

Predstavljanje znanja zasnovano na integraciji ontologija i Bayesovih mreža

Prcela, Marin

Doctoral thesis / Disertacija

2010

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:192256>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-10**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Marin Prcela

**PREDSTAVLJANJE ZNANJA
ZASNOVANO NA INTEGRACIJI
ONTOLOGIJA I BAYESOVIH MREŽA**

DOKTORSKA DISERTACIJA

Zagreb, 2010.

Doktorska disertacija je izrađena na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva te na Zavodu za elektroniku Instituta „Ruđer Bošković“ u Zagrebu.

Mentor: Prof.dr.sc. Nikola Bogunović

Dokrotska disertacija ima 150 stranica.

Disertacija br.:

Povjerenstvo za ocjenu doktorske disertacije:

- 1) Dr.sc. Bojana Dalbelo Bašić, redovita profesorica
Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva
- 2) Dr.sc. Nikola Bogunović, redoviti profesor
Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva
- 3) Dr.sc. Dragan Gamberger, znanstveni savjetnik
Institut Ruđer Bošković, Zagreb

Povjerenstvo za obranu doktorske disertacije:

- 1) Dr.sc. Bojana Dalbelo Bašić, redovita profesorica
Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva
- 2) Dr.sc. Nikola Bogunović, redoviti profesor
Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva
- 3) Dr.sc. Dragan Gamberger, znanstveni savjetnik
Institut Ruđer Bošković, Zagreb
- 4) Dr.sc. Ratko Magjarević, redoviti profesor
Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva
- 5) Dr.sc. Tomislav Šmuc, viši znanstveni suradnik
Institut Ruđer Bošković, Zagreb

Sadržaj

1	UVOD.....	1
1.1	SUSTAVI ZA POTPORU PRI ODLUČIVANJU U MEDICINI	3
1.2	HEARTFAID PROJEKT.....	5
1.3	ONTOLOGIJE I BAYESOVE MREŽE	6
1.4	CILJEVI INTEGRACIJE ONTOLOGIJA I BAYESOVIH MREŽA	7
2	SRODNA ISTRAŽIVANJA	11
2.1	POSTOJEĆI SUSTAVI ZA POTPORU PRI ODLUČIVANJU	11
2.2	POSTOJEĆE ONTOLOGIJE U DOMENI SRČANOG ZATAJENJA.....	13
2.3	INTEGRACIJA ONTOLOGIJA I BAYESOVIH MREŽA	14
2.4	UKLJUČIVANJE ČINJENIČNOG ZNANJA.....	16
3	TEORIJSKE OSNOVICE RASUĐIVANJA.....	18
3.1	DESKRIPTIVNA LOGIKA.....	18
3.1.1	<i>Jezični oblici.....</i>	<i>18</i>
3.1.2	<i>Svojstveni jezik \mathcal{AL}.....</i>	<i>19</i>
3.1.3	<i>Interpretacija.....</i>	<i>20</i>
3.1.4	<i>Porodica \mathcal{AL} jezika.....</i>	<i>23</i>
3.1.5	<i>Komponente baze znanja.....</i>	<i>25</i>
3.1.6	<i>TBox.....</i>	<i>26</i>
3.1.7	<i>Oblici rasuđivanja u TBox-u.....</i>	<i>27</i>
3.1.8	<i>ABox.....</i>	<i>29</i>
3.1.9	<i>Oblici rasuđivanja u ABox-u.....</i>	<i>30</i>
3.1.10	<i>Posredno rasuđivanje</i>	<i>31</i>
3.1.11	<i>Semantike otvorenog i zatvorenog svijeta</i>	<i>33</i>
3.1.12	<i>Deskriptivna logika i predikatna logika prvog reda</i>	<i>34</i>
3.1.13	<i>Tableau postupak rasuđivanja.....</i>	<i>35</i>
3.1.14	<i>Postojeći sustavi za rasuđivanje.....</i>	<i>39</i>

