relativni komplement), ki delujejo na parih delno urejene potenčne množice $\mathcal{P}(A)$, mogoče posplošiti na teorijo kategorij, če bo razred regularnih podobjektov $\text{RegSub}(A)$ poljubnega objekta A vseboval največjo spodnjo (inf) in najmanjšo zgornjo mejo (sup) poljubnih regularnih podobjektov ter vse relativne komplemente.

Kategorna izbira

Kategorno izbiro izračunamo z uporabo kategornega preseka. Kategorni presek

$$\cap : \text{RegSub}(C) \times \text{RegSub}(C) \rightarrow \text{RegSub}(C)$$

na delno urejenem razredu regularnih podobjektov objekta C definiramo kot operacijo največja spodnja meja (inf) za relacijo delne urejenosti \leq . Operacija kategorni presek torej paru regularnih podobjektov A, B objekta C priredi največji regularni podobjekt $A \cap B$ objekta C , za katerega velja:

$$\begin{aligned} A \cap B &\leq A \\ A \cap B &\leq B \end{aligned}$$

Naj bo C poljuben objekt kategorije \mathcal{C} , ki ima v razredu regularnih podobjektov končne kategorne preseke, $A \subseteq C$ regularni podobjekt z izbrano lastnostjo, $B \subseteq C$ pa poljuben regularni podobjekt. Kategorni presek $A \cap B$, ki je regularni podobjekt objekta C , imenujemo *kategorna izbira* regularnega podobjekta objekta B , ki zadošča izbrani lastnosti regularnega podobjekta A . Kategorna izbira σ je (tako kot kategorni presek) preslikava

$$\sigma : \text{RegSub}(C) \times \text{RegSub}(C) \rightarrow \text{RegSub}(C),$$

ki paru regularnih podobjektov objekta C priredi regularni podobjekt objekta C . Kategorno izbiro in kategorni presek modeliramo z isto operacijo, tj. operacijo največja spodnja meja.

Primer 9.25. *Naj bo A množica kategorije množic **Set** z izbrano lastnostjo (tj. množica vseh elementov, ki zadoščajo dani formuli F), B pa poljubna množica. Kategorna izbira podmnožice množice B , ki zadošča izbrani lastnosti, je presek množic A in B .*

Primer 9.26. *Naj bo G_1 graf v kategoriji grafov **Graph** z izbranimi vozlišči in povezavami, G_2 pa poljuben graf. Kategorna izbira podgrafa grafa G_2 , katerega vozlišča in povezave zadoščajo izbiri, podani z grafom G_1 , je presek grafov G_1 in G_2 .*

Kategorna unija

Kategorno unijo

$$\cup : \text{RegSub}(C) \times \text{RegSub}(C) \rightarrow \text{RegSub}(C)$$

v delno urejenem razredu regularnih podobjektov objekta C definiramo kot operacijo najmanjša zgornja meja (sup) za relacijo delne urejenosti \leq . Operacija kategorna unija torej paru A, B regularnih podobjektov objekta C priredi najmanjši regularni podobjekt $A \cup B$ objekta C , za katerega velja:

$$\begin{aligned} A &\leq A \cup B \\ B &\leq A \cup B \end{aligned}$$

Primer 9.27. *Kategorna unija podmnožic A in B množice C v kategoriji množic **Set** je unija $A \cup B$ množic A in B , ki je podmnožica množice C .*

Primer 9.28. *Kategorna unija podgrafov G_1 in G_2 grafa G v kategoriji grafov **Graph** je unija $G_1 \cup G_2$ grafov G_1 in G_2 , tj. graf z množico vozlišč $V(G_1) \cup V(G_2)$ in množico povezav $E(G_1) \cup E(G_2)$, ki je podgraf grafa G .*

Primer 9.29. *Kategorna unija podmrež L_A in L_B polne mreže L_C v kategoriji polnih mrež **CLat** je unija podmrež L_A in L_B , dopolnjena z vsemi manjkajočimi največjimi spodnjimi in najmanjšimi zgornjimi mejami.*

Kategorna razlika

Kategorno razliko

$$- : \text{RegSub}(C) \times \text{RegSub}(C) \rightarrow \text{RegSub}(C)$$

v delno urejenem razredu regularnih podobjektov objekta C kategorije \mathcal{C} , ki ima v razredu regularnih podobjektov končne kategorne preseke in končne kategorne unije, definiramo kot operacijo, ki paru regularnih podobjektov A, B objekta C priredi regularni podobjekt $A - B$ objekta C , za katerega velja:

$$(A \cap B) \cup (A - B) = A$$

$$B \cap (A - B) = \mathbf{0}$$

Z $\mathbf{0}$ smo označili začetni objekt kategorije \mathcal{C} . Relativni komplement regularnega podobjekta B v objektu C tedaj izračunamo kot

$$B^{\mathbf{c}} = C - B.$$

Ker velja

$$A - B = A \cap B^{\mathbf{c}},$$

kategorne razlike v $\text{RegSub}(C)$ obstajajo natanko tedaj, ko obstajajo relativni komplementi. Za kategorno razliko $A - B$ velja:

$$A - B \leq (A \cap B) \cup (A - B) = A$$

Primer 9.30. Naj bosta A in B podmnožici množice C v kategoriji množic **Set**. Razlika množic

$$A - B = \{x \in A; x \notin B\}$$

je podmnožica množice C , za katero velja $(A \cap B) \cup (A - B) = A$ in $B \cap (A - B) = \emptyset$. Ker je \emptyset začetni objekt kategorije množic (primer 9.10), je razlika množic kategorialna razlika v kategoriji **Set**.

Primer 9.31. Naj bosta G_1 in G_2 podgrafa grafa G kategorije grafov **Graph**. Vzemimo presek $G_1 \cap G_2$ in poskušajmo poiskati kategorialno razliko $G_1 - G_2$. Veljati mora

$$(G_1 \cap G_2) \cup (G_1 - G_2) = G_1,$$

kar za množice vozlišč in povezav pomeni:

$$V(G_1 \cap G_2) \cup V(G_1 - G_2) = V(G_1)$$

$$E(G_1 \cap G_2) \cup E(G_1 - G_2) = E(G_1)$$

Iz pogoja za množico vozlišč grafa $G_1 - G_2$ sledi

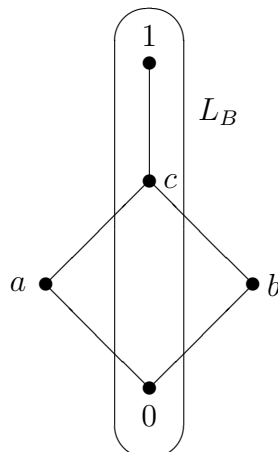
$$V(G_1 - G_2) = V(G_1) - V(G_2).$$

Recimo, da obstajata taki vozlišči $u \in V(G_1 \cap G_2)$ in $v \in V(G_1 - G_2)$, da je povezava $uv \in E(G_1)$. Primer grafov, ki temu zadoščata, sta:

$$\begin{array}{ll} G_1 : & V(G_1) = \{u, v\} \quad E(G_1) = \{uv\} \\ G_2 : & V(G_2) = \{u\} \quad E(G_2) = \emptyset \end{array}$$

Ker povezava uv ni element $E(G_1 \cap G_2)$, iz pogoja za množico povezav grafa $G_1 - G_2$ sledi $uv \in E(G_1 - G_2)$, kar pa ne more veljati, saj $u \notin V(G_1 - G_2)$. Sledi, da v kategoriji grafov **Graph** kategorialna razlika regularnih podobjektov ne obstaja.

Primer 9.32. V kategoriji polnih mrež **CLat** kategorialna razlika regularnih podobjektov ne obstaja. Kot zgled vzemimo polno mrežo L_A in njeno regularno podmrežo L_B :



Za kategorno razliko mrež $L_A - L_B$ mora po definiciji veljati naslednje:

$$(L_A \cap L_B) \cup (L_A - L_B) = L_A$$

$$L_B \cap (L_A - L_B) = L_2$$

Ker mora biti $L_A - L_B$ regularni podobjekt objekta L_A , torej podmreža polne mreže L_A , mora vsebovati najmanjši element 0 in največji element 1. Zaradi prvega pogoja mora $L_A - L_B$ vsebovati tudi a in b , posledično pa še najmanjšo zgornjo mejo elementov a in b , tj. c . Sledi, $L_A - L_B = L_A$, kar je v nasprotju z drugim pogojem. Kategorna razlika $L_A - L_B$ torej ne obstaja.

9.2.4 Sklep

Videli smo, da za posplošitev relacijske algebre iz običajne teorije množic na teorijo kategorij potrebujemo:

- ↪ natančen začetni objekt, ki je posplošitev pojma prazne podmnožice,
- ↪ končni objekt, ki je posplošitev pojma podmnožice z enim elementom.

Za posplošitev kartezičnega produkta in projekcije (tj. dveh izmed petih osnovnih operacij relacijske algebre) potrebujemo še:

- ↪ končne kategorne produkte in
- ↪ slike.

Delno urejen razred $\text{RegSub}(C)$ regularnih podobjektov poljubnega objekta C kategorije, v kateri bo mogoče modelirati posplošeno relacijsko algebro, mora imeti kategorni produkt poljubnega para regularnih podobjektov. Za poljuben regularni podobjekt poljubnega kategornega produkta $A \times B$ morata obstajati tudi sliki kompozicij kategornih projekcij $p_1 : A \times B \rightarrow A$ in $p_2 : A \times B \rightarrow B$ ter regularnemu podobjektu pripadajočega regularnega monomorfizma.

Kategorna posplošitev preostalih treh osnovnih operacij relacijske algebre (izbire, unije in razlike) zahteva dodatne lastnosti delno urejenih razredov regularnih podobjektov. Poljubna regularna podobjekta danega objekta A morata imeti enolično določeni

- ↪ največjo spodnjo ter
- ↪ najmanjšo zgornjo mejo

glede na relacijo \leq in v poljubnem razredu $\text{RegSub}(A)$ mora obstajati

- ↪ relativni komplement.

Kategorije, ki izpolnjujejo vse naštetje pogoje, so primerna izbira za kategorno predstavitev relacijske algebre. Kategorija množic **Set** je prva in najnaravnejša tovrstna kategorija, tj. kategorija, iz katere smo izhajali. Kategorija grafov **Graph** in kategorija polnih mrež **CLat** pa nista dobri, saj npr. nimata kategorne razlike (primeri 9.31 in 9.32).

Za delno urejene razrede $(\text{RegSub}(A), \leq)$ bi lahko zahtevali še kakšno dodatno lastnost, npr. distributivnost operacij \cap in \cup :

$$(X \cup Y) \cap Z = (X \cap Z) \cup (Y \cap Z)$$

$$(X \cap Y) \cup Z = (X \cup Z) \cap (Y \cup Z)$$

To je namreč ena izmed mnogih lastnosti, ki jim zadoščajo operacije, definirane v kategoriji množic **Set**. Glede na zahtevnost in namembnost podatkovnih zbirk, ki jih želimo modelirati, lahko v dani situaciji izberemo dodatne lastnosti, ki naj jim zelene kategorije zadoščajo.

9.3 Kategorija podobnosti

V razdelku 5.1 o relacijski algebri smo povedali, da relacijski podatkovni model, ki je definiran v kategoriji množic **Set**, temelji na relacijah, tj. podmnožicah kartezičnega produkta ene ali več vrednostnih množic. Ker te množice niso urejene, relacijske podatkovne zbirke podpirajo le standardne urejenosti podatkovnih tipov, tj. abecedno urejenost črkovnih nizov, numerično urejenost števil in kronološko urejenost datumov. Omejena podpora urejenosti v relacijskih podatkovnih zbirkah pa privede do izgube pomembne semantične informacije (razdelek 8.5.2), kot je npr. sposobnost predstavitve sorodnih podatkov. Posledica neurejenosti vrednostnih množic, na katerih temelji relacijska algebra, je zato nesposobnost posploševanja poizvedb in odgovorov oz. relaksacije (razdelek 8.3.5).

Da omogočimo relaksacijo poizvedb in s tem pomembno zvišamo raven sodelujočega odgovarjanja, definirajmo *kategorijo podobnosti*, katere objekti niso navadne množice kot pri relacijski algebri, temveč množice, opremljene z mero podobnosti. Mera podobnosti opisuje podobnost, relacije in/ali razdalje med elementi dane množice. Videli bomo, da ima kategorija podobnosti vse lastnosti, potrebne za kategorno predstavitev relacijske algebre.

Definicija 9.33. Kategorijo podobnosti **SimObj** definirajmo takole:

1. Objekti naj bodo urejene trojice

$$(A, L_A, \sigma_A),$$

kjer je A množica, L_A polna mreža z najmanjšim elementom 0_A in največjim elementom 1_A ,

$$\sigma_A : A \times A \rightarrow L_A$$

pa mera podobnosti v množici A , za katero za vsak $x \in A$ velja

$$\sigma_A(x, x) = 1_A.$$

2. Morfizmi naj bodo preslikave oblike

$$(f, \varphi) : (A, L_A, \sigma_A) \rightarrow (B, L_B, \sigma_B),$$

kjer je

$$f : A \rightarrow B$$

preslikava množic,

$$\varphi : L_A \rightarrow L_B$$

pa homomorfizem polnih mrež (primer 9.3), za katerega vemo, da spoštuje urejenost. Preslikavi f in φ , meri podobnosti σ_A in σ_B ter delna urejenost \leq_B polne mreže L_B naj bodo v naslednji zvezi:

$$\forall x, y \in A : \varphi(\sigma_A(x, y)) \leq_B \sigma_B(f(x), f(y))$$

Morfizmi pretvarjajo podobnost v množici A v podobnost v množici B . Podobnost poljubnega urejenega para elementov pri tem kvečjemu povečajo. Kompozicijo morfizmov

$$(f, \varphi) : (A, L_A, \sigma_A) \rightarrow (B, L_B, \sigma_B)$$

$$(g, \psi) : (B, L_B, \sigma_B) \rightarrow (C, L_C, \sigma_C)$$

definirajmo takole:

$$(g, \psi) \circ (f, \varphi) = (g \circ f, \psi \circ \varphi)$$

Enota za kompozicijo je par identičnih izomorfizmov

$$(1_A, 1_{L_A}) : (A, L_A, \sigma_A) \rightsquigarrow (A, L_A, \sigma_A),$$

ki množico A in polno mrežo L_A identično preslika sami vase.

Monomorfizmi kategorije **SimObj** so morfizmi oblike

$$(f, \varphi) : (A, L_A, \sigma_A) \rightsquigarrow (B, L_B, \sigma_B),$$

kjer je f injektivna preslikava množic, φ pa monomorfizem polnih mrež (primer 9.6).

Dobro je vedeti tudi, kdaj sta dva objekta izomorfna. Morfizem

$$(f, \varphi) : (A, L_A, \sigma_A) \rightarrow (B, L_B, \sigma_B)$$

je izomorfizem kategorije podobnosti **SimObj**, če obstaja morfizem

$$(g, \psi) : (B, L_B, \sigma_B) \rightarrow (A, L_A, \sigma_A),$$

za katerega velja $f \circ g = 1_B$ in $g \circ f = 1_A$ ter $\varphi \circ \psi = 1_{L_B}$ in $\psi \circ \varphi = 1_{L_A}$. Ker sta f in g preslikavi množic, sta bijektivni. Tudi φ in ψ , ki sta homomorfizma polnih mrež, sta bijektivna. Po definiciji morfizma za mero podobnosti σ_A velja

$$\varphi(\sigma_A(x, y)) \leq_B \sigma_B(f(x), f(y)),$$

in obratno – po komponiranju s homomorfizmom polnih mrež φ iz

$$\psi(\sigma_B(f(x), f(y))) \leq_A \sigma_A(g(f(x)), g(f(y)))$$

sledi

$$\sigma_B(f(x), f(y)) \leq_B \varphi(\sigma_A(x, y)).$$

Izomorfizmi kategorije podobnosti so torej sestavljeni iz bijektivne preslikave množic f in izomorfizma polnih mrež φ , za katera velja

$$\varphi(\sigma_A(x, y)) = \sigma_B(f(x), f(y)).$$

Pišemo

$$(f, \varphi) : (A, L_A, \sigma_A) \rightsquigarrow (B, L_B, \sigma_B).$$

Podobjekti, ki predstavljajo poizvedbe

Namesto da bi računali regularne podobjekte objekta $\underline{A} = (A, L_A, \sigma_A)$ kategorije podobnosti **SimObj** in razlagali njihovo vlogo, se raje osredotočimo na podobjekte, ki predstavljajo poizvedbe. Če objekti ustrezajo relacijskim tabelam, so edini možni rezultati poizvedb podtabele z inducirano urejenostjo oz. podobnostjo. Te pa ustrezajo objektom oblike $\underline{B} = (B, L_A, \sigma_A)$, kjer je B podmnožica množice A ,

$$(id_B, 1_{L_A}) : (B, L_A, \sigma_A) \hookrightarrow (A, L_A, \sigma_A)$$

pa vložitev objekta (B, L_A, σ_A) v objekt (A, L_A, σ_A) . Pri tem je id_B injektivna vložitev podmnožice B v množico A , 1_{L_A} pa identični izomorfizem polne mreže L_A . Množico vseh *induciranih podobjektov* oblike (B, L_A, σ_A) , ki je očitno izomorfna potenčni množici $\mathcal{P}(A)$, bomo označevali s $\mathcal{P}(\underline{A})$.

Pokažimo, da je množica $\mathcal{P}(\underline{A})$ podmnožica razreda $\text{RegSub}(\underline{A})$, to je

$$\mathcal{P}(\underline{A}) \subseteq \text{RegSub}(\underline{A}).$$

To bo res, če bodo vse vložitev objektov iz $\mathcal{P}(\underline{A})$ v objekt \underline{A} regularni monomorfizmi. Vzemimo vložitev $(id_B, 1_{L_A})$ objekta $\underline{B} \in \mathcal{P}(\underline{A})$ v objekt \underline{A} . Pokažimo, da je vložitev $(id_B, 1_{L_A})$ zožek morfizmov

$$(f, !_A), (g, !_A) : (A, L_A, \sigma_A) \rightarrow (\{0, 1\}, L_1, \sigma_1),$$

kjer je f preslikava, ki vse elemente množice A preslika v 1, g pa preslikava s predpisom:

$$g(x) = \begin{cases} 1, & x \in B \\ 0, & \text{sicer} \end{cases}$$

Za $x \in B$ velja $f \circ id_B(x) = f(x) = 1 = g(x) = g \circ id_B(x)$ in $!_A \circ 1_{L_A}(x) = !_A(x) = 1$. Naj bo torej

$$(e', \varphi') : (B', L_{B'}, \sigma_{B'}) \rightarrow (A, L_A, \sigma_A)$$

poljuben morfizem, ki zadošča pogoju $f \circ e' = g \circ e'$. Tedaj za $y \in B'$ velja

$$f(e'(y)) = g(e'(y)) = 1$$

in zato $e'(y) \in B$. Med objektoma $(B', L_{B'}, \sigma_{B'})$ in (B, L_A, σ_A) torej obstaja morfizem (k, φ') , ki se od (e', φ') razlikuje le v kodomeni preslikave množic. Homomorfizem polnih mrež φ' je edini možen homomorfizem med $L_{B'}$ in L_A , pri katerem diagram morfizmov komutira. Ker je (k, φ') enolično določen, je $(id_B, 1_{L_A})$ zožek morfizmov $(f, !_A)$, $(g, !_A)$ in zato regularen monomorfizem.

Torej res, vse vložitve objektov iz $\mathcal{P}(\underline{A})$ v objekt \underline{A} so regularni monomorfizmi in množica $\mathcal{P}(\underline{A})$ je podmnožica razreda $\text{RegSub}(\underline{A})$.

Najmanjši element v $\mathcal{P}(\underline{A})$

Začetni objekt kategorije podobnosti **SimObj** je objekt $(\emptyset, L_2, \sigma_1)$, kjer je prazna množica \emptyset začetni objekt kategorije množic (primer 9.10), L_2 začetni objekt kategorije polnih mrež (primer 9.12), σ_1 pa mera podobnosti, ki vse pare preslika v 1. Edini morfizem iz $(\emptyset, L_2, \sigma_1)$ v poljuben objekt $\underline{A} = (A, L_A, \sigma_A)$ je morfizem, sestavljen iz prazne preslikave množic in začetnega homomorfizma polnih mrež, ki 0 preslika v 0 in 1 v 1. Ker polna mreža L_2 ni natančen začetni objekt kategorije polnih mrež, objekt $(\emptyset, L_2, \sigma_1)$ ni natančen začetni objekt kategorije podobnosti.

Natančen začetni objekt smo zahtevali zaradi posplošitve pojma prazne podmnožice, ki je kot natančen začetni objekt kategorije množic **Set** najmanjši podobjekt poljubnega delno urejenega razreda regularnih podobjektov. Čeprav začetni objekt $(\emptyset, L_2, \sigma_1)$ kategorije **SimObj** ni natančen, mu pripadajo začetni monomorfizmi in zato prevzame vlogo prazne podmnožice v razredih regularnih podobjektov $\text{RegSub}(\underline{A})$.

Vlogo, ki jo $(\emptyset, L_2, \sigma_1)$ ima v $\text{RegSub}(\underline{A})$, v delno urejeni množici $\mathcal{P}(\underline{A})$ prevzame podobjekt $(\emptyset, L_A, \sigma_A)$. Objekt $(\emptyset, L_A, \sigma_A)$ je najmanjši element v $\mathcal{P}(\underline{A})$. Največji element v $\mathcal{P}(\underline{A})$ pa je objekt $\underline{A} = (A, L_A, \sigma_A)$.

Končni objekt

Končni objekt kategorije podobnosti **SimObj** je objekt $(\{*\}, L_1, \sigma_1)$, kjer je $\{*\}$ končni objekt kategorije množic (primer 9.13), za katerega velja $\sigma_1(*, *) = 1$, L_1 pa končni objekt kategorije polnih mrež (primer 9.15).

Končni objekt smo zahtevali zaradi posplošitve pojma podmnožice z enim samim elementom. Vlogo, ki jo $(\{*\}, L_1, \sigma_1)$ ima v $\text{RegSub}(\underline{A})$, v delno urejeni množici $\mathcal{P}(\underline{A})$ prevzame induciran podobjekt $(\{*\}, L_A, \sigma_A)$ objekta (A, L_A, σ_A) .

Končni kategorni produkti

Pokažimo, da ima kategorija podobnosti **SimObj** končne produkte. Naj bosta

$$\underline{A} = (A, L_A, \sigma_A)$$

$$\underline{B} = (B, L_B, \sigma_B)$$

poljubna objekta kategorije **SimObj**. Dokažimo, da kategornemu produktu $\underline{A} \times \underline{B}$ ustreza objekt

$$\underline{A} \times \underline{B} = (A \times B, L_A \times L_B, \sigma_{A \times B}),$$

kjer za poljubna $(x, y), (x', y') \in A \times B$ velja

$$\sigma_{A \times B}((x, y), (x', y')) = (\sigma_A(x, x'), \sigma_B(y, y')).$$

Pri tem je $L_A \times L_B$ produkt polnih mrež (primer 9.18),

$$(\pi_1, p_1) : \underline{A} \times \underline{B} \rightarrow \underline{A}$$

$$(\pi_2, p_2) : \underline{A} \times \underline{B} \rightarrow \underline{B}$$

pa pripadajoči projekciji, sestavljeni iz projekcij množic $\pi_1 : A \times B \rightarrow A$ in $\pi_2 : A \times B \rightarrow B$ ter projekcij polnih mrež $p_1 : L_A \times L_B \rightarrow L_A$ in $p_2 : L_A \times L_B \rightarrow L_B$. Ker za poljuben $(x, y) \in A \times B$ in največja elementa 1_A in 1_B polnih mrež L_A in L_B velja

$$\sigma_{A \times B}((x, y), (x, y)) = (\sigma_A(x, x), \sigma_B(y, y)) = (1_A, 1_B) = 1_{A \times B},$$

je $\sigma_{A \times B}$ ustrezna mera podobnosti, $\underline{A} \times \underline{B}$ pa objekt kategorije **SimObj**. Preveriti je treba še, ali sta projekciji (π_1, p_1) in (π_2, p_2) morfizma. Odgovor je da:

$$p_1(\sigma_{A \times B}((x, y), (x', y'))) = p_1(\sigma_A(x, x'), \sigma_B(y, y')) = \sigma_A(x, x')$$

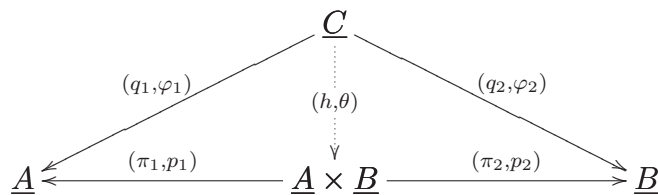
$$p_2(\sigma_{A \times B}((x, y), (x', y'))) = p_2(\sigma_A(x, x'), \sigma_B(y, y')) = \sigma_B(y, y')$$

Naj bo torej $\underline{C} = (C, L_C, \sigma_C)$ poljuben objekt,

$$(q_1, \varphi_1) : \underline{C} \rightarrow \underline{A}$$

$$(q_2, \varphi_2) : \underline{C} \rightarrow \underline{B}$$

pa morfizma kategorije **SimObj**. Glej sliko 9.9.



Slika 9.9: Kategorni produkt v kategoriji podobnosti.

Pokazati moramo, da obstaja natanko en morfizem

$$(h, \theta) = ((h_1, h_2), (\theta_1, \theta_2)) : \underline{C} \rightarrow \underline{A} \times \underline{B},$$

da velja:

$$(\pi_1 \circ h, p_1 \circ \theta) = (q_1, \varphi_1)$$

$$(\pi_2 \circ h, p_2 \circ \theta) = (q_2, \varphi_2)$$

Ker tedaj za $i = 1, 2$ velja

$$\pi_i \circ h(x) = \pi_i(h_1(x), h_2(x)) = h_i(x) = q_i(x),$$

je h s preslikavama q_1 in q_2 enolično določena:

$$h = (h_1, h_2) = (q_1, q_2)$$

Podobno, ker za $i = 1, 2$ velja

$$p_i \circ \theta(x) = p_i(\theta_1(x), \theta_2(x)) = \theta_i(x) = \varphi_i(x),$$

je θ s preslikavama φ_1 in φ_2 enolično določena:

$$\theta = (\theta_1, \theta_2) = (\varphi_1, \varphi_2)$$

Ker sta φ_1 in φ_2 homomorfizma polnih mrež, je tudi preslikava θ homomorfizem polnih mrež. Ker sta (q_1, φ_1) in (q_2, φ_2) morfizma kategorije podobnosti, preslikava (h, θ) zadošča pogoju

$$\theta(\sigma_C(x, y)) \leq \sigma_{A \times B}(h(x), h(y))$$

in je zato morfizem. Sledi, $\underline{A} \times \underline{B}$ je kategori produkt objektov \underline{A} in \underline{B} . Ker ima kategorija **SimObj** tudi končni objekt, ima končne kategorne produkte.

Slike

Slike smo zahtevali zaradi posplošitve projekcije (razdelek 9.2.2). V resnici potrebujemo le slike kompozicij projekcij in regularnih monomorfizmov (razdelek 9.2.4), v primeru kategorije podobnosti **SimObj** pa le slike kompozicij projekcij in vložitev. Pokažimo torej, da kategorija **SimObj** dovoljuje posplošitev projekcije. Naj bosta

$$\underline{A} = (A, L_A, \sigma_A)$$

$$\underline{B} = (B, L_B, \sigma_B)$$

poljubna objekta kategorije podobnosti **SimObj**, $\underline{A} \times \underline{B}$ pa njun kategori produkt. Vprašajmo se, kaj naj bi bili kategori projekciji (π_1, p_1) in (π_2, p_2) objekta

$$\underline{X} = (X, L_A \times L_B, \sigma_{A \times B}) \in \mathcal{P}(\underline{A} \times \underline{B}) \subseteq \text{RegSub}(\underline{A} \times \underline{B}).$$

Pričakujemo, da sta to objekta

$$\pi_1(\underline{X}) = (\pi_1(X), L_A, \sigma_A) \in \mathcal{P}(\underline{A})$$

$$\pi_2(\underline{X}) = (\pi_2(X), L_B, \sigma_B) \in \mathcal{P}(\underline{B})$$

kjer je $\pi_1(X) = \{x \in A; \exists y \in B : (x, y) \in X\}$ in $\pi_2(X) = \{y \in B; \exists x \in A : (x, y) \in X\}$. Glej sliko 9.10. Preverimo, ali sta to tudi slike v smislu podobjektov (razdelek 9.2.2). Pokažimo, da je $\pi_1(\underline{X})$ najmanjši podobjekt objekta \underline{A} , skozi katerega lahko faktoriziramo morfizem $(\pi_1 \circ id_X, p_1 \circ 1_{L_A \times L_B})$.

$$\begin{array}{ccc}
\pi_1(\underline{X}) = (\pi_1(X), L_A, \sigma_A) & \xleftarrow{(\pi_1, p_1)} & (X, L_A \times L_B, \sigma_{A \times B}) = \underline{X} \\
\downarrow (id_{\pi_1(X)}, 1_{L_A}) & & \downarrow (id_X, 1_{L_A \times L_B}) \\
\underline{A} = (A, L_A, \sigma_A) & \xleftarrow{(\pi_1, p_1)} & (A \times B, L_A \times L_B, \sigma_{A \times B}) = \underline{A} \times \underline{B}
\end{array}$$

Slika 9.10: Kategorna projekcija v kategoriji podobnosti.

Naj bo (J, L_J, σ_J) poljuben podobjekt objekta \underline{A} , skozi katerega lahko faktoriziramo $(\pi_1 \circ id_X, p_1 \circ 1_{L_A \times L_B})$, (q, φ) pripadajoč epimorfizem, (m, ψ) pa pripadajoč monomorfizem. Ker za poljuben $x \in \pi_1(X)$ obstaja $y \in B$, da je $(x, y) \in X$, in za poljuben $a \in L_A$ obstaja $b \in L_B$, da je $(a, b) \in L_A \times L_B$, obstaja morfizem

$$(h, \theta) : (\pi_1(X), L_A, \sigma_A) \rightarrow (J, L_J, \sigma_J)$$

s predpisom $h(x) = q(x, y)$ in $\theta(a) = \varphi(a, b)$, za katerega diagram morfizmov komutira. Sledi, objekt $\pi_1(\underline{X})$ je najmanjši podobjekt, skozi katerega lahko faktoriziramo morfizem $(\pi_1 \circ id_X, p_1 \circ 1_{L_A \times L_B})$. Podobno za $\pi_2(\underline{X})$. Kategorija podobnosti **SimObj** torej ima slike kompozicij projekcij in vložitev, in zato obe projekciji poljubnega objekta $\underline{X} \in \mathcal{P}(\underline{A} \times \underline{B})$.

Kategorna izbira, unija in razlika

Ker je množica $\mathcal{P}(\underline{A}) \subseteq \text{RegSub}(\underline{A})$ izomorfná potenčni množici $\mathcal{P}(A)$, sledi, da v množici $\mathcal{P}(\underline{A})$ obstajajo kategorna izbira, unija in razlika. Kategorna izbira (oz. kategorni presek) je enaka preseku množic, kategorna unija je enaka uniji množic, kategorna razlika pa razliki množic. Za inducirana podobjekta $\underline{B} = (B, L_A, \sigma_A) \in \mathcal{P}(\underline{A})$ in $\underline{C} = (C, L_A, \sigma_A) \in \mathcal{P}(\underline{A})$ torej velja:

- Kategorni presek $\underline{B} \cap \underline{C}$ je objekt $(B \cap C, L_A, \sigma_A)$.
- Kategorna unija $\underline{B} \cup \underline{C}$ je objekt $(B \cup C, L_A, \sigma_A)$.
- Kategorna razlika $\underline{B} - \underline{C}$ je objekt $(B - C, L_A, \sigma_A)$.

Kategorni presek in kategorna unija v $\mathcal{P}(\underline{A})$ sta tudi največja spodnja in najmanjša zgornja meja v $\text{RegSub}(\underline{A})$, vprašanje relativnega komplementa pa se zdi v $\text{RegSub}(\underline{A})$ precej težavno.

Videli smo, da imajo delno urejene množice $\mathcal{P}(\underline{A})$ objektov kategorije podobnosti **SimObj** vse kategorne konstrukte za modeliranje relacijske algebre. Bodimo pozorni na dejstvo, da polne mreže, ki nastopajo v definiciji objektov kategorije podobnosti, niso pogoj za njihov obstoj. Dovolj bi bile že delno urejene množice z največjim elementom.

V nadaljevanju bomo predstavili dve podkategoriji kategorije podobnosti, in sicer *kategorijo urejenih množic* in *kategorijo kompaktnih metričnih prostorov*.

9.3.1 Kategorija urejenih množic

Definicija 9.34. Kategorijo urejenih množic **OrdSet** definirajmo takole:

1. Objekti naj bodo množice z izbrano refleksivno relacijo \triangleleft_A , tj. urejeni pari oblike

$$(A, \triangleleft_A),$$

kjer je A množica,

$$\triangleleft_A : A \times A \rightarrow \{0, 1\}$$

pa izbrana refleksivna relacija v množici A .

2. Morfizmi naj bodo preslikave množic, ki ohranjajo urejenost, tj. preslikave oblike

$$f : (A, \triangleleft_A) \rightarrow (B, \triangleleft_B),$$

da za $x, y \in A$ velja

$$x \triangleleft_A y \implies f(x) \triangleleft_B f(y).$$

Pokažimo, da je kategorija **OrdSet** podkategorija kategorije **SimObj**. Naj bo (A, \triangleleft_A) poljubni objekt kategorije urejenih množic **OrdSet**. Relaciji

$$\triangleleft_A : A \times A \rightarrow \{0, 1\}$$

pripada polna mreža $L_2 = (2; \min, \max)$ in za vsak $x \in A$ velja $\triangleleft_A(x, x) = 1$, kjer je 1 največji element mreže L_2 . To pa pomeni, da je urejena množica (A, \triangleleft_A) , zapisana kot $(A, L_2, \triangleleft_A)$, objekt kategorije podobnosti **SimObj**. Razred objektov kategorije **OrdSet** je torej podrazred razreda objektov kategorije **SimObj**. Naj bo

$$f : (A, \triangleleft_A) \rightarrow (B, \triangleleft_B)$$

poljubna preslikava urejenih množic, ki ohranja urejenost. Pogoji, da preslikava f ohranja urejenost, lahko zapišemo tudi kot

$$\triangleleft_A(x, y) \implies \triangleleft_B(f(x), f(y)).$$

Za identično preslikavo polnih mrež $1_{L_2} : L_2 \rightarrow L_2$ tedaj sledi

$$1_{L_2}(\triangleleft_A(x, y)) \leq_B \triangleleft_B(f(x), f(y)),$$

kar pomeni, da je $(f, 1_{L_2})$ morfizem kategorije **SimObj**. Razred morfizmov kategorije **OrdSet** je podrazred razreda morfizmov kategorije **SimObj**. Ker se kompozicija in enota za kompozicijo v kategoriji **OrdSet** ujemata s kompozicijo in enoto za kompozicijo v kategoriji **SimObj**, je kategorija **OrdSet** podkategorija kategorije **SimObj**.

Relacija \triangleleft_A , ki skupaj z množico A tvori objekt (A, \triangleleft_A) kategorije urejenih množic, ima (poleg refleksivnosti) lahko tudi druge lastnosti, kot sta simetričnost in tranzitivnost (primer 9.35).

Primer 9.35. *Primer objekta kategorije urejenih množic **OrdSet** je množica krajevnih delčkov informacij za Slovenijo skupaj z reflektivno in tranzitivno relacijo biti manj specifičen, predstavljeno s Hassejevim diagramom na sliki 2.2. Ker predstavlja relacija biti manj specifičen temelj relaksacije v dialog mIPC Kripkejevih modelih, uporabljenih za predstavitev znanja v drugem eksperimentu Čarovnik iz Oza, sposobnost relaksacije podeduje tudi podatkovni model relacijske algebre, zasnovan v kategoriji urejenih množic.*

Ker želimo, da podatkovni model omogoča podajanje bližnjih, sorodnih informacij, poleg relacije, s katero smo uredili množice, potrebujemo še relacijo, ki bo urejala množice urejenih podmnožic. Naj bo torej (A, \triangleleft_A) poljubna urejena množica kategorije **OrdSet**. Naj bosta Q in R dve lastnosti, katerima ustrezata podmnožici $Q \subseteq A$ in $R \subseteq A$. Podmnožici Q in R je smiselno urediti z induciranimi relacijama relacije \triangleleft_A :

$$(Q, \triangleleft_A) \subseteq (A, \triangleleft_A)$$

$$(R, \triangleleft_A) \subseteq (A, \triangleleft_A)$$

Relacija \triangleleft_A porodi v množici

$$\mathcal{P}(A, \triangleleft_A) \subseteq \text{RegSub}(A, \triangleleft_A),$$

tj. množici vseh urejenih podmnožic urejene množice (A, \triangleleft_A) , ki so oblike $(Q \subseteq A, \triangleleft_A)$, relacijo \triangleleft , ki jo določa naslednja zveza:

$$(Q, \triangleleft_A) \triangleleft (R, \triangleleft_A) \iff (\forall x \in Q \exists y \in R : x \triangleleft_A y) \text{ in } (\forall y \in R \exists x \in Q : x \triangleleft_A y)$$

Urejenost induciranih podobjektov, ki jo določa relacija \triangleleft , se imenuje *Egli-Milnerjeva urejenost* [Plotkin-1983, Abramsky-1994]. Na pomembnost te urejenosti v podatkovnih zbirkah so opozorili tudi Buneman, Jung in Ohori [Buneman-1991].

Bodimo pozorni, da prazna in poljubna neprazna urejena podmnožica iz $\mathcal{P}(A, \triangleleft_A)$ nista v relaciji \triangleleft . To je zelo koristna lastnost kategorije urejenih množic, ko razmišljamo o sodelujočih podatkovnih modelih, v katerih prazen odgovor ni zaželen. Relacija \triangleleft je reflektivna, od relacije \triangleleft_A pa podeduje še naslednje lastnosti: tranzitivnost, simetričnost, antisimetričnost in asimetričnost.