3.2	BAYESOVE MREŽE	40
3.2.1	<i>Računanje a priori očekivanja čvorova</i>	41
3.2.2	<i>Propagacija dokaza unaprijed kroz mrežu</i>	42
3.2.3	<i>Propagacija dokaza unatrag kroz mrežu</i>	42
3.2.4	<i>Propagacija unaprijed i unatrag</i>	44
3.2.5	<i>Međusobna zavisnost čvorova</i>	44
3.2.6	<i>Algoritam d-separacije</i>	46
3.2.7	<i>Algoritmi rasuđivanja</i>	46
3.2.8	<i>Uzorkovanje</i>	48
4	INTEGRACIJA ONTOLOGIJA I BAYESOVIH MREŽA	52
4.1	ONTOLOGIJE U SUSTAVU ZA POTPORU PRI ODLUČIVANJU	52
4.1.1	<i>Primjer problema</i>	52
4.1.2	<i>Opis domene pomoću ontologije</i>	53
4.1.3	<i>Opis pacijenta u ontologiji</i>	55
4.1.4	<i>Opis znanja logičkim izrazima</i>	56
4.2	MOTIVACIJA ZA INTEGRACIJU SA BAYESOVOM MREŽOM	59
4.2.1	<i>Ciljevi integracije</i>	60
4.2.2	<i>Očekivanja ciljnih koncepata</i>	60
4.2.3	<i>Slijed odluka temeljen na informacijskoj dobiti</i>	61
4.2.4	<i>Informacijska dobit grupiranih pretraga</i>	62
4.3	IZGRADNJA I UPORABA BAYESOVE MREŽE	66
4.3.1	<i>Učenje Bayesove mreže iz skupa podataka</i>	69
4.4	POVEZIVANJE FORMALIZAMA.....	73
4.4.1	<i>Ostvarivanje semantičke veze</i>	73
4.4.2	<i>Integracija logičkih izraza u Bayesovu mrežu</i>	74
4.4.3	<i>Pretpostavke pri integraciji</i>	78
4.5	RASUĐIVANJE	79
4.5.1	<i>Otkrivanje očekivanja ishoda</i>	79

4.5.2	<i>Otkrivanje informativnosti čimbenika</i>	80
4.5.3	<i>Informacijska dobit grupiranih pretraga</i>	82
4.5.4	<i>Otkrivanje informacijske dobiti egzaktnim postupcima</i>	85
4.5.5	<i>Otkrivanje informacijske dobiti uzorkovanjem</i>	88
4.5.6	<i>Podешavanje veličine skupa uzoraka</i>	90
5	UKLJUČIVANJE ČINJENIČNOG ZNANJA	92
5.1	ČINJENIČNO ZNANJE U MEDICINSKOM OKRUŽENJU	92
5.1.1	<i>Medicinski podaci i relacijske baze podataka</i>	93
5.1.2	<i>Uključivanje činjeničnog znanja u ontologiju</i>	95
5.1.3	<i>Okruženje za dohvata činjeničnog znanja</i>	96
5.2	OWLMAPPER – FORMALIZAM ZA PRESLIKAVANJE ČINJENIČNOG ZNANJA.....	97
5.2.1	<i>Slijed preslikavanja</i>	97
5.2.2	<i>Reference na ontologiju</i>	98
5.2.3	<i>Razmještanje vrijednosti iz baze podataka</i>	99
5.2.4	<i>Rječnik</i>	100
5.2.5	<i>Promjenjivi parametri preslikavanja</i>	101
5.2.6	<i>Program: niz preslikavanja</i>	102
5.2.7	<i>Konfiguracija sustava</i>	102
5.3	PRIMJER UKLJUČIVANJA ČINJENIČNOG ZNANJA	102
5.3.1	<i>Korisničko sučelje HEARTFAID platforme</i>	102
5.3.2	<i>Relacijski model: HEARTFAID eCRF</i>	103
5.3.3	<i>Primjeri preslikavanja</i>	104
5.4	DISKRETIZACIJA PODATKOVNIH VRIJEDNOSTI	108
6	VREDNOVANJE FUNKCIONALNIH ZNAČAJKI	112
6.1	KORISNIČKO SUČELJE: PRIMJER KORIŠTENJA SUSTAVA ZA PODRŠKU PRI ODLUČIVANJU.....	113
6.2	OKRUŽENJE SUSTAVA	118
6.3	VREMENA IZVOĐENJA.....	119
6.3.1	<i>Vrijeme generiranja uzoraka</i>	120