9.3.2 Kategorija kompaktnih metričnih prostorov

Pogosto imamo opravka z vrednostnimi množicami, kot je npr. množica krajev, ko relacija, ki dva elementa poveže ali ne, vrednostne množice ne uredi dovolj dobro. Velikokrat nas zanima več kot le to, da sta dva elementa sorodna. Pomembna postane npr. razdalja med kraji, razdalja med časovnimi trenutki ali stopnja sorodnosti oz. zamenljivosti informacij. Merjenje in predstavitev razdalj omogoča podkategorija kategorije podobnosti, imenovana *kategorija kompaktnih metričnih prostorov*.

Definicija 9.36. Kategorijo kompaktnih metričnih prostorov **CompMet** definirajmo takole:

1. Objekti naj bodo kompaktni metrični prostori (A, d_A) , kjer je

$$d_A : A \times A \rightarrow [0, \infty]$$

metrika na kompaktni množici A . Ker je d_A metrika, izpolnjuje naslednje pogoje ($x, y, z \in A$):

- nenegativnost: $d_A(x, y) \geq 0$ in $d_A(x, y) = 0 \Leftrightarrow x = y$
- simetričnost: $d_A(x, y) = d_A(y, x)$
- trikotniško neenakost: $d_A(x, z) \leq d_A(x, y) + d_A(y, z)$

2. Morfizmi naj bodo Lipschitzove preslikave kompaktnih metričnih prostorov. To je, preslikave oblike

$$f : (A, d_A) \rightarrow (B, d_B),$$

za katere obstaja pozitivna realna konstanta $0 < K < \infty$, da za $x, y \in A$ velja

$$d_B(f(x), f(y)) \leq K d_A(x, y).$$

Najmanjšo konstanto K , ki izpolnjuje dani pogoj, imenujemo Lipschitzova konstanta.

Metrika je običajno definirana kot preslikava v množico nenegativnih realnih števil $[0, \infty)$. Definicija 9.36 dovoljuje še vrednost ∞ , ki pomeni nepremostljivo razdaljo neprimerljivih, nezamenljivih oz. nepodobnih elementov.

Ker so množice, ki nastopajo v podatkovnih zbirkah, končne in zato kompaktno, zahteva po kompaktnosti metričnih prostorov, tj. objektov kategorije kompaktnih metričnih prostorov **CompMet**, ne predstavlja omejitve podatkovnega modela, ki temelji na kategoriji **CompMet**.

Lipschitzove preslikave, ki so dobile ime po nemškem matematiku Rudolfu Lipschitzu, so zvezne. Vsi morfizmi kategorije kompaktnih metričnih prostorov **CompMet** so zato zvezne preslikave kompaktnih metričnih prostorov. Zvezne preslikave kompaktnih metričnih prostorov pa so enakomerno zvezne [Vrabec-1993]. Lipschitzovo preslikavo z Lipschitzovo konstanto $K = 1$ imenujemo *izometrija*, Lipschitzovo preslikavo s $K < 1$ pa *skrčitev*.

Včasih pa se znajdemo v situaciji, ko želimo poudariti popolno zamenljivost določenih parov podatkov, kot sta npr. letališče Orly in letališče Charles De Gaulle v Parizu (primer 8.11). To lahko storimo tako, da razdaljo med njima postavimo na 0. V tem primeru razdalja ne bo metrika, ker ne bo izpolnjevala drugega dela pogoja o nenegativnosti. Dobljena kategorija, ki se bo od kategorije kompaktnih metričnih prostorov nekoliko razlikovala, bo včasih bolj uporabna za modeliranje podatkovnih zbirk kot kategorija kompaktnih metričnih prostorov, ki zadošča strogim matematičnim pogojem.

Pokažimo, da je tudi kategorija **CompMet** podkategorija kategorije **SimObj**. Naj bo (A, d_A) poljuben objekt kategorije kompaktnih metričnih prostorov **CompMet**. Metriki

$$d_A : A \times A \rightarrow [0, \infty]$$

pripada polna mreža $L_{[0,\infty]} = ([0, \infty]; \max, \min)$, urejena z relacijo \geq , in za vsak $x \in A$ velja $d_A(x, x) = 0$. Najmanjši element mreže $L_{[0,\infty]}$ je ∞ , največji pa 0. Kompakten metrični prostor (A, d_A) , zapisan kot $(A, L_{[0,\infty]}, d_A)$, je tedaj objekt kategorije podobnosti **SimObj**. Razred objektov kategorije **CompMet** je torej podrazred razreda objektov kategorije **SimObj**. Naj bo

$$f : (A, d_A) \rightarrow (B, d_B)$$

Lipschitzova preslikava kompaktnih metričnih prostorov z Lipschitzovo konstanto K . Ker za poljubna $x, y \in A$ velja

$$d_B(f(x), f(y)) \leq K d_A(x, y),$$

za homomorfizem polnih mrež $\varphi(x) = Kx$ sledi

$$\varphi(d_A(x, y)) \geq d_B(f(x), f(y)),$$

kar pomeni, da je (f, φ) morfizem kategorije **SimObj**. Razred morfizmov kategorije **CompMet** je podrazred razreda morfizmov kategorije **SimObj**. Ker se kompozicija in enota za kompozicijo v kategoriji **CompMet** ujemata s kompozicijo in enoto za kompozicijo v kategoriji **SimObj**, je kategorija **CompMet** podkategorija kategorije **SimObj**.

Poiščimo še metriko, ki ureja kompaktne metrične podprostore danih kompaktnih metričnih prostorov kategorije **CompMet**. Vzemimo torej poljuben kompakten metrični prostor (A, d_A) kategorije **CompMet**. Naj bosta \mathcal{Q} in \mathcal{R} dve lastnosti, katerima ustrezata neprazni kompaktni podmnožici $Q \subseteq A$ in $R \subseteq A$. Pripadajoča podobjekta z inducirano metriko sta kompaktna metrična podprostora:

$$(Q, d_A) \subseteq (A, d_A)$$

$$(R, d_A) \subseteq (A, d_A)$$

Metrika d_A porodi v množici

$$\mathcal{P}_\emptyset(A, d_A) \subseteq \text{RegSub}_\emptyset(A, d_A)$$

nepraznih kompaktnih metričnih podprostorov prostora (A, d_A) , ki so oblike $(Q \subseteq A, d_A)$, naslednjo metriko:

$$d((Q, d_A), (R, d_A)) = \max\left\{ \sup_{x \in Q} \inf_{y \in R} \{ d_A(x, y) \}, \sup_{y \in R} \inf_{x \in Q} \{ d_A(x, y) \} \right\}$$

Metrika d se imenuje *Hausdorffova metrika* [Braun-2003]. Definirana je na nepraznih kompaktnih metričnih podprostorih. Če bi dovolili tudi prazne množice oz. prazne metrične prostore, dobljena preslikava ne bi bila metrika, saj bi (ob upoštevanju $\inf \emptyset = \infty$ in $\sup \emptyset = 0$) za $Q \neq \emptyset$ in $R \neq \emptyset$ veljalo

$$d(Q, \mathbf{0}) = 0$$

$$d(\mathbf{0}, R) = 0$$

kar bi bilo v nasprotju s drugim delom pogoja o nenegativnosti (definicija 9.36), ki mu metrika zadošča. Hausdorffovo metriko je kot mero razdalje v začetku 20. stoletja uvedel Felix Hausdorff. Danes se uporablja na številnih področjih teoretične in uporabne matematike, med drugim pri geometriji fraktalov [Falconer-1990], numerični matematiki [Sendov-2001] in obdelavi slik [Ginchev-1998, Takacs-1998].

Objekti kategorije urejenih množic **OrdSet** (definicija 9.34) so urejeni z reflektivno relacijo. Če gledamo na objekte kot na tabele, na podobjekte, ki predstavljajo poizvedbe, pa kot na podtabele, to pomeni, da sta poljubni dve vrstici tabele ali podobni (tj. v relaciji) ali nista podobni (tj. nista v relaciji). Podobno so objekti kategorije kompaktnih metričnih prostorov **CompMet** (definicija 9.36) urejeni z metriko. Za poljubni dve vrstici tabele torej obstaja vrednost, ki pove, kako zelo sta si podobni oz. kolikšno škodo utrpimo, če eno nadomestimo z drugo.

9.4 Zmogljivost kategornega modela relacijskih podatkovnih zbirk

Videli bomo, da je podatkovne zbirke, modelirane v kategoriji podobnosti **SimObj**, mogoče udejanjiti kot zbirko relacijskih tabel. Zmogljivost kategornega modela relacijskih podatkovnih zbirk bomo primerjali z zmogljivostjo dialog mIPC Kripkejevih modelov iz razdelka 2.5 in zbirko vremenskih podatkov, predhodno modeliranih z dialog mIPC Kripkejevim modelom, predstavili v kategoriji urejenih množic **OrdSet**.

9.4.1 Implementacija kategornih relacijskih podatkovnih zbirk

Kako udejanjiti podatkovno zbirko, ki je modelirana z relacijsko algebro na kategoriji podobnosti **SimObj**? Relacijske podatkovne zbirke, ki temeljijo na relacijski algebri, definirani v kategoriji množic **Set**, udejanjimo kot zbirko tabel. Podobno bi radi udejanjili tudi podatkovne zbirke, ki temeljijo na kategorni posplošitvi relacijske algebre, saj bi se v nasprotnem primeru od našega cilja – zgraditi zmogljivejši podatkovni model in hkrati čim bolj ohraniti obliko priljubljenih relacijskih podatkovnih zbirk – precej oddaljili. Izkaže se, da lahko tudi posplošene relacijske podatkovne zbirke, ki temeljijo na kategorno posplošeni relacijski algebri, definirani v kategoriji podobnosti, udejanjimo kot zbirko relacijskih tabel.

Vzemimo torej neko podkategorijo \mathcal{S} kategorije podobnosti **SimObj**. Naj bo (A, L_A, σ_A) poljuben objekt podkategorije \mathcal{S} . Podobnost

$$\sigma_A : A \times A \rightarrow L_A$$

lahko predstavimo kot relacijo

$$\sigma_A \subseteq A \times A \times L_A,$$

ki jo udejanjimo kot relacijsko tabelo $T[A]$. Objektu (A, L_A, σ_A) tako pripadata dve relacijski tabeli, in sicer tabela množice A in tabela $T[A]$. Podobjektu

$$(B, L_A, \sigma_A) \in \mathcal{P}(A, L_A, \sigma_A)$$

pa poleg tabele množice $B \subseteq A$ pripada tudi podtabela $T[B]$ tabele $T[A]$, ki se nanaša na podmnožico $B \subseteq A$ in inducirano mero podobnosti.

Primer 9.37 ilustrira implementacijo podkategorije kategorije podobnosti **SimObj**, ki opisuje del železniških povezav z Ljubljano.

Primer 9.37. *Naj bodo*

$$\begin{aligned} \underline{OZNAKA} &= (OZNAKA, L_1, \triangleleft_{OZNAKA}) \\ \underline{ODHOD} &= (ODHOD, L_{[0,\infty]}, d_{ODHOD}) \\ \underline{PRIHOD} &= (PRIHOD, L_{[0,\infty]}, d_{PRIHOD}) \\ \underline{\check{C}AS} &= (\check{C}AS, L_2, \triangleleft_{\check{C}AS}) \\ \underline{VLAK} &= (VLAK, L_1 \times L_{[0,\infty]} \times L_{[0,\infty]} \times L_2, d_{VLAKI}) \end{aligned}$$

izbrani objekti kategorije podobnosti **SimObj**, na katerih temelji podkategorija **Vlaki** \subseteq **SimObj**. Pri tem so:

$$\begin{aligned} OZNAKA &= \{ic247, ic518, ics12, ec51\} \\ ODHOD &= \{ljubljana\} \\ PRIHOD &= \{murska_sobota, ormož, pragersko\} \\ \check{C}AS &= \{7:40, 11:00, 15:00, 15:45, zjutraj, dopoldan, popoldan, zvečer\} \\ VLAK &= \{vlakIC247 = (ic247, ljubljana, murska_sobota, 7:40), \\ &\quad vlakIC518 = (ic518, ljubljana, murska_sobota, 11:00), \\ &\quad vlakICS12 = (ics12, ljubljana, pragersko, 15:00), \\ &\quad vlakEC51 = (ec51, ljubljana, ormož, 15:45)\} \\ &\subseteq OZNAKA \times ODHOD \times PRIHOD \times \check{C}AS \end{aligned}$$

$$\begin{aligned} \triangleleft_{OZNAKA}(ic247, ic247) &= 1 \\ \triangleleft_{OZNAKA}(ic247, ic518) &= 1 \\ \triangleleft_{OZNAKA}(ic247, ics12) &= 1 \\ \triangleleft_{OZNAKA}(ic247, ec51) &= 1 \\ \triangleleft_{OZNAKA}(ic518, ic247) &= 1 \\ \triangleleft_{OZNAKA}(ic518, ic518) &= 1 \\ \triangleleft_{OZNAKA}(ic518, ics12) &= 1 \\ \triangleleft_{OZNAKA}(ic518, ec51) &= 1 \\ \triangleleft_{OZNAKA}(ics12, ic247) &= 1 \\ \triangleleft_{OZNAKA}(ics12, ic518) &= 1 \\ \triangleleft_{OZNAKA}(ics12, ics12) &= 1 \\ \triangleleft_{OZNAKA}(ics12, ec51) &= 1 \\ \triangleleft_{OZNAKA}(ec51, ic247) &= 1 \\ \triangleleft_{OZNAKA}(ec51, ic518) &= 1 \\ \triangleleft_{OZNAKA}(ec51, ics12) &= 1 \\ \triangleleft_{OZNAKA}(ec51, ec51) &= 1 \end{aligned}$$

$$d_{ODHOD}(ljubljana, ljubljana) = 0$$

$$d_{PRIHOD}(murska_sobota, murska_sobota) = 0$$

$$d_{PRIHOD}(murska_sobota, ormož) = 38$$

$$d_{PRIHOD}(murska_sobota, pragersko) = 75$$

$$d_{PRIHOD}(ormož, murska_sobota) = 38$$

$$d_{PRIHOD}(ormož, ormož) = 0$$

$$d_{PRIHOD}(ormož, pragersko) = 37$$

$$d_{PRIHOD}(pragersko, murska_sobota) = 75$$

$$d_{PRIHOD}(pragersko, ormož) = 37$$

$$d_{PRIHOD}(pragersko, pragersko) = 0$$

$$\triangleleft_{\check{C}AS}(7:40, 7:40) = 1$$

$$\triangleleft_{\check{C}AS}(7:40, 11:00) = 0$$

$$\triangleleft_{\check{C}AS}(7:40, 15:00) = 0$$

$$\triangleleft_{\check{C}AS}(7:40, 15:45) = 0$$

$$\triangleleft_{\check{C}AS}(7:40, zjutraj) = 1$$

$$\triangleleft_{\check{C}AS}(7:40, dopoldan) = 1$$

$$\triangleleft_{\check{C}AS}(7:40, popoldan) = 0$$

$$\triangleleft_{\check{C}AS}(7:40, zvečer) = 0$$

$$\triangleleft_{\check{C}AS}(11:00, 7:40) = 0$$

$$\triangleleft_{\check{C}AS}(11:00, 11:00) = 1$$

$$\triangleleft_{\check{C}AS}(11:00, 15:00) = 0$$

$$\triangleleft_{\check{C}AS}(11:00, 15:45) = 0$$

$$\triangleleft_{\check{C}AS}(11:00, zjutraj) = 0$$

$$\triangleleft_{\check{C}AS}(11:00, dopoldan) = 1$$

$$\triangleleft_{\check{C}AS}(11:00, popoldan) = 0$$

$$\triangleleft_{\check{C}AS}(11:00, zvečer) = 0$$

$$\triangleleft_{\check{C}AS}(15:00, 7:40) = 0$$

$$\triangleleft_{\check{C}AS}(15:00, 11:00) = 0$$

$$\triangleleft_{\check{C}AS}(15:00, 15:00) = 1$$

$$\triangleleft_{\check{C}AS}(15:00, 15:45) = 1$$

$$\triangleleft_{\check{C}AS}(15:00, zjutraj) = 0$$

$$\triangleleft_{\check{C}AS}(15:00, dopoldan) = 0$$

$$\triangleleft_{\check{C}AS}(15:00, popoldan) = 1$$

$$\triangleleft_{\check{C}AS}(15:00, zvečer) = 0$$

$$\triangleleft_{\check{C}AS}(15:45, 7:40) = 0$$

$$\triangleleft_{\check{C}AS}(15:45, 11:00) = 0$$

$$\triangleleft_{\check{C}AS}(15:45, 15:00) = 1$$

$$\triangleleft_{\check{C}AS}(15:45, 15:45) = 1$$

$$\triangleleft_{\check{C}AS}(15:45, zjutraj) = 0$$

$$\triangleleft_{\check{C}AS}(15:45, dopoldan) = 0$$

$$\triangleleft_{\check{C}AS}(15:45, popoldan) = 1$$

$$\triangleleft_{\check{C}AS}(15:45, zvečer) = 0$$

$$\triangleleft_{\check{C}AS}(zjutraj, 7:40) = 1$$

$$\triangleleft_{\check{C}AS}(zjutraj, 11:00) = 0$$

$$\triangleleft_{\check{C}AS}(zjutraj, 15:00) = 0$$

$$\triangleleft_{\check{C}AS}(zjutraj, 15:45) = 0$$

$$\triangleleft_{\check{C}AS}(zjutraj, zjutraj) = 1$$

$$\triangleleft_{\check{C}AS}(zjutraj, dopoldan) = 0$$

$$\triangleleft_{\check{C}AS}(zjutraj, popoldan) = 0$$

$$\triangleleft_{\check{C}AS}(zjutraj, zvečer) = 0$$

$$\triangleleft_{\check{C}AS}(dopoldan, 7:40) = 1$$

$$\triangleleft_{\check{C}AS}(dopoldan, 11:00) = 1$$

$$\triangleleft_{\check{C}AS}(dopoldan, 15:00) = 0$$

$$\triangleleft_{\check{C}AS}(dopoldan, 15:45) = 0$$

$$\triangleleft_{\check{C}AS}(dopoldan, zjutraj) = 0$$

$$\triangleleft_{\check{C}AS}(dopoldan, dopoldan) = 1$$

$$\triangleleft_{\check{C}AS}(dopoldan, popoldan) = 0$$

$$\triangleleft_{\check{C}AS}(dopoldan, zvečer) = 0$$

$$\triangleleft_{\check{C}AS}(popoldan, 7:40) = 0$$

$$\triangleleft_{\check{C}AS}(popoldan, 11:00) = 0$$

$$\triangleleft_{\check{C}AS}(popoldan, 15:00) = 1$$

$$\triangleleft_{\check{C}AS}(popoldan, 15:45) = 1$$

$$\triangleleft_{\check{C}AS}(popoldan, zjutraj) = 0$$

$$\triangleleft_{\check{C}AS}(zvečer, 7:40) = 0$$

$$\triangleleft_{\check{C}AS}(zvečer, 11:00) = 0$$

$$\triangleleft_{\check{C}AS}(zvečer, 15:00) = 0$$

$$\triangleleft_{\check{C}AS}(zvečer, 15:45) = 1$$

$$\triangleleft_{\check{C}AS}(zvečer, zjutraj) = 0$$

$$\begin{aligned}
\triangleleft_{\check{C}AS}(\text{popoldan}, \text{dopoldan}) &= 0 & \triangleleft_{\check{C}AS}(\text{zvečer}, \text{dopoldan}) &= 0 \\
\triangleleft_{\check{C}AS}(\text{popoldan}, \text{popoldan}) &= 1 & \triangleleft_{\check{C}AS}(\text{zvečer}, \text{popoldan}) &= 0 \\
\triangleleft_{\check{C}AS}(\text{popoldan}, \text{zvečer}) &= 0 & \triangleleft_{\check{C}AS}(\text{zvečer}, \text{zvečer}) &= 1
\end{aligned}$$

$$d_{VLAKE}((x, y, z, u), (x', y', z', u')) = (\triangleleft_{OZNAKA}(x, x'), d_{ODHOD}(y, y'), d_{PRIHOD}(z, z'), \triangleleft_{\check{C}AS}(u, u'))$$

Dejstvo, da relacija \triangleleft_{OZNAKA} vsem parom iz množice $OZNAKA$ pripiše podobnost 1, pomeni, da je vrsta vlaka (LP – potniški vlak, RG – regionalni vlak, IC – InterCity vlak, ICS – InterCity Slovenija, EC – EuroCity vlak) za snovalce danega modela povsem nepomembna. Metriki d_{ODHOD} in d_{PRIHOD} podajata razdaljo med danimi pari krajev, $\triangleleft_{\check{C}AS}$ pa opisuje podobnost oz. relevantnost časovnih trenutkov iz množice $\check{C}AS$.