6.3.2	<i>Vrijeme računanja očekivanja i informativnosti</i>	121
6.3.3	<i>Vrijeme učitavanja ontologije</i>	122
6.4	POGREŠKA PRI RASUĐIVANJU	123
6.4.1	<i>Pogreška i broj uzoraka</i>	124
6.4.2	<i>Pogreška i veličina grupirane pretrage</i>	126
6.4.3	<i>Pogreška pri rasuđivanju uz izbačene ekstremne vrijednosti</i>	127
6.5	POBOLJŠANJE PERFORMANSI	129
7	ZAKLJUČAK	130
	LITERATURA	133
	ŽIVOTOPIS	142
	BIOGRAPHY	143

1 Uvod

Sustav za potporu pri odlučivanju (engl. *decision support system*) podrazumijeva računalni sustav koji ima mogućnost pružanja podrške pri donošenju odluka u problemima visoke kompleksnosti na temelju *znanja* pohranjenog u sustavu i na temelju sveobuhvatnog pregleda situacije (značajki promatranog problema).

Potreba za podrškom u odlučivanju danas je prisutna u gotovo svim područjima ljudskog djelovanja. Općenito se može reći da su sustavi za podršku pri odlučivanju primjenjivi u situacijama koje imaju neke od sljedećih značajki:

- Visoka kompleksnost problema – problemi u kojima je potrebno promatrati mnogo čimbenika (varijabli) istovremeno te na temelju sveukupne situacije donijeti odluku.
- Potrebno je ekspertno znanje – mnogi problemi zahtjevaju poznavanje vrlo specifičnog znanja u promatranj domeni.
- Potrebno je donositi odluke u realnom vremenu – često je potrebno odluke donositi u vrlo kratkom vremenskom periodu jer nije moguće uložiti vrijeme u detaljnije analize.
- Nije uvijek dostupan potpun uvid u značajke problema – često je potrebno donositi odluke na temelju nepotpune informacije u sustavu. U takvim situacijama odlučuje se na temelju pretpostavki i/ili statistike a često čak i na temelju subjektivnog osjećaja eksperta u domeni.

Ekspert je osoba koja ima superiorne rezultate pri rješavanju konkretnih instanci problema u svojoj domeni. Primjerice, liječnik koji je specijalizirao kardiologiju je ekspert ako ima dobre rezultate pri liječenju pacijenata sa kardiološkim problemima. Pri tome uopće nije ključna sama razina stručne spreme ili starosna dob promatrane osobe. Važno je uočiti da je moguće da više eksperata za identičan problem može imati više međusobno različitih (čak i konfliktnih) mišljenja. Dakle, nije neobično da ponekad pri donošenju odluke eksperti i pogriješe.

Sustavi za potporu pri odlučivanju koji pokušavaju rješavati probleme za koje je u realnom svijetu potrebno ekspertno znanje nazivaju se *ekspertni sustavi* (engl. *expert*

systems). Pri tome se raznim računalnim postupcima pokušava oponašati proces donošenja odluka kojeg obavlja ekspert pri rješavanju konkretnih problema¹.

Jasno je da ključnu ulogu u ekspertnom sustavu igra upravo ekspertno znanje. Stoga se pri izgradnji sustava za potporu pri odlučivanju (ili ekspertnog sustava) mogu izdvojiti tri temeljne zadaće:

- Sakupljanje znanja (engl. *knowledge acquisition*),
- Predstavljanje znanja (engl. *knowledge representation*) i
- Korištenje znanja u trenutku potrebe.

Iako je iz formalne perspektive razumljivo konceptualno razdvajanje ovih pojmova, u praktičnim se realizacijama oni često isprepliću te je teško razlučiti jasnu granicu među njima. U ovoj disertaciji naglasak je stavljen na samu metodologiju *predstavljanja* znanja, ali se u određenoj mjeri dotiču i preostale dvije navedene točke.

Sakupljanje znanja [29], [36] podrazumijeva proces prikupljanja, analize, transformacije i organizacije znanja kako bi ono postalo prikladno za prikaz i korištenje u računalnim sustavima. Znanje je moguće prikupljati:

- iz raznovrsnih dostupnih medija (članci, enciklopedije, smjernice, ...) i/ili
- razgovorima s ekspertima u domeni.

Izvori znanja razlikuju se u kvaliteti i obliku znanja kojeg pružaju, ali i u samoj složenosti postupka sakupljanja znanja.