Vsakemu izmed naštetih objektov pripadata dve relacijski tabeli:

<u>OZNAKA</u>	<u>OZNAKA</u>	<u>OZNAKA'</u>	<u>\triangleleft_{OZNAKA}</u>
ic247	ic247	ic247	1
ic518	ic247	ic518	1
ics12	ic247	ics12	1
ec51	ic247	ec51	1
	ic518	ic247	1
	ic518	ic518	1
	ic518	ics12	1
	ic518	ec51	1
	ics12	ic247	1
	ics12	ic518	1
	ics12	ics12	1
	ics12	ec51	1
	ec51	ic247	1
	ec51	ic518	1
	ec51	ics12	1
	ec51	ec51	1

OZNAKA

<u>ODHOD</u>	<u>ODHOD</u>	<u>ODHOD'</u>	<u>d_{ODHOD}</u>
ljubljana	ljubljana	ljubljana	0

ODHOD

<u>PRIHOD</u>	<u>PRIHOD</u>	<u>PRIHOD'</u>	d_{PRIHOD}
<i>murska_sobota</i>	<i>murska_sobota</i>	<i>murska_sobota</i>	0
<i>ormož</i>	<i>murska_sobota</i>	<i>ormož</i>	38
<i>pragersko</i>	<i>murska_sobota</i>	<i>pragersko</i>	75
	<i>ormož</i>	<i>murska_sobota</i>	38
	<i>ormož</i>	<i>ormož</i>	0
	<i>ormož</i>	<i>pragersko</i>	37
	<i>pragersko</i>	<i>murska_sobota</i>	75
	<i>pragersko</i>	<i>ormož</i>	37
	<i>pragersko</i>	<i>pragersko</i>	0

PRIHOD

<u>ČAS</u>	<u>ČAS</u>	<u>ČAS'</u>	$\triangleleft_{\check{C}AS}$
<i>7:40</i>	<i>7:40</i>	<i>7:40</i>	1
<i>11:00</i>	<i>7:40</i>	<i>11:00</i>	0
<i>15:00</i>	<i>7:40</i>	<i>15:00</i>	0
<i>15:45</i>	<i>7:40</i>	<i>15:45</i>	0
<i>zjutraj</i>	<i>7:40</i>	<i>zjutraj</i>	0
<i>dopoldan</i>	<i>7:40</i>	<i>dopoldan</i>	1
<i>popoldan</i>	<i>7:40</i>	<i>popoldan</i>	0
<i>zvečer</i>	<i>7:40</i>	<i>zvečer</i>	0
	<i>11:00</i>	<i>7:40</i>	0
	<i>11:00</i>	<i>11:00</i>	1
	<i>11:00</i>	<i>15:00</i>	0
	<i>11:00</i>	<i>15:45</i>	0
	<i>11:00</i>	<i>zjutraj</i>	0
	<i>11:00</i>	<i>dopoldan</i>	1
	<i>11:00</i>	<i>popoldan</i>	0
	<i>11:00</i>	<i>zvečer</i>	0
	<i>15:00</i>	<i>7:40</i>	0
	<i>15:00</i>	<i>11:00</i>	0
	<i>15:00</i>	<i>15:00</i>	1
	<i>15:00</i>	<i>15:45</i>	1
	<i>15:00</i>	<i>zjutraj</i>	0
	<i>15:00</i>	<i>dopoldan</i>	0
	<i>15:00</i>	<i>popoldan</i>	1
	<i>15:00</i>	<i>zvečer</i>	0
	<i>15:45</i>	<i>7:40</i>	0
	<i>15:45</i>	<i>11:00</i>	0
	<i>15:45</i>	<i>15:00</i>	1
	<i>15:45</i>	<i>15:45</i>	1
	<i>15:45</i>	<i>zjutraj</i>	0

15:45	dopoldan	0
15:45	popoldan	1
15:45	zvečer	0
zjutraj	7:40	1
zjutraj	11:00	0
zjutraj	15:00	0
zjutraj	15:45	0
zjutraj	zjutraj	1
zjutraj	dopoldan	0
zjutraj	popoldan	0
zjutraj	zvečer	0
dopoldan	7:40	1
dopoldan	11:00	1
dopoldan	15:00	0
dopoldan	15:45	0
dopoldan	zjutraj	0
dopoldan	dopoldan	1
dopoldan	popoldan	0
dopoldan	zvečer	0
popoldan	7:40	0
popoldan	11:00	0
popoldan	15:00	1
popoldan	15:45	1
popoldan	zjutraj	0
popoldan	dopoldan	0
popoldan	popoldan	1
popoldan	zvečer	0
zvečer	7:40	0
zvečer	11:00	0
zvečer	15:00	0
zvečer	15:45	0
zvečer	zjutraj	0
zvečer	dopoldan	0
zvečer	popoldan	0
zvečer	zvečer	1

ČAS

<u>VLAK</u>	<u>VLAK</u>	<u>VLAK'</u>	<u>d_{VLAK}</u>
<i>vlakIC247</i>	<i>vlakIC247</i>	<i>vlakIC247</i>	$(1, 0, 0, 1)$
<i>vlakIC518</i>	<i>vlakIC247</i>	<i>vlakIC518</i>	$(1, 0, 0, 0)$
<i>vlakICS12</i>	<i>vlakIC247</i>	<i>vlakICS12</i>	$(1, 0, 75, 0)$
<i>vlakEC51</i>	<i>vlakIC247</i>	<i>vlakEC51</i>	$(1, 0, 38, 0)$
	<i>vlakIC518</i>	<i>vlakIC247</i>	$(1, 0, 0, 0)$

<i>vlakIC518</i>	<i>vlakIC518</i>	$(1, 0, 0, 1)$
<i>vlakIC518</i>	<i>vlakICS12</i>	$(1, 0, 75, 0)$
<i>vlakIC518</i>	<i>vlakEC51</i>	$(1, 0, 38, 0)$
<i>vlakICS12</i>	<i>vlakIC247</i>	$(1, 0, 75, 0)$
<i>vlakICS12</i>	<i>vlakIC518</i>	$(1, 0, 75, 0)$
<i>vlakICS12</i>	<i>vlakICS12</i>	$(1, 0, 0, 1)$
<i>vlakICS12</i>	<i>vlakEC51</i>	$(1, 0, 37, 1)$
<i>vlakEC51</i>	<i>vlakIC247</i>	$(1, 0, 38, 0)$
<i>vlakEC51</i>	<i>vlakIC518</i>	$(1, 0, 38, 0)$
<i>vlakEC51</i>	<i>vlakICS12</i>	$(1, 0, 37, 1)$
<i>vlakEC51</i>	<i>vlakEC51</i>	$(1, 0, 0, 1)$

VLAK

Relaciji *VLAK*, ki je podmnožica relacije $OZNAKA \times ODHOD \times PRIHOD \times \check{C}AS$, pripada ključna, informativno najpomembnejša tabela:

<i>OZNAKA</i>	<i>ODHOD</i>	<i>PRIHOD</i>	<i>ČAS</i>
<i>ic247</i>	<i>ljubljana</i>	<i>murska_sobota</i>	7:40
<i>ic518</i>	<i>ljubljana</i>	<i>murska_sobota</i>	11:00
<i>ics12</i>	<i>ljubljana</i>	<i>pragersko</i>	15:00
<i>ec51</i>	<i>ljubljana</i>	<i>ormož</i>	15:45

Vzemimo poizvedbo

“Ob kateri uri popoldne pelje vlak iz Ljubljane v kraj, ki od Murske Sobote ni oddaljen več kot 50 km?”

Ta se v jeziku relacijske algebre glasi takole:

$$\pi_{1,3,4}(\sigma_{\$2=ljubljana}(VLAK)) \bowtie \pi_2(\sigma_{\$1=murska_sobota \wedge \$3 \leq 50}(T[PRIHOD])) \bowtie \pi_2(\sigma_{\$1=popoldan \wedge \$3=1}(T[\check{C}AS]))$$

Odgovor na poizvedbo je podtabela

<i>OZNAKA</i>	<i>ODHOD</i>	<i>PRIHOD</i>	<i>ČAS</i>
<i>ec51</i>	<i>ljubljana</i>	<i>ormož</i>	15:45

ki pripada regularnemu podobjektu $(\{ec51, ljubljana, ormož, 15:45\}, L_1 \times L_{[0,\infty]} \times L_{[0,\infty]} \times L_2, d_{VLAKI})$ objekta *VLAK* podkategorije **Vlaki** \subseteq **SimObj**.

9.4.2 Relaksacija

Kategorija urejenih množic in kategorija kompaktnih metričnih prostorov omogočata relaksacijo poizvedb oz. posploševanje odgovorov. Relaksacija temelji na reflektivni Egli-Milnerjevi relaciji \triangleleft (razdelek 9.3.1), definirani na podobjektivih kategorije **OrdSet**, oz. Hausdorffovi metriki d (razdelek 9.3.2), definirani na nepraznih podobjektivih kategorije **CompMet**.

Če je inducirana podmnožica $(R, \triangleleft_A) \in \mathcal{P}(A, \triangleleft_A)$ urejene množice (A, \triangleleft_A) odgovor na poizvedbo Q , je relaksiran odgovor vsaka inducirana podmnožica $(P, \triangleleft_A) \in \mathcal{P}(A, \triangleleft_A)$, za katero velja:

$$(R, \triangleleft_A) \triangleleft (P, \triangleleft_A)$$

Do takih odgovorov pridemo posredno z uporabo relaksacije poizvedbe Q , opisane v razdelku 8.3.5, ali direktno s posploševanjem eksaktnega odgovora (R, \triangleleft_A) , ki temelji na relaciji \triangleleft . Odločitev o tem, katerega izmed relaksiranih odgovorov bomo podali, je odvisna od obsega posameznih odgovorov in zahtev uporabnika. V naslednjih korakih lahko s posploševanjem odgovorov nadaljujemo, dokler ne pridemo do sodelujočega odgovora, ki uporabnika zadovolji.

Če je neprazen kompakten metrični podprostor $(R, d_A) \in \mathcal{P}_\emptyset(A, d_A)$ kompaktnega metričnega prostora (A, d_A) odgovor na poizvedbo Q , je relaksiran odgovor vsak neprazen kompakten metrični podprostor $(P, d_A) \in \mathcal{P}_\emptyset(A, d_A)$, ki je od (R, d_A) oddaljen manj od neke vnaprej predpisane vrednosti d_{max} :

$$d((P, d_A), (R, d_A)) < d_{max}$$

Običajno so relevantnejši tisti relaksirani odgovori, ki so od eksaktnega odgovora (R, d_A) najmanj oddaljeni.

Težnja po čim bolj naravni in zmoglivi relaksaciji, ki dvigne raven sodelujočega odgovarjanja podatkovnih sistemov, upraviči definicijo kategorije podobnosti **SimObj**, ki združuje oba načina urejenosti, tj. urejenost z relacijo \triangleleft in urejenost z metriko d .

Ker prazen podobjekt in poljuben neprazen podobjekt objekta (A, \triangleleft_A) kategorije urejenih množic **OrdSet** nista v relaciji \triangleleft , je razdaljo med praznim podobjektom $(\emptyset, d_A) \in \mathcal{P}(A, d_A)$ in nepraznim podobjektom $(R, d_A) \in \mathcal{P}_\emptyset(A, d_A)$ kategorije kompaktnih metričnih prostorov **CompMet** smiselno dodefimirati kot

$$d((\emptyset, d_A), (R, d_A)) = d((R, d_A), (\emptyset, d_A)) = \infty,$$

kjer je ∞ najmanjši element polne mreže $L_{[0, \infty]}$. Meri podobnosti oz. različnosti \triangleleft in d , definirani v množici $\mathcal{P}(\underline{A}) \subseteq \text{RegSub}(\underline{A})$ dane kategorije, tedaj poenotimo s preslikavo

$$\sigma : \mathcal{P}(\underline{A}) \times \mathcal{P}(\underline{A}) \rightarrow (L; \wedge, \vee),$$

ki urejenim parom podobjektov iz $\mathcal{P}(\underline{A})$ kategorije podobnosti **SimObj** priredi vrednost iz polne mreže $(L; \wedge, \vee)$. Naj bosta torej

$$\underline{Q} = (Q, L_A, \sigma_A) \in \mathcal{P}(\underline{A})$$

$$\underline{R} = (R, L_A, \sigma_A) \in \mathcal{P}(\underline{A})$$

podobjekta objekta $\underline{A} = (A, L_A, \sigma_A)$ kategorije **SimObj**. Mera podobnosti σ_A porodi v množici $\mathcal{P}(\underline{A})$ mero podobnosti

$$\sigma(\underline{Q}, \underline{R}) = \left(\bigwedge_{x \in \underline{Q}} \bigvee_{y \in \underline{R}} \sigma_A(x, y) \right) \wedge \left(\bigwedge_{y \in \underline{R}} \bigvee_{x \in \underline{Q}} \sigma_A(x, y) \right),$$

pri čemer sta \wedge in \vee operaciji mreže L_A . Podobnost med najmanjšim elementom $\underline{\mathbf{0}} = (\emptyset, L_A, \sigma_A)$ delno urejene množice $\mathcal{P}(\underline{A})$ in poljubnim drugim podobjektom $\underline{R} \in \mathcal{P}(\underline{A})$ je enaka

$$\sigma(\underline{\mathbf{0}}, \underline{R}) = \sigma(\underline{R}, \underline{\mathbf{0}}) = 0_A,$$

kjer je 0_A najmanjši element polne mreže L_A . Z najmanjšim elementom 0_A preprečimo zadovoljstvo podatkovnega sistema ob praznem odgovoru. Omenimo še, da za vse podobjekte $\underline{Q} \in \mathcal{P}(\underline{A})$, torej tudi najmanjši element $\underline{\mathbf{0}}$, velja

$$\sigma(\underline{Q}, \underline{Q}) = 1_A.$$

Tako definirana mera podobnosti σ je posplošitev relacije \triangleleft in metrike d .

Če je induciran podobjekt $(R, L_A, \sigma_A) \in \mathcal{P}(A, L_A, \sigma_A)$ objekta (A, L_A, σ_A) kategorije podobnosti **SimObj** odgovor na poizvedbo Q , je relaksiran odgovor vsak induciran podobjekt $(\overline{P}, L_A, \sigma_A) \in \mathcal{P}(A, L_A, \sigma_A)$, ki je podobjektu (R, L_A, σ_A) glede na mero podobnosti σ 'dovolj podoben'. Ponavadi bodo relevantnejši tisti relaksirani odgovori, ki bodo eksaktnemu odgovoru (R, L_A, σ_A) najbolj podobni. Glej primer 9.38.

Primer 9.38. *Vzemimo podatkovno zbirko iz primera 9.37 in poizvedbo*

“Ali ob 15:00 pelje kakšen vlak iz Ljubljane v Mursko Soboto?”

Ta poizvedba je neuspešna. Če pogoj o uri odhoda vlaka z uporabo tabele $T[\check{C}AS]$ transformiramo (15:00 in popoldan sta v relaciji $\triangleleft_{\check{C}AS}$), dobimo npr. poizvedbo

“Ali popoldne pelje kakšen vlak iz Ljubljane v Mursko Soboto?”

Če pa pogoj o času odhoda izpustimo, dobimo poizvedbo

“Ali pelje kakšen vlak iz Ljubljane v Mursko Soboto?”