Smjernice (engl. *guidelines*) predstavljaju naprikladniji medij za sakupljanje znanja jer sadrže znanje koje je u manjoj ili većoj mjeri formalizirano i eksplicitno izrečeno. Međutim, znanje je u smjernicama pisano od eksperata za eksperte, što zapravo znači da smjernice imaju visoku razinu znanja koje se među ekspertima u domeni podrazumijeva. Stoga primjerice smjernice za liječenje srčanog zatajenja nisu prikladne za čitanje od strane ne-medicinskog osoblja. Nadalje, smjernice često sadrže samo prijedloge koji se moraju interpretirati ovisno o kontekstu (koji je često nejasan), što ne pruža dovoljnu podlogu za eksplicitnu formalizaciju u računalnom sustavu. Kako bi se znanje iz

¹ Danas se pri izradi ekspertnih sustava u praktičnim implementacijama često kao formalizam za predstavljanje znanja koriste produkcijska pravila, pa se stoga termin *ekspertni sustav* često koristi i kao sinonim za produkcijska pravila. U općem slučaju, ekspertni sustav može biti realiziran i na bilo koji drugi način.

smjernica iskoristilo nužno ga je obogatiti implicitnim (skrivenim, intuitivnim, ne-eksplicitnim) ekspertnim znanjem [11].

Znanje koje posjeduju eksperti se može ugrubo podijeliti u dvije kategorije: eksplicitno i implicitno [8]. *Eksplicitno* znanje je znanje koje se može artikulirati prirodnim jezikom. Ono se izražava riječima, matematičkim i logičkim izrazima, grafovima i slično. *Implicitno* znanje je znanje koje je teško izreći formalnim jezikom. Ono je osobna karakteristika svakog pojedinog eksperta i temelji se na individualnom iskustvu. Implicitno znanje u sebi uključuje subjektivna uvjerenja, subjektivni sustav vrijednosti i subjektivnu percepciju događaja od strane eksperta.

Dok se eksplicitno znanje relativno lako sakuplja za to predviđenim tehnikama, implicitno znanje predstavlja mnogo složeniji problem. Kako bi se postigao kvalitetan sustav za potporu pri odlučivanju potrebno je obuhvatiti obje kategorije znanja u maksimalnom mogućem obujmu.

Predstavljanje znanja podrazumijeva formalizaciju sakupljenog znanja u obliku u kojem je ono pogodno za uporabu. Danas postoji velik broj različitih formalizama za predstavljanje znanja sa različitim razinama izražajnosti. Pri tome se redovito javlja problem kompromisa između jednostavnosti prikaza znanja (jasnoće) i izražajnosti formalizma.

Korištenje znanja podrazumijeva uporabu sakupljenog i predstavljenog znanja u realnom okruženju. Pri tome se stavlja naglasak na mogućnost *rasuđivanja*, tj. na mogućnost izvođenja implicitnog (novog, ne-izrečenog) znanja iz onog koje je eksplicitno izrečeno (postojeće). Kako bi bilo moguće koristiti znanje u realnom okruženju ključno je sustavu za potporu pri odlučivanju omogućiti pristup svim potrebnim informacijama koje karakteriziraju promatrani problem kako bi se moglo korištenjem baze znanja rasuđivati na temelju stvarnih i aktualnih značajki problema. Tu se podrazumijevaju razna sučelja na baze podataka, sučelja na raznu instrumentaciju i senzore koji očitavaju trenutne vrijednosti parametara, itd.

1.1 Sustavi za potporu pri odlučivanju u medicini

U medicinskim primjenama sustavi za potporu pri odlučivanju već imaju zapaženu ulogu, a najčešće kao pomoćno sredstvo pri dijagnosticiranju raznih bolesti, pri utvrđivanju potrebne terapije pri liječenju, pri prognostičkim procjenama stanja pacijenta i slično [16], [17].

Sustav za potporu pri odlučivanju u medicinskom okruženju nikada ne može samostalno donositi odluke i izvršavati medicinske akcije. To je posljedica vrlo osjetljive prirode problema (liječenje stvarnih pacijenata) – zbog pitanja *odgovornosti* nad provedenim

medicinskim postupcima konačne se odluke uvijek prepuštaju stvarnoj osobi (liječniku), dok je uloga sustava više savjetodavne naravi.

Medicinsko okruženje karakterizirano je sa svim navedenim značajkama ekspertnog sustava:

- Vrlo je visok stupanj kompleksnosti zbog prisutnosti velikog broja čimbenika kojima se opisuje stanje pacijenta. Uobičajena je praksa da pacijent kroz niz posjeta liječniku obavlja niz različitih mjerenja (dijagnostičkih pretraga) u kojima svaki pojedini izmjereni parametar (ili kombinacija parametara) može imati svoju težinu u procesu donošenja odluke. Jasno je da računalo može imati puno jasniji uvid u sve podatke istovremeno nego što to može imati čovjek (ekspert).
- Potrebno je ekspertno znanje. Upravo je medicinska znanost karakteristična po tome da obuhvaća velik broj grana i podgrana (stoga i velik broj različitih specijalizacija i podspecijalizacija). Nekada je za donošenje ispravne odluke dovoljno znati jedno jedino pravilo koje je izrazito specifično u promatranj domeni. Takvo je znanje dostupno samo ekspertima u domeni, pa je stoga izrazito korisno formalizirati ga i po mogućnosti automatizirati njegovu primjenu.
- Potrebno je donostiti odluke u realnom vremenu. U mnogim situacijama u medicini vrijeme je ključan faktor. Primjerice, od iznimne je važnosti početi sa ispravnim tretmanom bolesti što je ranije moguće, pa stoga često ne postoji mogućnost dugotrajne i iscrpne analize postojećih podataka o pacijentu.
- Nije uvijek dostupna potpuna slika o problemu. Često nije moguće, čak i uz obavljene sve moguće medicinske pretrage, sa apsolutnom sigurnošću utvrditi dijagnozu nekom konkretnom pacijentu. U takvim slučajevima korisno je voditi se prema ranijim slučajevima bolesti (istraživanja, statistike, prediktivni modeli...).

Medicinska domena karakterizirana je i velikom dozom implicitnog znanja. Znanje medicinskog eksperta se pri liječenju pacijenata često reflektira kroz njegovo iskustvo, njegovu subjektivnu percepciju situacije u kojoj se pacijent nalazi, itd. Stoga je proces sakupljanja ekspertnog znanja u medicinskoj domeni neobično težak i mukotrpan proces. Poteškoća se javlja i zbog velikog komunikacijskog jaza između medicinskih eksperata (koji posjeduju znanje) i tehničkog osoblja (koji znanje formaliziraju u računalno prihvatljiv oblik).

Ova je poteškoća izravno vezana na probleme verifikacije i validacije sustava [47]. U sustavima za potporu pri odlučivanju *verifikacija* podrazumijeva proces kojim se utvrđuje da li sakupljeno medicinsko znanje uistinu odražava realnu sliku promatranog problema (ocjena kvalitete *sakupljanja znanja*). Pojam *validacija* podrazumijeva proces utvrđivanja u kojoj mjeri znanje koje je formalno predstavljeno u sustavu odražava sakupljeno znanje

(ocjena kvalitete *predstavljanja znanja*). Oba problema izrazito su složena i zahtjevaju visok angažman i medicinskih eksperata i tehničkog osoblja.

Na koncu, poteškoća koja je također prisutna u praktičnoj primjeni sustava jest njegova adaptacija u postojeću rutinu u medicinskoj praksi. Mnoge su praktične implementacije sustava za potporu pri odlučivanju propale u fazi primjene upravo zbog neuspješne adaptacije sustava u stvarno okruženje (problemi u edukaciji medicinskog osoblja za primjenu sustava, nepovjerenje liječnika prema zaključcima sustava, itd.) [35].

1.2 HEARTFAID projekt

HEARTFAID (2006-2009) [19], [55] je eurposki znanstveno-istraživački projekt kojemu je cilj razvoj medicinske platforme usluga *temeljene na znanju* za potporu pri ranoj dijagnozi srčanog zatajenja i za efikasnije medicinsko/kliničko vođenje pacijenata sa srčanim zatajenjem u poodmakloj životnoj dobi. Projekt obuhvaća jedanaest partnera (medicinskih, tehničkih i istraživačkih) iz šest europskih država. Ključan dio platforme je upravo medicinsko ekspertno znanje koje je potrebno sakupiti, predstaviti i integrirati ga u platformu te ga na pravilan način iskoristiti u trenutku potrebe.