To sta primera poizvedb, dobljenih z relaksacijo na osnovi transformacije pogoja. Ker se eksakten odgovor na zadnjo poizvedbo glasi

“Da.”

je to poizvedbo ugodneje preoblikovati v relaksirano poizvedbo

“Kdaj pelje vlak iz Ljubljane v Mursko Soboto?”

ki je dobljena z relaksacijo na osnovi transformacije izbora atributov. Njej pripadajoč izraz relacijske algebre se glasi takole:

$$\pi_4(\sigma_{\$2=ljubljana\wedge\$3=murska_sobota}(VLAK))$$

Z izrazom relacijske algebre zapišimo še ‘popolno’ relaksacijo začetne poizvedbe v dani podatkovni zbirki, tj. relaksacijo, ki temelji na transformaciji vseh pogojev (časovne informacije, kraja odhoda in kraja prihoda):

$$VLAK \bowtie \pi_2(\sigma_{\$1=ljubljana\wedge\$3\leq d}(ODHOD)) \bowtie \pi_2(\sigma_{\$1=murska_sobota\wedge\$3\leq d}(PRIHOD)) \bowtie \pi_2(\sigma_{\$1=15:00\wedge\$3=1}(\check{C}AS))$$

Konstanta d označuje zgornjo mejo dovoljene oddaljenosti še relevantnih krajev. Pri tem bi za kraje odhoda in kraje prihoda lahko izbrali različni zgornji meji dovoljene oddaljenosti. V primeru enakih zgornjih mej dobimo za $38 \leq d \leq 74$ enovrstično podtabelo A_1 :

OZNAKA	ODHOD	PRIHOD	ČAS
ec51	ljubljana	ormož	15:45

Njej pripadajoč podobjekt objekta $\underline{A} = \underline{OZNAKA} \times \underline{ODHOD} \times \underline{PRIHOD} \times \underline{\check{C}AS}$ označimo z \underline{A}_1 . Za $d \geq 75$ pa dobimo dvovrstično podtabelo A_2 :

OZNAKA	ODHOD	PRIHOD	ČAS
ics12	ljubljana	pragersko	15:00
ec51	ljubljana	ormož	15:45

Njej pripadajoč podobjekt objekta \underline{A} označimo z \underline{A}_2 . Dobljeni podtabeli predstavljata relevantna odgovora na sicer neuspešno poizvedbo. Poglejmo, kako sta podtabeli A_1 in A_2 podobni teoretični podtabeli A_0 , ki izpolnjuje vse pogoje začetne poizvedbe:

OZNAKA	ODHOD	PRIHOD	ČAS
ic247	ljubljana	murska_sobota	15:00
ic518	ljubljana	murska_sobota	15:00
ics12	ljubljana	murska_sobota	15:00
ec51	ljubljana	murska_sobota	15:00

Podobnost med induciranimi podobjekti objekta \underline{A} meri preslikava

$$\sigma : \mathcal{P}(\underline{A}) \times \mathcal{P}(\underline{A}) \rightarrow L_1 \times L_{[0,\infty]} \times L_{[0,\infty]} \times L_2,$$

ki podobjektoma $\underline{A}_0 \in \mathcal{P}(\underline{A})$ in $\underline{A}_1 \in \mathcal{P}(\underline{A})$ pripiše vrednost:

$$\begin{aligned}
\sigma(\underline{A}_0, \underline{A}_1) &= (\sigma(A_0.OZNAKA, A_1.OZNAKA), \sigma(A_0.ODHOD, A_1.ODHOD), \\
&\quad \sigma(A_0.PRIHOD, A_1.PRIHOD), \sigma(A_0.ČAS, A_1.ČAS)) \\
&= (1, 0, \max\{ \sup_{x \in A_0.PRIHOD} \{ \inf_{y \in A_1.PRIHOD} \{ d_{PRIHOD}(x, y) \} \}, \\
&\quad \sup_{y \in A_1.PRIHOD} \{ \inf_{x \in A_0.PRIHOD} \{ d_{PRIHOD}(x, y) \} \} \}, 1) \\
&= (1, 0, 38, 1)
\end{aligned}$$

Za podobjekta $\underline{A}_0 \in \mathcal{P}(A)$ in $\underline{A}_2 \in \mathcal{P}(A)$ pa velja:

$$\begin{aligned}
\sigma(\underline{A}_0, \underline{A}_2) &= (\sigma(A_0.OZNAKA, A_2.OZNAKA), \sigma(A_0.ODHOD, A_2.ODHOD), \\
&\quad \sigma(A_0.PRIHOD, A_2.PRIHOD), \sigma(A_0.ČAS, A_2.ČAS)) \\
&= (1, 0, \max\{ \sup_{x \in A_0.PRIHOD} \{ \inf_{y \in A_2.PRIHOD} \{ d_{PRIHOD}(x, y) \} \}, \\
&\quad \sup_{y \in A_2.PRIHOD} \{ \inf_{x \in A_0.PRIHOD} \{ d_{PRIHOD}(x, y) \} \} \}, 1) \\
&= (1, 0, 75, 1)
\end{aligned}$$

Ker velja $(1, 0, 0, 1) \leq (1, 0, 38, 1) \leq (1, 0, 75, 1)$, kjer je \leq relacija delne urejenosti, $(1, 0, 0, 1)$ pa največji element mreže $L_1 \times L_{[0, \infty]} \times L_{[0, \infty]} \times L_2$, sta si objekta \underline{A}_0 in \underline{A}_1 bolj podobna kot objekta \underline{A}_0 in \underline{A}_2 .

Za mrežo L_A , definirano kot del objekta (A, L_A, σ_A) kategorije podobnosti **SimObj**, smo zahtevali, da je polna (definicija 9.33). Ureditve polne mreže L_A torej ni nujno linearna, kar pomeni, da lahko vsebuje pare elementov, ki niso primerljivi. To je zelo dobrodošla lastnost kategorije podobnosti, saj na relevantnost oz. bližino relaksiranega odgovora pogosto vpliva več dejavnikov, katerih pomembnost ni za vse uporabnike enaka. Glej primer 9.39.

Primer 9.39. Vzemimo objekt $(POTOVANJE, L_2 \times L_{[0, \infty]} \times L_{[0, \infty]}, \sigma_{POTOVANJE})$ s pripadajočima relacijskima tabelama $POTOVANJE$ in $T[POTOVANJE]$:

NAČIN	ODHOD	PRIHOD
letalo	ljubljana	berlin
vlak	ljubljana	hamburg
letalo	ljubljana	hamburg

POTOVANJE

NAČIN	ODHOD	PRIHOD	NAČIN	ODHOD	PRIHOD	$\sigma_{POTOVANJE}$
letalo	ljubljana	berlin	letalo	ljubljana	berlin	$(1, 0, 0)$
letalo	ljubljana	berlin	vlak	ljubljana	hamburg	$(0, 0, 292)$
letalo	ljubljana	berlin	letalo	ljubljana	hamburg	$(1, 0, 292)$
vlak	ljubljana	hamburg	letalo	ljubljana	berlin	$(0, 0, 292)$
vlak	ljubljana	hamburg	vlak	ljubljana	hamburg	$(1, 0, 0)$
vlak	ljubljana	hamburg	letalo	ljubljana	hamburg	$(0, 0, 0)$

letalo	ljubljana	hamburg		letalo	ljubljana	berlin		(1, 0, 292)
letalo	ljubljana	hamburg		vlak	ljubljana	hamburg		(0, 0, 0)
letalo	ljubljana	hamburg		letalo	ljubljana	hamburg		(1, 0, 0)

$T[POTOVANJE]$

Največji element mreže $L_2 \times L_{[0,\infty]} \times L_{[0,\infty]}$ je $(1, 0, 0)$. Iz tabele $T[POTOVANJE]$ tedaj razberemo, da vrednosti $(1, 0, 292)$ in $(0, 0, 0)$ iz mreže $L_2 \times L_{[0,\infty]} \times L_{[0,\infty]}$, ki pripadata elementoma $(letalo, ljubljana, berlin)$ in $(vlak, ljubljana, hamburg)$ glede na element $(letalo, ljubljana, hamburg)$ nista primerljiva. Neprimerljivost podatkov pa je pogosto zelo koristna lastnost podatkovnih sistemov. Razlaga je preprosta. Recimo, da želimo z letalom potovati v Hamburg. Če to ni možno, lahko pa izbiramo med potovanjem z vlakom in poletom v 292 km oddaljen Berlin, se znajdemo pred izbiro, ki je odvisna od situacije, osebnih želja in/ali izkušenj. Taka dilema torej ni univerzalno (tj. za vse uporabnike enako) rešljiva in je zato najbolje, da odločitev o najugodnejši rešitvi podatkovni sistem prepusti uporabniku.

9.4.3 Primerjava kategornega modela relacijskih podatkovnih zbirk in dialog mIPC Kripkejevih modelov

Vremenske podatke (slike 2.2, 2.3, 2.4 in 2.5), ki smo jih v okviru drugega eksperimenta Čarovnik iz Oza modelirali z dialog mIPC Kripkejevim modelom (razdelek 2.5), bomo predstavili kot podkategorijo kategorije podobnosti **SimObj** in to podkategorijo pretvorili v zbirko relacijskih tabel.

Refleksivna in tranzitivna relacija dosegljivosti R , definirana v množici svetov dialog mIPC Kripkejevega modela, je temeljila na vnaprej definirani refleksivni in tranzitivni relaciji *biti manj specifičen*, ki je povezovala manj specifične delčke informacij z bolj specifičnimi (razdelek 2.5). Če želimo na slikah 2.2, 2.3, 2.4 in 2.5 podane vremenske podatke predstaviti v kategoriji podobnosti **SimObj**, jim moramo prirediti ustrezno podkategorijo **Vreme** \subseteq **SimObj**. Zaradi narave mIPC Kripkejevih modelov (definicija 2.2) bo **Vreme** celo podkategorija kategorije urejenih množic **OrdSet**. Osnovni objekti podkategorije **Vreme** bodo štiri urejene množice:

- $\underline{SLO} = (SLO, \triangleleft_{SLO})$, kjer je SLO množica krajevnih delčkov informacij na sliki 2.2, ki se navezujejo na Slovenijo,

$$\triangleleft_{SLO} : SLO \times SLO \rightarrow L_2$$

pa refleksivna in tranzitivna relacija v množici SLO, ki ustreza delu relacije *biti manj specifičen*.

- $\underline{EUR} = (EUR, \triangleleft_{EUR})$, kjer je EUR množica krajevnih delčkov informacij na sliki 2.3, ki se navezujejo na Evropo (izvzeta je Slovenija),

$$\triangleleft_{EUR} : EUR \times EUR \rightarrow L_2$$

pa refleksivna in tranzitivna relacija v množici EUR, ki ustreza delu relacije *biti manj specifičen*.

- $\check{\text{CAS}} = (\check{\text{CAS}}, \triangleleft_{\check{\text{CAS}}})$, kjer je $\check{\text{CAS}}$ množica časovnih delčkov informacij na sliki 2.4,

$$\triangleleft_{\check{\text{CAS}}} : \check{\text{CAS}} \times \check{\text{CAS}} \rightarrow L_2$$

pa refleksivna in tranzitivna relacija v množici $\check{\text{CAS}}$, ki ustreza delu relacije *biti manj specifičen*. Spomnimo, da smo, zaradi majhnega števila možnih dosegljivih časovnih trenutkov, v množici $\check{\text{CAS}}$ vzpostavili vse povezave.

- $\text{PODATEK} = (\text{PODATEK}, \triangleleft_{\text{PODATEK}})$, kjer je PODATEK množica podatkovnih delčkov informacij na sliki 2.5,

$$\triangleleft_{\text{PODATEK}} : \text{PODATEK} \times \text{PODATEK} \rightarrow L_2$$

pa refleksivna in tranzitivna relacija v množici PODATEK , ki ustreza delu relacije *biti manj specifičen*.

Vsakemu izmed omenjenih objektov pripadata dve relacijski tabeli – ena tabela predstavlja množico, druga pa pripadajočo relacijo (razdelek 9.4.1). Objektu SLO pripadata tabeli SLO in $T[\text{SLO}]$, katere prva dva stolpca sestavljajo urejeni pari krajev iz množice SLO , tretjega pa pripadajoče vrednosti mere podobnosti $\triangleleft_{\text{SLO}}$, tj. 0, če kraja nista v relaciji *biti manj specifičen*, in 1, če sta v relaciji *biti manj specifičen*. Objektu EUR pripadata tabeli EUR in $T[\text{EUR}]$, katere prva dva stolpca sestavljajo urejeni pari krajev iz množice EUR , tretjega pa pripadajoče vrednosti mere podobnosti $\triangleleft_{\text{EUR}}$. Podobno, objektu $\check{\text{CAS}}$ pripadata tabeli $\check{\text{CAS}}$ in $T[\check{\text{CAS}}]$, objektu PODATEK pa tabeli PODATEK in $T[\text{PODATEK}]$.

Ker so našteje relacijske tabele zelo velike (tabeli SLO in $T[\text{SLO}]$ imata 65 oz. $\binom{65}{2} = 2080$ vrstic, tabeli EUR in $T[\text{EUR}]$ imata 108 oz. $\binom{108}{2} = 5778$ vrstic, tabeli $\check{\text{CAS}}$ in $T[\check{\text{CAS}}]$ imata 57 oz. $\binom{57}{2} = 1596$ vrstic, tabeli PODATEK in $T[\text{PODATEK}]$ pa imata 71 oz. $\binom{71}{2} = 2485$ vrstic), jih ne bomo navajali.

V razdelku 8.4 smo povedali, da dialog mIPC Kripkejevi modeli podpirajo naslednje tehnike sodelujočega odgovarjanja:

- ↪ vrednotenje domnev, vsebovanih v poizvedbah,
- ↪ odkrivanje in odpravljanje nesporazumov,
- ↪ oblikovanje intenzionalnih odgovorov ter
- ↪ posploševanje poizvedb in odgovorov.

Ker smo kategorni model relacijskih podatkovnih zbirk in pripadajočo kategorijo podobnosti **SimObj** dobili pri posplošitvi relacijske algebre, ki (v nasprotju z dialog mIPC Kripkejevimi modeli) ne temelji na logiki, pri dobljenem modelu ne moremo govoriti o integritetnih omejitvah (razdelek 8.2). Za integritetne omejitve smo sicer trdili, da

so osnova sodelujočih podatkovnih sistemov. Kakor koli že, dobljen kategorni model relacijskih podatkovnih zbirk (v svoji osnovi) ne omogoča naslednjih tehnik sodelujočega odgovarjanja: vrednotenja domnev, vsebovanih v poizvedbah, odkrivanja in odpravljanja nesporazumov ter oblikovanja intenzionalnih odgovorov. Vsekakor pa omogoča tehniko posploševanja poizvedb in odgovorov oz. relaksacijo (razdelek 9.4.2), ki je ključnega pomena takrat, ko govorimo o skupih in časovno odvisnih podatkovnih zbirkah ali o nujnosti ponujanja dosegljivih, sorodnih podatkov.

Omeniti je še treba, da relaksacija v dialog mIPC Kripkejevih modelih ni najbolje implementirana. Če se namreč od pomena relacije *biti manj specifičen*, na kateri temelji relacija dosegljivosti R , preveč oddaljimo, tako da vzpostavimo povezave tudi med krajevno, časovno ali kako drugače sosednimi oz. relevantnimi delčki informacij, lahko tranzitivnost relacije dosegljivosti R vodi do odvečnih, nerelevantnih povezav [Hajdinjak-2004a]. Tranzitivnost relacij predstavlja torej slabost dialog mIPC Kripkejevih modelov in posledično premoč kategornega modela relacijskih podatkovnih zbirk v učinkovitosti relaksacije. Ne le, da kategorija podobnosti **SimObj** omogoča relacije, ki niso tranzitivne, omogoča celo več različnih relacij, definiranih na isti množici. Če definiramo na isti množici več različnih relacij, dobimo pač več različnih objektov kategorije podobnosti.

Kaj so torej prednosti in slabosti dobljenega kategornega modela relacijske algebre (in uporabe teorije kategorij nasploh) pred ostalimi modeli? Prednosti so naslednje:

- ↪ Model je (matematično) naravna posplošitev uveljavljene in priljubljene relacijske algebre iz kategorije množic **Set** na kategorijo podobnosti **SimObj**. Ena izmed podkategorij kategorije podobnosti je tudi kategorija množic. Ostale podkategorije, kot sta npr. kategorija urejenih množic **OrdSet** in kategorija kompaktnih metričnih prostorov **CompMet**, od katerih vsaka porodi svojo kategorno relacijsko algebro, so kategoriji množic v nekem smislu vzporedne. Kategorija urejenih množic **OrdSet** in kategorija kompaktnih metričnih prostorov **CompMet** kategorijo množic **Set** v resnici obogatita.
- ↪ S posplošitvijo relacijske algebre smo dobili močan formalizem, ki že v osnovi dovoljuje relaksacijo poizvedb. Ideja je bila ista kot pri urejenih relacijskih podatkovnih modelih (razdelek 8.5.2) in sodelujočih relacijskih podatkovnih modelih (razdelek 8.5.4), tj. urediti vrednostne množice atributov. Rezultat, ki smo ga dobili, pa je model, ki ima vse sposobnosti omenjenih razširitev relacijskega podatkovnega modela in je poleg tega veliko bolj povezan, pregleden in obetaven. Razširitve poizvedovalnega jezika, ki jih naš model zahteva, so odvisne od stopnje zahtevnosti podatkovnega sistema. Podobno kot pri delno urejeni relacijski algebri, je potreben vsaj en dodatni predikat, ki ponazarja relaksacijo izbranega atributa.
- ↪ Urejenost vrednostnih množic je podana z relacijskimi tabelami in je torej integrirana tako, da je do nje mogoče brez težav dostopati in jo po potrebi spreminjati ali prilagajati uporabniku (razdelek 8.5.2).
- ↪ V eni od prejšnjih točki smo omenili, da je naš model obetavnejši od podobno zmogljivih razširitev relacijskega podatkovnega modela. Ta trditev temelji na

moči in enotnosti formalizma, ki ga daje teorija kategorij, ter verjetnosti obstoja takih kategornih modelov, ki podpirajo tudi katero izmed preostalih tehnik sodelujočega odgovarjanja, ne le relaksacijo. Potrditev te domneve ostane pomembna točka nadaljnjega dela.

Za konec preglejmo še slabosti dobljenega kategornega modela:

- ↔ Model zahteva razširitev poizvedovalnega jezika običajnih relacijskih podatkovnih zbirk, ki temeljijo na klasični relacijski algebri.
- ↔ Izmed vseh petih tehnik sodelujočega odgovarjanja je omogočena le relaksacija oz. posploševanje poizvedb in odgovorov. Idealen sodelujoč podatkovni model bi bil model, ki bi podpiral vse tehnike sodelujočega odgovarjanja.

10 Sklep

10.1 Pregled in pomen doseženih rezultatov

10.2 Nadaljnje delo

Povzamemo rezultate, dosežene v okviru raziskav, predstavljenih v doktorski disertaciji, in opredelimo njihov pomen.

Predlagamo nekatere smernice za nadaljnje raziskovalno delo in predstavimo možnosti izboljšave razvitih podatkovnih modelov.

10.1 Pregled in pomen doseženih rezultatov

10.1.1 Vrednotenje učinkovitosti

V doktorski disertaciji smo podrobno preučili ogrodje PARADISE, ki velja za potencialno splošno metodologijo vrednotenja učinkovitosti sistemov za dialog. Opozorili smo na nekatere pomanjkljivosti in omejitve te metode ter predlagali morebitne rešitve [Hajdinjak-2006]:

1. Model učinkovitosti, ki ga zajema ogrodje PARADISE, vsebuje normalizirane vrednosti neodvisnih spremenljivk. Z normalizacijo dosežemo relevantnost in primerljivost uteži parametrov, ki nastopajo v funkciji učinkovitosti. Opozorili smo na dejstvo, da je treba, če se želimo izogniti prevelikim napakam ocen, normalizirati tudi odvisno spremenljivko, ki izraža zadovoljstvo uporabnikov. Če želimo z dobljenim modelom napovedati zadovoljstvo uporabnikov, je napovedano normalizirano vrednost zadovoljstva uporabnika bolje transformirati nazaj na začetni interval, saj smo pokazali, da je ocena nenormalizirane vrednosti zadovoljstva uporabnika v večini primerov veliko boljša.
2. Uporaba multiple linearne regresije pri reševanju predločenega linearnega sistema zahteva izpolnitev vrste pogojev. Med drugimi mora obstajati približno linearna zveza med odvisno spremenljivko na eni strani in neodvisnimi spremenljivkami na drugi strani, noben par neodvisnih spremenljivk ne sme previsoko korelirati in iz modela je dobro odstraniti tiste neodvisne spremenljivke, ki so z odvisno spremenljivko v zelo nizki korelaciji ali imajo v funkciji učinkovitosti majhne neničelne uteži.
3. Avtorice ogrodja PARADISE [Walker-1997a] zadovoljstvo uporabnikov merijo z vprašalnikom, sestavljenim iz osmih vprašanj, od katerih se vsako vprašanje nanaša na drugi vidik učinkovitosti oz. obnašanja sistema za dialog, tj. na učinkovitost modula za tvorjenje govora, učinkovitost modula za razpoznavanje govora, težavnost pridobivanja informacij, hitrost interakcije, izkušnost uporabnikov, ustreznost odzivov sistema, pričakovano obnašanje sistema in načrtovano rabo sistema v prihodnosti. Poudarili smo, da ta vprašalnik ne temelji niti na teoriji niti na ustreznih empiričnih raziskavah, in zato ne more šteti za veljaven psihometrični instrument. Naslednja sporna točka, na katero smo opozorili, je seštevanje ocen, dodeljenih posameznim vprašanjem. Teorija pravi, da je to dejanje opravičeno le, če vsa vprašanja merijo isto količino, v nasprotnem primeru predstavlja vsota nesmiselno vrednost. Argumentirali smo, da taka vsota ni popolnoma nesmiselna, če merimo učinkovitost celotnega sistema za dialog ali učinkovitost kakšnega od njegovih modulov. Ker nas ponavadi ne zanimajo le izolirane lastnosti izbranega modula, temveč tudi uspešnost modula kot celote, je smiselno na izbran modul gledati kot na merjeno količino. V tem primeru se zdi edino pravilno sešteti ocene, dodeljene le tistim vprašanjem, ki se nanašajo na učinkovitost oz. obnašanje izbranega modula.
4. Parameter, ki igra v funkciji učinkovitosti najpogosteje najpomembnejšo vlogo,

je parameter, ki meri učinkovitost modula za razpoznavanje govora. Prišli smo do sklepa, da bo vrednotenje učinkovitosti posameznih modulov zelo verjetno zanesljivejše in natančnejše, če odstranimo vpliv razpoznavanja govora, torej simuliramo tako rekoč popolno razpoznavanje. Menili smo, da bodo v tem primeru prišli v ospredje tudi tisti parametri modela učinkovitosti, ki jim zaradi izjemnega vpliva učinkovitosti razpoznavanja govora svoje vloge v preteklih študijah ni uspelo dokazati, in trdili, da bodo tako dobljene uteži funkcije učinkovitosti realneje izražale vpliv parametrov na zadovoljstvo uporabnikov.

Ogrodje PARADISE smo uporabili pri vrednotenju učinkovitosti dveh nedograjenih sistemov za podajanje informacij o vremenu in vremenski napovedi, s katerima smo izvajali eksperiment Čarovnik iz Oza. V skladu z našo trditvijo, da je treba vplive samodejnega razpoznavanja govora iz sistema odstraniti, če želimo vrednotiti učinkovitost kakšnega drugega modula (v našem primeru modula za vodenje dialoga), je človek čarovnik v prvem sistemu simuliral razumevanje govora (razpoznavanje govora in razumevanje naravnega jezika) ter vodenje dialoga, v drugem sistemu pa le razumevanje govora. Oba sistema sta se poleg načina vodenja dialoga razlikovala še v vrsti podatkovne zbirke – v prvem eksperimentu je sistem dostopal do relacijske zbirke vremenskih podatkov, v drugem pa do sodelujoče podatkovne zbirke, ki je temeljila na dialog mIPC Kripejevih modelih.

Za namene vrednotenja smo izbrali in določili 25 regresijskih parametrov. Pri vrednotenju učinkovitosti sistemov za podajanje informacij smo predlagali še neuveljavljene parametre podatkovne zbirke, ki izražajo velikost in sestavo podatkovne zbirke. Definirali smo parametre podatkovne zbirke, ki merijo število in delež potez, s katerimi sistem uporabniku poda informacije, ki jih najde v podatkovni zbirki, število in delež potez sistema, ki uporabnika usmerjajo k izbiri relevantnih, dosegljivih podatkov, ter število in delež potez, s katerimi sistem uporabniku sporoča, da zahtevanega podatka nima in ga pri tem ne usmerja k izbiri relevantnih, dosegljivih podatkov. V raziskave smo vključili kvantitativne in proporcionalne parametre podatkovne zbirke. Ugotovili smo, da so bili uporabniki prvega sistema bolj dojemljivi za kvantitativne parametre, v drugem pa za proporcionalne parametre. Menili smo, da je to posledica konsistentno povečanega ponujanja relevantnih informacij v drugem eksperimentu, ki je vodilo do več novih informacijskih iger in s tem do večje dojemljivosti uporabnikov za proporcionalne količine.

Ker je bila razlika v natančnosti obeh funkcij učinkovitosti za odvisno spremenljivko, ki jo predlaga ogrodje PARADISE, torej seštevek vseh ocen, pridobljenih z vprašalnikom o zadovoljstvu uporabnikov, prevelika ($R^2 = 0.58$ proti $R^2 = 0.24$), smo sklepali, da te odvisne spremenljivke ni mogoče dovolj dobro modelirati. Predvidevali smo, da se to zgodi, ker verjetno ne merimo tistega, kar bi želeli meriti. Ker smo želeli poiskati razlike med dvema sistemoma Čarovnik iz Oza, ki sta se razlikovala le v načinu vodenja dialoga in predstavitvi znanja, smo novo mero zadovoljstva uporabnikov definirali kot vsoto ocen, ki se nanašajo na vpeljane spremembe. Na ta način nam je uspelo razliko v natančnosti funkcij učinkovitosti obeh sistemov Čarovnik iz Oza izrazito zmanjšati ($R^2 = 0.57$ proti $R^2 = 0.44$).

Po vzratni eliminaciji smo dobili funkciji učinkovitosti, ki ne vsebujeta nobenega skupnega parametra. Še več, nobeden od parametrov, ki jih vsebuje funkcija učinkovitosti prvega sistema Čarovnik iz Oza, ni bil statistično značilen za izbrano odvisno spremenljivko v drugem eksperimentu. Edini parameter, ki nastopa v funkciji učinkovitosti drugega sistema in je bil statistično značilen tudi v prvem eksperimentu, pa je eden od parametrov podatkovne zbirke. Prišli smo do spoznanja, da so parametri podatkovne zbirke edina podobnost med funkcijama učinkovitosti obeh sistemov Čarovnik iz Oza in da ima predstavitev znanja v sistemih za podajanje informacij velik pomen. Ugotovili smo tudi, da strategija usmerjanja uporabnika k izbiri dosegljivih, relevantnih podatkov na zadovoljstvo uporabnikov ne vpliva negativno in da je treba težiti k povečanju stopnje sodelujočega odgovarjanja, predvsem relaksacije.

Preverili smo, kako dobro je mogoče modelirati posamezne ocene uporabnikov in kateri parametri na njih najmočneje vplivajo. Najslabše izmed vseh smo modelirali hitrost interakcije in načrtovano rabo sistema v prihodnosti. Argumentirali smo, da je bila hitrost interakcije s sistemom pod velikim vplivom čarovnika in zato precej nepredvidljiva, ter da pozitivne izkušnje uporabnikov v testiranjih sistemov za dialog zelo verjetno niso kazalec njihovih prihodnjih navad. Kljub pričakovanjem nas je dejstvo, da je imel parameter, ki meri število nepodanih informacij, v prvem eksperimentu absolutno največjo (sicer negativno) utež sedmih izmed osmih funkcij učinkovitosti, v drugem eksperimentu pa je bil statistično neznačilen za prvih sedem ocen, nekoliko presenetilo. Razen tega je bila večina najznačilnejših parametrov posameznih ocen iz drugega eksperimenta statistično neznačilna za isto oceno iz prvega eksperimenta.

Ker šteje ogrodje PARADISE za potencialno splošno metodologijo vrednotenja učinkovitosti sistemov za dialog in je trenutno najpogosteje citirana metoda na tem področju, predstavljajo analiza metode in rezultati, dobljeni v doktorski disertaciji, pomemben prispevek k izpopolnitvi vrednotenja učinkovitosti sistemov za dialog ter dobro začrtane smernice nadaljnjih raziskav.

10.1.2 Predstavitev znanja

Ker imajo priljubljeni relacijski podatkovni modeli vrsto pomanjkljivosti, od katerih v sistemih za podajanje informacij najbolj izstopa nesposobnost sodelujočega odgovarjanja (predvsem nepodpora relaksacije), smo izpostavili potrebo po razvoju podatkovnega modela, ki bi bil relacijski algebrni kot najbolj znanemu relacijskemu podatkovnemu modelu čim bolj podoben in hkrati sodelujoč. Dialog mIPC Kripkejevi podatkovni modeli, ki so rezultat magistrskega dela [Hajdinjak-2004a], sicer podpirajo skoraj vse tehnike sodelujočega odgovarjanja, se pa od relacijske algebre zelo razlikujejo, saj temeljijo na precej zahtevni intuicionistični modalni logiki. Naš cilj je bil zato dialog mIPC Kripkejeve modele približati univerzalno zastavljeni relacijski algebri oz. razviti relacijski algebrni podoben podatkovni model, ki bi bil po zmogljivosti vsaj delno primerljiv z dialog mIPC Kripkejevimi modeli.

Formalizem, ki smo ga v ta namen uporabili, je bila teorija kategorij. Relacijsko algebro kot najbolj znan model relacijskih podatkovnih zbirk, definiran v kategoriji množic, smo (matematično) naravno posplošili na teorijo kategorij. Osnovnim objek-

tom relacijske algebre, tj. relacijam oz. podmnožicam kartezičnega produkta množic in nad njimi definiranim operacijam (unija, razlika, kartezični produkt, projekcija in izbira) smo priredili ustrezne kategorne konstrukte. Ugotovili smo, da morajo kategorije, ki bodo (poleg najnaravnejše kategorije množic) primerna osnova kategornega modela relacijskih podatkovnih zbirk, najprej vsebovati natančen začetni objekt, ki je kategorna posplošitev pojma prazne podmnožice, ter končni objekt, ki je kategorna posplošitev pojma podmnožice z enim samim elementom. Kategorno posplošitev kartezičnega produkta in projekcije smo dosegli s končnimi kategornimi produkti in slikami. Za kategorno posplošitev preostalih treh osnovnih operacij relacijske algebre (izbire, unije in razlike) smo zahtevali še obstoj natančne spodnje in natančne zgornje meje poljubnega para regularnih podobjektov ter obstoj relativnega komplementa v delno urejenih razredih regularnih podobjektov. Sklenili smo, da so kategorije, ki izpolnjujejo vse naštetje pogoje, primerna izbira za kategorno predstavitev relacijske algebre, kategorija množic pa prva in najnaravnejša tovrstna kategorija.

Podali smo primer kategorije, v kateri je mogoče predstaviti relacijsko algebro, tj. kategorijo podobnosti, katere objekti so urejene trojice, sestavljene iz množice, polne mreže in mere podobnosti. O dejstvu, da je kategorija podobnosti zelo koristna pri modeliranju sodelujočih podatkovnih zbirk, sta nas prepričali njeni podkategoriji, imenovani kategorija urejenih množic in kategorija kompaktnih metričnih prostorov. Objekti kategorije urejenih množic so množice, urejene z refleksivno relacijo, morfizmi pa preslikave, ki ohranjajo urejenost. Ker je urejenost kot osnovna lastnost skoraj vseh podatkovnih tipov naravno prisotna v številnih podatkovnih zbirkah, relacijske podatkovne zbirke pa podpirajo le standardne urejenosti podatkovnih tipov, tj. abecedno urejenost črkovnih nizov, numerično urejenost števil in kronološko urejenost datumov, kategorija urejenih množic prepreči izgubo pomembne semantične informacije. Ta integracija urejenosti predstavlja temelj tehnike posploševanja poizvedb in odgovorov oz. relaksacije. Objekti kategorije kompaktnih metričnih prostorov so kompaktni metrični prostori, urejeni z metriko, morfizmi pa Lipschitzove preslikave kompaktnih metričnih prostorov. Metrika pride npr. do izraza, ko poizvedbo pogojimo z razdaljami med kraji.

Ker relacijske podatkovne zbirke, ki temeljijo na relacijski algebri, definirani v kategoriji množic, udejanjimo kot zbirko tabel, smo želeli na ta način udejanjiti tudi podatkovne zbirke, ki temeljijo na kategorni posplošitvi relacijske algebre, predstavljeni v kategoriji podobnosti. Videli smo, da lahko mero podobnosti, ki je del objektov kategorije podobnosti, predstavimo kot relacijo in jo posledično udejanjimo kot relacijsko tabelo. Na ta način smo vsakemu objektu priredili dve relacijski tabeli, ena predstavlja množico, druga pa pripadajočo mero podobnosti. Podobnost na vrednostnih množicah, ki smo jo podali z dodatnimi relacijskimi tabelami, je torej integrirana tako, da je do nje mogoče brez težav dostopati in jo po potrebi spreminjati ali prilagajati uporabniku. Na meri podobnosti tako temelji implementacija relaksacije v posplošene relacijske podatkovne sisteme, katere namen je v odgovor zajeti večjo količino sorodnih, relevantnih in/ali smiselnih podatkov, in jo uporabljamo, ko je poizvedba neuspešna ali ko uporabnik v množici odgovorov ne najde sprejemljivega podatka.

Z relacijo *biti manj specifičen* urejene vremenske podatke, uporabljene v drugem eksperimentu Čarovnik iz Oza, smo predstavili v kategoriji urejenih množic in ugotovili, da omogoča kategorija podobnosti (v resnici pa že njena podkategorija, kategorija urejenih

množic) bolj kakovostno in natančno relaksacijo kot dialog mIPC Kripkejevi modeli.

S posplošitvijo relacijske algebre smo dobili močan formalizem, ki torej že v osnovi dovoljuje relaksacijo poizvedb. Trditev, da je tak model obetavnejši od podobno zmogljivih razširitev relacijskega podatkovnega modela, smo podprli z enotnostjo formalizma, ki ga daje teorija kategorij, in verjetnostjo obstoja takih kategorij modelov, ki podpirajo tudi katero izmed preostalih tehnik sodelujočega odgovarjanja, ne le relaksacijo.

10.2 Nadaljnje delo

Ker za nobeno od obstoječih tehnik merjenja zadovoljstva uporabnikov sistemov za dialog ni dokazano, da izpolnjuje pogoje za veljaven psihometrični instrument, je treba vse sklepe, ki zajemajo zadovoljstvo uporabnikov, obravnavati zelo previdno. Žal je bil prvi resen poskus razvoja vprašalnika, ki bi zanesljivo, veljavno, objektivno in diskriminativno meril zadovoljstvo uporabnikov sistemov za dialog, (začasno) prekinjen [Hone-2000]. Izrazili smo mnenje, da je tudi modifikacija SERVQUAL metode [Hartikainen-2004] pristop, ki veliko obeta. Menimo, da je ravno neustrezno merjenje zadovoljstva uporabnikov najbolj bistvena in kritična pomanjkljivost ogrodja PARADISE.

Obremenjeni z vprašalnikom, ki ga zajema ogrodje PARADISE, in njegovimi pomanjkljivostmi, smo podali zamisel o idealnem načinu merjenja zadovoljstva uporabnikov z učinkovitostjo sistemov za dialog. Predlagali smo uteženo vsoto ali uteženo srednjo vrednost posameznih ocen, ki se nanašajo na različne vidike obnašanja sistema za dialog. Lastnostim, ki bi se uporabnikom zdele bolj pomembne, bi v uteženi vsoti (ali uteženi srednji vrednosti) dodelili večje uteži. Vprašanje o relativni velikosti uteži, ki je pri tem osrednjega pomena, bi moralo temeljiti na ustreznih empiričnih raziskavah, v katere bi bilo vključenih dovolj uporabnikov sistemov za dialog.

Pri analizi podatkov, pridobljenih v dveh eksperimentih Čarovnik iz Oza, smo ugotovili, da so bili uporabniki prvega sistema bolj dojemljivi za kvantitativne parametre, v drugem pa za proporcionalne parametre. Izpostavili smo nujnost nadaljnjih raziskav glede tega vprašanja, kajti izbira regresijskih parametrov je lahko ključnega pomena pri doseganju zadostne natančnosti funkcije učinkovitosti. Kljub temu smo postavili hipotezo, da bi bila lahko ta razlika posledica konsistentno povečanega ponujanja relevantnih informacij v drugem eksperimentu, kar bi vodilo do več novih informacijskih iger in s tem do morebitne večje dojemljivosti uporabnikov za proporcionalne količine.

Izstopa tudi ugotovitev, da je bila večina statistično najznačilnejših parametrov posameznih ocen iz drugega eksperimenta statistično neznačilna za isto oceno iz prvega eksperimenta. Na to ugotovitev lahko gledamo kot na potrditev neprimernosti posamičnih ocen za merjenje uporabnikovega zadovoljstva z učinkovitostjo sistemov za dialog. Po drugi strani pa bi to lahko pomenilo, da odgovori na izbrana vprašanja kažejo le na velike razlike med obema sistemoma Čarovnik iz Oza. Treba je torej tudi razumeti in biti sposoben pravilno opredeliti razlike med različnimi funkcijami učinkovitosti – naj bo to med funkcijami, ki se nanašajo na več različic danega sistema

za dialog, ali med funkcijami, ki se našajo na različne sisteme ali celo različne načine merjenja zadovoljstva uporabnikov.

Posplošitev relacijske algebre na teorijo kategorij je vodila do karakterizacije kategorij, s katerimi je mogoče modelirati relacijsko algebro, in primera tovrstne kategorije, imenovane kategorija podobnosti. Videli smo, da dobljeni kategorni modeli relacijske algebre omogočajo posplošitev poizvedb in odgovorov oz. relaksacijo, ki temelji na meri podobnosti, integrirani v objekte kategorije podobnosti. Posplošitev relacijske algebre pa zahteva tudi razširitev poizvedovalnega jezika. Čeprav je razširitev poizvedovalnega jezika, ki jo podatkovni model zahteva, odvisna od stopnje zahtevnosti podatkovnega sistema, je treba poizvedovalni jezik relacijske algebre razširiti, tako da bo omogočal izkoriščanje celotne zmogljivosti dobljenih kategornih modelov relacijske algebre.

Trdili smo, da je naš model obetavnejši od podobno zmogljivih razširitev relacijskega podatkovnega modela. Po eni strani je to res, ker ima teorija kategorij močan in enoten formalizem. Res pa je tudi, da v razredu kategorij, s katerimi je mogoče modelirati relacijsko algebro, poleg kategorije podobnosti, ki podpira le relaksacijo, lahko obstajajo tudi kategorije, ki podpirajo katero izmed preostalih tehnik sodelujočega odgovarjanja, ne le relaksacijo. Identifikacija takih kategorij, njihova opredelitev in implementacija so pomembni cilji nadaljnjega dela.

Viri in literatura

- [Abramowitz-1972] M. Abramowitz in I. A. Stegun. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover Publications, Inc., New York. 1972.
- [Abramsky-1994] S. Abramsky in A. Jung. Domain Theory. str. 1–168. V Handbook of Logic in Computer Science, Volume 3. Uredili S. Abramsky, D. Gabbay in T. S. E. Maibaum. Clarendon Press, Oxford. 1994.
- [Allen-1994] J. F. Allen. Natural Language Understanding. Benjamin/Cummings, Redwood City. 1994.
- [Allen-1995] J. F. Allen, L. K. Schubert, G. Ferguson, P. Heeman, C.-H. Hwang, T. Kato, M. Light, N. G. Martin, B. W. Miller, M. Poesio in D. R. Traum. The TRAINS Project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*. Zv. 7, str. 7–48. 1995.
- [Allen-1997] J. F. Allen in M. Core. Draft of DAMSL: Dialog Act Markup in Several Layers. *Multiparty Discourse Group at the Discourse Research Initiative Meeting, Schloss Dagstuhl, Nemčija*. Dosegljivo na <http://www.cs.rochester.edu/research/cisd/resources/damsl/RevisedManual/RevisedManual.html>.
- [Awodey-2005] S. Awodey. Category Theory. Carnegie Mellon University, ZDA. Dosegljivo na <http://www.andrew.cmu.edu/course/80-413-713/notes/cats.pdf>. 2005.
- [Barr-1999] M. Barr in C. Wells. Category Theory: Lecture Notes for ESSLLI. McGill University, Montreal, Kanada. Dosegljivo na <http://www.math.upatras.gr/~cdrossos/Docs/B-W-Lecture Notes.pdf>. 1999.
- [Barras-2001] C. Barras, E. Geoffrois, Z. Wu in M. Liberman. Transcriber: use of a tool for assisting speech corpora production. *Speech Communication: Special issue on Speech Annotation and Corpus Tools*. Zv. 33(1–2), str. 5–22. 2001.

- [Bates-1991] M. Bates in D. Ayuso. A Proposal for Incremental Dialogue Evaluation. *V Proceedings of DARPA Speech and Natural Language Workshop, Pacific Grove, ZDA. str. 319–322. 1991.*
- [Borceux-1994a] F. Borceux. Handbook of Categorical Algebra 1, Basic Category Theory. *Cambridge University Press, Cambridge. 1994.*
- [Borceux-1994b] F. Borceux. Handbook of Categorical Algebra 2, Categories and Structures. *Cambridge University Press, Cambridge. 1994.*
- [Borceux-1994c] F. Borceux. Handbook of Categorical Algebra 3, Categories of Sheaves. *Cambridge University Press, Cambridge. 1994.*
- [Bouwman-1998] G. Bouwman in J. Hulstijn. Dialogue Strategy Redesign with Reliability Measures. *V Proceedings of the 1st Language Resources and Evaluation Conference, Granada, Španija. 1998.*
- [Boyce-1996] A. L. Boyce in A. L. Gorin. User Interface Issues for Natural Spoken Dialogue Systems. *V Proceedings of the 1996 International Symposium on Spoken Dialogue, Philadelphia, ZDA. str. 65–68. 1996.*
- [Bratman-1988] M. Bratman, D. Israel in M. Pollack. Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence. Zv. 4, str. 349–355. 1988.*
- [Braun-2003] D. Braun, J. Mayberry, A. Powers in S. Schlicker. The Geometry of the Hausdorff Metric. *Dosegljivo na http://faculty.gvsu.edu/schlicks/Hausdorff_paper.pdf. 2003.*
- [Bucur-1968] I. Bucur in A. Deleanu. Introduction to the Theory of Categories and Functors. *John Wiley & Sons, London. 1968.*
- [Buneman-1991] P. Buneman, P. Jung in A. Ohori. Using Powerdomains to Generalize Relational Databases. *Theoretical Computer Science. Zv. 9 (1), str. 23–55. 1991.*
- [Carletta-1996a] J. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon in A. Anderson. HCRC Dialogue Structure Coding Manual. *Research paper 82, Human Communication Research Centre, University of Edinburgh, Edinburgh, Velika Britanija. 1996.*
- [Carletta-1996b] J. C. Carletta. Assessing the Reliability of Subjective Codings. *Computational Linguistics. Zv. 22(2), str. 249–254. 1996.*
- [Ceri-1989] S. Ceri, G. Gottlob in L. Tanca. What You Always Wanted to Know About Datalog (And Never Dared to Ask). *IEEE Transactions on Knowledge and Data Engineering. Zv. 1 (1), str. 146–166. 1989.*

- [Chaudhuri-1998] S. Chaudhuri. An Overview of Query Optimization in Relational Systems. *V Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Seattle, ZDA. str. 34-43. 1998.*
- [Cholvy-1986] L. Cholvy in R. Demolombe. Querying a Rule Base. *V Proceedings of the 1st International Workshop on Expert Database Systems, Charleston, ZDA. str. 365-371. 1986.*
- [Cholvy-1990] L. Cholvy. Answering Queries Addressed to a Rule Base. *Revue d'Intelligence Artificielle. Zv. 1(1), str. 79-98. 1990.*
- [Chu-1990] W. W. Chu, Q. Chen in R-C. Lee. Cooperative Query Answering via Type Abstraction Hierarchy. *Technical Report, University of California, Los Angeles, ZDA. 1990.*
- [Chu-1994] W. W. Chu in Q. Chen. A Structured Approach for Cooperative Query Answering. *IEEE Transactions on Knowledge and Data Engineering. Zv. 6 (5), str. 738-749. 1994.*
- [Chu-1996] W. W. Chu, H. Jung, K. Chiang, M. Minock, G. Chow in C. Larson. CoBase: A Scalable and Extensible Cooperative Information System. *Journal of Intelligent Information Systems. Zv. 6(2/3), str. 223-259. 1996.*
- [Codd-1970] E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM. Zv. 13(6), str. 377-387. 1970.*
- [Codd-1974] E. F. Codd. Seven Steps to RENDEZVOUS with the Casual User. *V Data Base Management. Uredila J. Kimbie in K. Koffeman. North-Holland Publishers, Amsterdam. 1974.*
- [Cohen-1995] P. R. Cohen in S. L. Oviatt. The Role of Voice Input for Human-Machine Communication. *V Proceedings of the National Academy of Sciences, ZDA. Zv. 92(22), str. 9921-9927. 1995.*
- [Colomb-1998] R. M. Colomb. Deductive Databases and their Applications. *Taylor & Francis Ltd, London. 1998.*
- [Cuppens-1988] F. Cuppens in R. Demolombe. Cooperative Answering: A Methodology to Provide Intelligent Access to Databases. *V Proceedings of the 2nd International Conference on Expert Database Systems, Virginia, ZDA. str. 621-643. 1988.*
- [Dahlbäck-1993] N. Dahlbäck, A. Jönsson in L. Ahrenberg. Wizard of Oz studies: why and how. *V Proceedings of the international workshop on Intelligent user interfaces, Orlando, ZDA. str. 193-200. 1993.*

- [Dampney-1992] C. N. G. Dampney, M. Johnson in G. P. Monro. An Illustrated Mathematical Foundation for ERA. *V Proceedings of the Institute for Mathematics and its Applications*. *Zv. 35*, str. 77–84. 1992.
- [Danieli-1995] M. Danieli in E. Gerbino. Metrics for Evaluating Dialogue Strategies in a Spoken Language System. *V Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, Stanford, ZDA*. str. 34–39. 1995.
- [Das-1992] S. K. Das. *Deductive Databases and Logic Programming*. Addison-Wesley Publishers Ltd, Wokingham. 1992.
- [Demmel-1997] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia. 1997.
- [DiEugenio-2004] B. Di Eugenio in M. Glass. The Kappa Statistic: A Second Look. *Computational Linguistics*. *Zv. 30(1)*, str. 95–101. 2004.
- [Diestel-2000] R. Diestel. *Graph Theory*. Springer Verlag, New York. 2000.
- [Eilenberg-1945] S. Eilenberg in S. MacLane. General Theory of Natural Equivalences. *Transactions of the American Mathematical Society*. *Zv. 58*, str. 231–294. 1945.
- [Elmasri-2004] R. Elmasri in S. B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Pearson Education, Inc., Boston. 2004.
- [Eskenazi-1999] M. Eskenazi, A. Rudnicky, K. Gregory, P. Constantinides, R. Brennan, C. Bennett in J. Allen. Data Collection and Processing in the Carnegie Mellon Communicator. *V Proceedings of the 6th European Conference on Speech Communication and Technology, Budimpešta, Madžarska*. *Zv. 6*, str. 2695–2698. 1999.
- [Falconer-1990] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, New York. 1990.
- [Fraser-1991] N. M. Fraser in G. N. Gilbert. Simulating Speech Systems. *Computer Speech and Language*. *Zv. 5(1)*, str. 81–99. 1991.
- [Gaasterland-1992a] T. Gaasterland, P. Godfrey in J. Minker. An Overview of Cooperative Answering. *Journal of Intelligent Information Systems*. *Zv. 1(2)*, str. 123–157. 1992.
- [Gaasterland-1992b] T. Gaasterland, P. Godfrey in J. Minker. Relaxation as a Platform of Cooperative Answering. *Journal of Intelligent Information Systems*. *Zv. 1(3/4)*, str. 293–321. 1992.

- [Gal-1987] A. Gal in J. Minker. Informative and Cooperative Answers in Databases Using Integrity Constraints. V *Proceedings of the 2nd Natural Language Understanding and Logic Programming Workshop, Vancouver, Kanada*. str. 277-300. 1987.
- [Gallaire-1978] H. Gallaire in J. Minker. Logic and Databases. Plenum Press, New York. 1978.
- [Ginchev-1998] I. Ginchev in A. Hoffman. The Hausdorff Nearest Circle to a Convex Compact Set in the Plane. *Journal for Analysis and its Applications*. Zv. 17(2), str. 479-499. 1998.
- [Goddeau-1996] D. Goddeau, H. Meng, J. Polifroni, S. Seneff in S. Busayapongchai. A Form-Based Dialogue Manager for Spoken Language Applications. V *Proceedings of the 4th International Conference on Spoken Language Processing, Philadelphia, ZDA*. str. 701-704. 1996.
- [Godfrey-1994] P. Godfrey, J. Minker in L. Novik. An Architecture for a Cooperative Database System. str. 3-24. V *Lecture Notes in Computer Science 819, Applications of Databases*. Uredila W. Litwin in T. Risch. Springer Verlag, Berlin - Heidelberg - New York. 1994.
- [Godfrey-1998] P. Godfrey, J. Grant, J. Gryz in J. Minker. Integrity Constraints: Semantics and Applications. str. 265-307. V *Logics for Databases and Information Systems*. Uredila J. Chomicki in G. Saake. Kluwer Academic Publishers, Boston. 1998.
- [Grätzer-1998] G. Grätzer. General Lattice Theory, Second Edition. Birkhäuser Verlag, Basel - Boston - Berlin. 1998.
- [Grice-1975] H. Grice. Logic and Conversation. V *Syntax and Semantics*, Zv. 3, *Speech Acts*. str. 41-58. Uredila P. Cole in J. Morgan. Academic Press, New York. 1975.
- [Grimes-1998] S. Grimes. Modeling Object/Relational Databases. *DBMS Online*. Dosegljivo na <http://www.dbmsmag.com/9804d13.html>. Zv. 11(4), str. 51-56. 1998.
- [Gros-1997] J. Gros. Samodejno tvorjenje govora iz besedil. *Doktorska disertacija*. Fakulteta za elektrotehniko, Univerza v Ljubljani. Ljubljana. 1997.
- [Guting-1989] R. H. Guting, R. Zicari in D. M. Choy. An Algebra for Structured Office Documents. *ACM Transactions on Office Information Systems*. Zv. 7(4), str. 123-157. 1989.
- [Hafenrichter-2005] B. Hafenrichter in W. Kiessling. Optimization of Relational Preference Queries. V *Proceedings of the 16th Australasian*

- Database Conference, Newcastle, Australija. str. 175–184. 2005.*
- [Hajdinjak-2002a] M. Hajdinjak in F. Mihelič. Podatkovna baza nastajajočega sistema za podajanje vremenskih napovedi. V *Zbornik 11. mednarodne Elektrotehniške in računalniške konference ERK 2002, Portorož, Slovenija. Zv. B, str. 299–302. 2002.*
- [Hajdinjak-2002b] M. Hajdinjak in F. Mihelič. Semantična analiza vremenskih napovedi. V *Zbornik B 5. mednarodne multikonference Informacijska družba IS'2002, Ljubljana, Slovenija. str. 10–13. 2002.*
- [Hajdinjak-2003a] M. Hajdinjak in F. Mihelič. The wizard of Oz system for weather information retrieval. *str. 400–405. V Lecture notes in computer science, Lecture notes in artificial intelligence, 2807, Uredila V. Matoušek in P. Mautner. Springer Verlag, Berlin. 2003.*
- [Hajdinjak-2003b] M. Hajdinjak in F. Mihelič. Wizard of Oz experiments. V *Proceedings of the IEEE Region 8 EUROCON 2003 : computer as a tool, Ljubljana, Slovenija. Zv. 2, str. 112–116. 2003.*
- [Hajdinjak-2004a] M. Hajdinjak. Vodenje dialoga med človekom in računalnikom v naravnem jeziku. *Magistrsko delo. Fakulteta za elektrotehniko, Univerza v Ljubljani. Ljubljana. 2004.*
- [Hajdinjak-2004b] M. Hajdinjak in F. Mihelič. Conducting the Wizard-of-Oz experiment. *Informatica. Zv. 28(4), str. 425–430. 2004.*
- [Hajdinjak-2004c] M. Hajdinjak in F. Mihelič. Information-Providing Dialogue Management. *str. 595-602. V Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence 3206. Uredili P. Sojka, I. Kopeček in K. Pala. Springer Verlag, Berlin. 2004.*
- [Hajdinjak-2005] M. Hajdinjak in F. Mihelič. Results from an Evaluation of a Dialogue Manager. V *Proceedings of the COST278 Final Workshop and ISCA Tutorial and Research Workshop (ITRW) on Applied Spoken Language Interaction in Distributed Environments, Aalborg, Danska. 2005.*
- [Hajdinjak-2006] M. Hajdinjak in F. Mihelič. The PARADISE Evaluation Framework: Issues and Findings. *Computational Linguistics. Zv. 32(2). 2006.*
- [Hartikainen-2004] M. Hartikainen, E.-P. Salonen in M. Turunen. Subjective Evaluation of Spoken Dialogue Systems Using SERVQUAL Method. V *Proceedings of the 8th International Conference on Spoken Language Processing, Jeju, Koreja. str. 2273–2276. 2004.*

- [Hintikka-1975] *J. Hintikka*. The Intentions of Intentionality and Other New Models for Modalities. *D. Reidel Publishing Company, Dordrecht*. 1975.
- [Hladnik-2002] *M. Hladnik*. Verjetnost in statistika – Zapiski predavanj. Založba FE in FRI, Ljubljana. 2002.
- [Hoeven-1995] *G. van der Hoeven, J. Andernach, S. van der Burgt, G. J. Kruijff, A. Nijholt, J. Schaake* in *F. de Jong*. SCHISMA: A natural language accessible theatre information and booking system. *V Proceedings of the 1st International Workshop on Applications of Natural Language to Data Bases, Versailles, Francija*. str. 271–285. 1995.
- [Hone-2000] *K. S. Hone* in *R. Graham*. Towards a tool for the Subjective Assessment of Speech System Interfaces (SASSI). *Natural Language Engineering*. *Zv. 6(3/4)*, str. 287–303. 2000.
- [Houghton-1987] *G. Houghton* in *S. D. Isard*. Why to speak, what to say and how to say it: Modelling language production in discourse. str. 249–267. *V Modelling Cognition*. Uredil *P. Morris*. *John Wiley and Sons, London*. 1987.
- [Huth-2004] *M. Huth* in *M. Ryan*. Logic in Computer Science, Modelling and Reasoning about Systems, Second Edition. *Cambridge University Press, Cambridge*. 2004.
- [Imrich-2000] *W. Imrich* in *S. Klavžar*. Product Graphs: Structure and Recognition. *John Wiley & Sons, New York*. 2000.
- [Ipšič-1999] *I. Ipšič, F. Mihelič, S. Dobrišek, J. Gros* in *N. Pavešič*. A Slovenian Spoken Dialog System for Air Flight Inquires. *V Proceedings of the 6th European Conference on Speech Communication and Technology, Budimpešta, Madžarska*. str. 2659–2662. 1999.
- [Islam-1994] *A. Islam* in *W. Phoa*. Categorical Models of Relational Databases I: Fibrational Formulation, Schema Integration. str. 618–641. *V Lecture Notes in Computer Science 789, Theoretical Aspects of Computer Software*. Uredila *M. Hagiya* in *J. C. Mitchell*. *Springer Verlag, Berlin – Heidelberg – New York*. 1994.
- [Johnson-1996] *M. Johnson*. A Categorical Logic for Information Systems. *Journal of the IGPL*. *Zv. 4*, str. 502–504. 1996.
- [Johnson-2002a] *M. Johnson* in *R. Rosebrugh*. Sketch Data Models, Relational Schema and Data Specifications. *Electronic Notes in Theoretical Computer Science*. *Zv. 61*, str. 1–13. 2002.

- [Johnson-2002b] M. Johnson, R. Rosebrugh in R. J. Wood. Entity-Relationship-Attribute Designs and Sketches. *Theory and Applications of Categories*. *Zv. 10*, str. 94–112. 2002.
- [Johnson-2002c] R. A. Johnson in D. W. Wichern. Applied multivariate statistical analysis. *Prentice-Hall, Upper Saddle River (NJ)*. 2002.
- [Joshi-1981] A. Joshi, B. Webber in I. Sag. Elements of Discourse Understanding. *Cambridge University Press, Cambridge*. 1981.
- [Jurafsky-1994] D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler in N. Morgan. The Berkeley Restaurant Project. *V Proceedings of the International Conference on Spoken Language Processing, Yokohama, Japonska*. str. 2139–2142. 1994.
- [Kamm-1998] C. A. Kamm, D. J. Litman in M A. Walker. From Novice to Expert: The Effect of Tutorials on User Expertise with Spoken Dialogue Systems. *V Proceedings of the 5th International Conference on Spoken Language Processing, Rundle Mall, Avstralija*. *Zv. 4*, str. 1211–1214. 1998.
- [Kamm-1999] C. Kamm, M. Walker in D. Litman. Evaluating Spoken Language Systems. *V Proceedings of the American Voice Input/Output Society, San Jose, ZDA*. 1999.
- [Kim-1995] W. Kim. Modern Database Systems: The Object Model, Interoperability, and Beyond. *ACM Press in Addison-Wesley, New York*. 1995.
- [Kim-1997] W. Kim in M.-W. Koo. A Korean Speech Corpus for Train Ticket Reservation Aid system Based on Speech Recognition. *V Proceedings of the 5th European Conference on Speech Communication and Technology, Rodos, Grčija*. *Zv. 4*, str. 1723–1726. 1997.
- [Kowalski-1974] R. Kowalski. Predicate Logic as a Programming Language. *V Proceedings of the 6th International Federation for Information Processing Congress, Stockholm, Švedska*. str. 569–574. 1974.
- [Kowtko-1992] J. Kowtko in S. Isard. Conversational Games Within Dialogue. *Research paper 31, Human Communication Research Centre, University of Edinburgh, Edinburgh, Velika Britanija*. 1992.
- [Krahmer-2001] E. J. Krahmer. The Science and Art of Voice Interfaces. *Philips Research Report, Philips, Eindhoven, Nizozemska*. 2001.
- [Kuhns-1967] J. Kuhns. Answering Questions by Computer: A Logical Study. *Technical Report RM-5428-PR, The Rand Corporation, Santa Monica, ZDA*. 1967.

- [Kurzweil-1999] R. Kurzweil. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*. Viking Press, New York. 1999.
- [Larsen-2003] L. B. Larsen. Issues in the Evaluation of Spoken Dialogue Systems using Objective and Subjective Measures. *V Proceedings of the Automatic Speech Recognition and Understanding Workshop, St. Thomas, ZDA. str. 209–214. 2003.*
- [Larsson-2000] S. Larsson in D. Traum. Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering. Zv. 6(3/4), str. 323–340. 2000.*
- [Levien-1965] R. Levien in M. Maron. Relational Data File: A Tool for Mechanized Inference Execution and Data Retrieval. *Technical Report RM-4793-PR, The Rand Corporation, Santa Monica, ZDA. 1965.*
- [Lewin-2000] I. Lewin. A formal model of Conversational Game Theory. *V Götalog: Proceedings of the 4th Workshop on the Semantics and Pragmatics of Dialogue, Göteborg, Švedska. 2000.*
- [Litman-2002] D. J. Litman in P. Shimei. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction. Zv. 12, str. 111–137. 2002.*
- [Lynn-1982] N. Lynn. Implementation of Ordered Relations in a Data Base System. *Magistrsko delo. University of California, Department of Electrical Engineering and Computer Science. Berkeley. 1982.*
- [Maier-1993] D. Maier in B. Vance. A Call to Order. *V Proceedings of the 12th ACM Symposium on Principles of Database Systems, Washington, ZDA. str. 1–16. 1993.*
- [Martinčić-2003] S. Martinčić-Ipšić, J. Žibert, I. Ipšić, F. Mihelič in N. Pavešić. Bilingual Speech Recognition for a Weather Information Retrieval Dialogue System. *str. 380–387. V Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence 2807. Uredila V. Matoušek in P. Mautner. Springer Verlag, Berlin. 2003.*
- [Mata-Toledo-2000] R. Mata-Toledo in P. Cushman. *Schaum's Outline of Fundamentals of Relational Databases*. McGraw-Hill, New York. 2000.
- [McSkimin-1977] J. McSkimin in J. Minker. The Use of a Semantic Network in Deductive Question-Answering Systems. *V Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, ZDA. str. 50–58. 1977.*

- [Minker-1982] J. Minker. On Indefinite Databases and the Closed World Assumption. *V Proceedings of the 6th Conference on Automated Deduction, New York, ZDA. str. 292–308. 1982.*
- [Minker-1996] J. Minker. Logic and Databases: a 20 Year Retrospective. *V Proceedings of the 1st International Workshop on Logic in Databases, San Miniato, Italija. str. 3–57. 1996.*
- [Mohor-1997] T. Mohor. Načrtovanje relacijskih podatkovnih baz. *BI-TIM d.o.o, Ljubljana. 1997.*
- [Möller-2005] S. Möller. Evaluating Telephone-Based Interactive Systems. *V Proceedings of the COST278 Final Workshop and ISCA Tutorial and Research Workshop (ITRW) on Applied Spoken Language Interaction in Distributed Environments, Aalborg, Danska. 2005.*
- [Moore-2005] R. K. Moore. Research Challenges in the Automation of Spoken Language Interaction. *V Proceedings of the Workshop on Applied Spoken Language Interaction in Distributed Environments, Aalborg, Danska. 2005.*
- [Motro-1988] A. Motro. VAGUE: A User Interface to Relational Databases that Permits Vague Queries. *ACM Transactions on Office Information Systems. Zv. 6(3), str. 187–214. 1988.*
- [Motro-1990] A. Motro. FLEX: A Tolerant and Cooperative User Interface to Databases. *IEEE Transactions on Knowledge and Data Engineering. Zv. 2(2), str. 231–246. 1990.*
- [Nelson-1995] D. A. Nelson in B. N. Rossiter. Prototyping a Formal Object-Oriented Database in P/FDM. *V Proceedings of the 7th ERCIM Database Research Group Workshop on Object-Oriented Databases, Lizbona, Portugalska. 1995.*
- [Ng-2001] W. Ng. An Extension of the Relational Data Model to Incorporate Ordered Domains. *ACM Transactions on Database Systems. Zv. 26(3), str. 344–383. 2001.*
- [Nöth-2004] E. Nöth, A. Horndasch, F. Gallwitz in J. Haas. Experiences with Commercial Telephone-based Dialogue Systems. *it – Information Technology. Zv. 46, str. 315–321. 2004.*
- [Paredaens-1992] L. Paredaens. Converting Nested Algebra Expressions into Flat Algebra Expressions. *ACM Transactions on Database Systems. Zv. 17(1), str. 65–93. 1992.*
- [Patel-2004] N. R. Patel in P. C. Bruce. Multiple Linear Regression. *str. 43–54. V Data Mining in Excel: Lecture Notes and Cases. Resampling Stats, Arlington. 2004.*

- [Pirker-1999] *H. Pirker, G. Loderer in H. Trost. Thus Spoke the User to the Wizard. V Proceedings of the 6th European Conference on Speech Communication and Technology, Budimpešta, Madžarska. Zv. 3, str. 1171–1174. 1999.*
- [Plotkin-1983] *G. Plotkin. Pisa Notes (On Domain Theory). University of Edinburgh, Velika Britanija. Dosegljivo na <http://homepages.inf.ed.ac.uk/gdp/publications/Domains.pdf>. 1983.*
- [Poesio-1997] *M. Poesio in D. R. Traum. Conversational actions and discourse situations. Computational Intelligence. Zv. 13(3), str. 309–349. 1997.*
- [Polifroni-1992] *J. Polifroni, L. Hirschman, S. Seneff in V. Zue. Experiments in Evaluating Interactive Spoken Language Systems. V Proceedings of DARPA Speech and Natural Language Workshop, Harriman, ZDA. str. 28–33. 1992.*
- [Power-1979] *R. Power. The organisation of purposeful dialogues. Linguistics. Zv. 17, str. 107–152. 1979.*
- [Price-1992] *P. Price, L. Hirschman, E. Shriberg in E. Wade. Subject-based Evaluation Measures for Interactive Spoken Language Systems. V Proceedings of the DARPA Speech and Natural Language Workshop, Harriman, ZDA. str. 34–39. 1992.*
- [Ramakrishnan-1995] *R. Ramakrishnan in J. D. Ullman. A Survey of Research on Deductive Database Systems. Journal of Logic Programming. Zv. 23(2), str. 125–149. 1995.*
- [Ramakrishnan-2003] *R. Ramakrishnan in J. Gehrke. Database Management Systems. McGraw-Hill, Boston. 2003.*
- [Raymond-1996] *D. Raymond. Partial Order Databases. Doktorska disertacija. University of Waterloo, Department of Computer Science. Kanada. 1996.*
- [Reiter-1978] *R. Reiter. On Closed World Data Bases. str. 55–76. V Logic and Data Bases. Uredila H. Gallaire in J. Minker. Plenum, New York. 1978.*
- [Reiter-1984] *R. Reiter. Towards a Logical Reconstruction of Relational Database Theory. str. 163–189. V On Conceptual Modelling. Uredili M. Brodie, J. Mylopoulos in J. Schmit. Springer Verlag, Berlin – Heidelberg – New York. 1984.*
- [Rossiter-1994] *B. N. Rossiter, D. A. Nelson in M. A. Heather. The Categorical Data Model as a Formalism for Object-Relational Databases. Computing Science Technical Report no. 505, University of Newcastle upon Tyne, Newcastle, Velika Britanija. 1994.*

- [*Sanders-1993*] *M. S. Sanders* in *E. J. McCormick*. Human Factors in Engineering and Design, Seventh Edition. *McGraw-Hill, New York*. 1993.
- [*Schek-1986*] *H.-J. Schek* in *M. H. Scholl*. The Relational Model with Relation-Valued Attributes. *Information Systems. Zv. 11(2)*, str. 137–147. 1986.
- [*Seber-1977*] *G. A. F. Seber*. Linear Regression Analysis. *John Wiley & Sons, New York*. 1977.
- [*Sendov-2001*] *B1. Sendov*. Hausdorff Geometry of Polynomials. *East Journal of Approximations. Zv. 7*, str. 123–178. 2001.
- [*Shriberg-1992*] *E. Shriberg, E. Wade* in *P. Price*. Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction. *V Proceedings of the DARPA Speech and Natural Language Workshop, Harri-man, ZDA*. str. 49–54. 1992.
- [*Smith-1997*] *R. W. Smith* in *S. A. Gordon*. Effects of Variable Initiative on Linguistic Behavior in Human-Computer Spoken Natural Language Dialogue. *Computational Linguistics. Zv. 23(1)*, str. 141–168. 1997.
- [*Sparck-Jones-1996*] *K. Sparck-Jones* in *R. Galliers*. Evaluating Natural Language Processing Systems: An Analysis and Review (Lecture Notes in Artificial Intelligence 1083). *Springer Verlag, Berlin – Heidelberg – New York*. Dosegljivo na <http://citeseer.ist.psu.edu/galliers93evaluating.html>. 1996.
- [*Stallard-2000*] *D. Stallard*. Talk'n'Travel: A Conversational System for Air Travel Planning. *V Proceedings of the Association for Computational Linguistics 6th Applied Natural Language Processing Conference, Seattle, ZDA*. str. 68–75. 2000.
- [*Stolcke-2000*] *A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. Van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin* in *M. Meteer*. Dialogue Act Modelling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics. Zv. 26(3)*, str. 339–373. 2000.
- [*Sturm-1999*] *J. Sturm, E. den Os* in *L. Boves*. Dialogue management in the Dutch ARISE train timetable information system. *V Proceedings of the 6th European Conference on Speech Communication and Technology, Budimpešta, Madžarska*. str. 1419–1422. 1999.
- [*Tabachnick-1996*] *B. G. Tabachnick* in *L. S. Fidell*. Using Multivariate Statistics, Third Edition. *Harper Collins, New York*. 1996.

- [Takacs-1998] *B. Takacs in H. Wechsler. Face Identification Using the Hausdorff Metric. V Proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition, Nara, Japonska. 1998.*
- [Thomas-1986] *S. J. Thomas in P. C. Fischer. Nested Relational Structures. str. 269–307. V Advances in Computing Research III, The Theory of Databases. Uredil P. C. Kanellakis. JAI Press, Greenwich. 1986.*
- [Troelstra-www] *A. S. Troelstra. History of constructivism in the 20th century. Dosegljivo na <http://citeseer.nj.nec.com/96765.html>.*
- [Tsichritzis-1982] *D. C. Tsichritzis in F. H. Lochovsky. Data Models. Prentice-Hall, Englewood Cliffs, New Jersey. 1982.*
- [Ullman-1988] *J. D. Ullman. Principles of Database and Knowledge-Base Systems, Volume I. Computer Science Press, Inc., Rockville, Maryland. 1988.*
- [Ullman-1989] *J. D. Ullman. Principles of Database and Knowledge-Base Systems, Volume II: The New Technologies. Computer Science Press, Inc., Rockville, Maryland. 1989.*
- [Vesnicer-2003] *B. Vesnicer. Umetno tvorjenje govora z uporabo prikritih Markovovih modelov. Magistrsko delo. Fakulteta za elektrotehniko, Univerza v Ljubljani. Ljubljana. 2003.*
- [Vidav-2003] *I. Vidav. Algebra. Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana. 2003.*
- [Vrabc-1993] *J. Vrabc. Metrični prostori. Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana. 1993.*
- [Walker-1997a] *M. A. Walker, D. Litman, C. A. Kamm in A. Abella. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. V Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, Madrid, Španija. str. 271–280. 1997.*
- [Walker-1997b] *M. A. Walker, D. Hindle, J. Fromer, G. Di Fabbrizio in C. Mestel. Evaluating Competing Agent Strategies for a Voice Email Agent. V Proceedings of the 5th European Conference on Speech Communication and Technology, Rodos, Grčija. str. 2219–2222. 1997.*
- [Walker-1998] *M. A. Walker, D. J. Litman, C. A. Kamm in A. Abella. Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies. Computer Speech and Language. Zv. 12(3), str. 317–347. 1998.*

- [Walker-1999] M. A. Walker, J. Boland in C. Kamm. The Utility of Elapsed Time as a Usability Metric for Spoken Dialogue Systems. V *Proceedings of the Automatic Speech Recognition and Understanding Workshop, Keystone, ZDA. Zv. 1, str. 317–320. 1999.*
- [Walker-2000] M. A. Walker. An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email. *Journal of Artificial Intelligence Research. Zv. 12, str. 387–416. 2000.*
- [Walker-2001a] M. A. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard in S. Whittaker. DARPA Communicator Dialog Travel Planning Systems: The June 2000 data Collection. V *Proceedings of the 7th European Conference on Speech Communication and Technology, Aalborg, Danska. str. 1371–1375. 2001.*
- [Walker-2001b] M. A. Walker, R. Passonneau in J. E. Boland. Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems. V *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, Francija. str. 515–522. 2001.*
- [Whittaker-1989] S. Whittaker in P. Stenton. User Studies and the Design of Natural Language Systems. V *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics, Manchester, Anglija. str. 116–123. 1989.*
- [Whittaker-2002] S. Whittaker, M. Walker in J. Moore. Fish or Fowl: A Wizard of Oz Evaluation of Dialogue Strategies in the Restaurant Domain. V *Proceedings of the 3rd Language Resources and Evaluation Conference, Las Palmas, Španija. 2002.*
- [Woods-1972] W. A. Woods, R. M. Kaplan in B. N. Webber. The Lunar sciences Natural Language Information System: Final Report. *BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachusetts. 1972.*
- [Zoltan-Ford-1991] E. Zoltan-Ford. How to Get People to Say and Type What Computers Can Understand. *Journal of Man-Machine Studies. Zv. 34, str. 527–547. 1991.*
- [Zue-2000] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen in L. Hetherington. JUPITER: A Telephone Based Conversational Interface for Weather Information. *IEEE Transactions on Speech and Audio Processing. Zv. 8(1), str. 85–96. 2000.*

[Žibert-2003]

J. Žibert, S. Martinčić-Ipšić, M. Hajdinjak, I. Ipšić in F. Mihelič. Development of a Bilingual Spoken Dialog System for Weather Information Retrieval. V Proceedings of the 8th European Conference on Speech Communication and Technology, Ženeva, Švica. str. 1917–1920. 2003.

Izvirni prispevki k znanosti

Disertacija vsebuje naslednje pomembnejše izvirne prispevke k znanosti:

- Analiza in interpretacija postopka vrednotenja učinkovitosti sistemov za dialog z ogrođjem PARADISE (poglavje 3).

Analizirala sem teoretično zasnovo ogrođja PARADISE, opozorila na njene pomankljivosti ter izpostavila nekatere sporne točke ogrođja PARADISE.

Pokazala in argumentirala sem pravilno uporabo normalizacije v modelu učinkovitosti, ki ga zajema ogrođje PARADISE (razdelek 3.3.1). Dosedanje raziskave namreč poročajo le o normalizaciji neodvisnih spremenljivk funkcije učinkovitosti, ne pa tudi o nujnosti normalizacije odvisne spremenljivke, ki je predpogoj numerične stabilnosti. Za napovedovanje zadovoljstva uporabnikov pa s stališča natančnosti ocen ni ugodno uporabljati normalizirane vrednosti, ki je rezultat funkcije učinkovitosti. Zaradi zmanjšanja napak je normalizirane vrednosti treba (s transformacijo, ki je inverzna normalizaciji) pretvoriti nazaj na začetni interval.

Izpostavila sem verjetno najšibkejšo točko ogrođja PARADISE, tj. vprašalnik, s katerim merimo zadovoljstvo uporabnikov (razdelek 3.3.4). Ta vprašalnik ne temelji niti na teoriji niti na ustreznih empiričnih raziskavah, in zato ne more šteti za veljaven psihometrični instrument. Predlagala sem način merjenja zadovoljstva uporabnikov – uteženo vsoto ali uteženo srednjo vrednost ocen, pridobljenih z ustreznim (tj. zanesljivim, veljavnim, objektivnim in diskriminativnim) vprašalnikom. To področje vsekakor zahteva obsežne nadaljnje raziskave.

Študija znanstvenih prispevkov na področju vrednotenja učinkovitosti sistemov za dialog je pokazala veliko izstopanje parametrov, ki merijo učinkovitost razpoznavanja govora (razdelek 3.3.5). To me je pripeljalo do sklepa, da bo vrednotenje, ki je usmerjeno v učinkovitost posameznih modulov sistema za dialog, realnejše, nazornejše in bolj informativno, če učinke samodejnega razpoznavanja govora iz sistema odstranimo. To lahko npr. naredimo tako, da izvajamo eksperiment Čarovnik iz Oza, v katerem simuliramo popolno razpoznavanje govora.

- Definicija in vloga parametrov podatkovne zbirke pri delovanju in vrednotenju samodejnih sistemov za podajanje informacij (poglavje 4).

Definirala in uporabila sem t. i. parametre podatkovne zbirke (razdelek 4.2), ki merijo število in delež potez, s katerimi sistem uporabniku poda informacije, ki jih najde v podatkovni zbirki, število in delež potez, ki uporabnika usmerjajo k izbiri relevantnih, dosegljivih podatkov, ter število in delež potez, s katerimi sistem uporabniku sporoča, da zahtevanega podatka nima in ga pri tem ne usmerja k

izbiri relevantnih, dosegljivih podatkov. V eksperimentih se je izkazalo, da so taki parametri, ki izražajo velikost podatkovne zbirke in stopnjo sodelujočega odgovarjanja, v sistemih za podajanje informacij ključnega pomena in zato pri vrednotenju učinkovitosti nepogrešljivi.

- Predstavitev znanja v sodelujočih samodejnih sistemih za dialog s kategornimi relacijskimi podatkovnimi modeli (poglavje 9).

Relacijsko algebro kot najbolj znan model relacijskih podatkovnih zbirk, definiran v kategoriji množic, sem (matematično) naravno posplošila na teorijo kategorij (razdelek 9.2). Osnovnim objektom relacijske algebre in nad njimi definiranim operacijam (unija, razlika, kartezični produkt, projekcija in izbira) sem poiskala ustrezne kategorne konstrukte. Ugotovila sem, da morajo kategorije, ki bodo primerna osnova kategornega modela relacijskih podatkovnih zbirk, vsebovati natančen začetni objekt, ki je kategorna posplošitev pojma prazne podmnožice, končni objekt, ki je kategorna posplošitev pojma podmnožice z enim samim elementom, končne kategorne produkte, slike ter unije, preseke in razlike tistih podobjektov, s katerimi modeliramo poizvedbe.

Predstavila sem primer tovrstne kategorije, imenovane kategorija podobnosti (razdelek 9.3), in njeni podkategoriji, imenovani kategorija urejenih množic in kategorija kompaktnih metričnih prostorov. Posplošitev relacijske algebre na močan formalizem, ki ga nudi teorija kategorij, je pripeljal do naravne implementacije relaksacije v posplošene relacijske podatkovne sisteme. Za objekte kategorije podobnosti sem pokazala, da jih (tako kot relacije, ki so osnovni objekti relacijske algebre) lahko udejanjimo z relacijskimi tabelami.

Ljubljana, 15. 6. 2006

Izjava o avtorstvu

Podpisana Melita Hajdinjak, rojena 20. junija 1977 v Grazu v Avstriji, izjavljam, da sem avtorica doktorske disertacije z naslovom

PREDSTAVITEV ZNANJA IN VREDNOTENJE UČINKOVITOSTI
SODELUJOČIH SAMODEJNIH SISTEMOV ZA DIALOG.

Melita Hajdinjak