Metodologija predstavljena u ovoj doktorskoj disertaciji je u jednom dijelu uključena u trenutni prototip HEARTFAID platforme. Stoga se kroz disertaciju u praktičnim primjerima često referencira na domenu srčanog zatajenja.

Srčano zatajenje je poremećaj rada ljudskog srca koji vodi do poremećene razine krvi u cijelom tijelu, a naročito u bubrezima i mišićima. Često se kod oboljelih pojavljuju natekline po cijelom tijelu (zbog zadržavanja krvi/tekućine u tijelu) te voda u plućima (zbog slabijeg pumpanja krvi iz pluća). Jedna od najtežih posljedica je i poremećaj ravnoteže u endokrinom sustavu koji remeti razinu hormona u cijelom tijelu što dodatno pridonosi kompleksnosti bolesti. Ovako složena bolest nema preciznu definiciju, ali se najčešće uzima da je srčano zatajenje prisutno kada su prisutni simptomi (bol u prsima, zadihanost, umor, nemoć, oticanje zglobova) uz objektivne dokaze pokazatelja srčane disfunkcije, po mogućnosti pomoću ehografije. Ozbiljnost bolesti se najbolje reflektira u činjenici da oko 50% oboljelih umre u četiri godine, i da oko 80% oboljelih ima barem dvije hospitalizacije godišnje. Najčešće se primjenjuje terapija koja djeluje izravno na brzinu i intenzitet otkucaja srca (što pruža dovoljno krvi u periferiji ali dodatno opterećuje srce), te terapija za suzbijanje simptoma (uklanjanje posljedica bolesti). Kod ekstremno teških slučajeva postoji i mogućnost transplantacije srca. U posljednjim se godinama srčanom zatajenju posvećuje sve više pozornosti, čemu je jasan pokazatelj i sam broj projekata i medicinskih studija posvećen ovom problemu.

1.3 Ontologije i Bayesove mreže

Ontologija (engl. *ontology*) u računarskim znanostima podrazumijeva način predstavljanja znanja kojim se oblikuje promatrana domena imenovanjem koncepata u domeni i formiranjem relacija među njima. Danas su ontologije široko prihvaćene u uporabi u mnogim konkretnim primjenama i predstavljaju svojevrstan standard kod izrade baze znanja u sustavima za potporu pri odlučivanju.

Pojam *ontologija* veoma je širok i predstavlja više konceptualnu ideju nego neki strogo definiran formalizam za izražavanje znanja. Drugim riječima, postoji više različitih formalizama koji predstavljaju znanje na način koji je obuhvaćen pojmom ontologija. Pri tome je ključno da u samom formalizmu postoji mogućnost strukturiranja promatrane domene definicijom koncepata u domeni i relacija među njima.

Deskriptivna logika (engl. *description logic*) samo je jedan od formalizama koji ima mogućnost oblikovanja ontološke strukture. Osnovna mu je karakteristika da ima semantiku zasnovanu na logici što omogućava opis pojmova logičkim izrazima. Pri tome dolazi do izražaja mogućnost izvođenja novog (implicitnog) znanja iz već postojećeg (eksplicitno izrečenog) znanja korištenjem alata za automatsko rasuđivanje.

Danas je deskriptivna logika ponovno u fokusu znanstvenog istraživanja zbog pojave novih jezika i alata za oblikovanje ontologija u deskriptivnoj logici, prvenstveno u domeni semantičkog web-a. Kao posljedica toga, sve više se i u ostalim srodnim područjima istraživanja koriste jezici i alati zasnovani upravo na deskriptivnoj logici. Primjerice, danas je uobičajeno pri izgradnji medicinske baze znanja u sustavu za potporu pri odlučivanju koristiti OWL formalizam (engl. *Web Ontology Language*) i Protégé-OWL sustav za uređivanje ontologija, iako je primarna namjena tih alata i jezika nominalno u nekoj drugoj domeni [30], [43], [57].

S obzirom na semantiku zasnovanu na logici, zaključci dobiveni procesom rasuđivanja nad ontologijom nisu probabilističke (neizrazite) prirode. Kako u realnim primjenama vrlo često postoji potreba za procjenama probabilističke prirode (primjerice, kolika je vjerojatnost neke dijagnoze s obzirom na poznate značajke pacijenta), ontologija sama redovito nije dovoljna za realizaciju kompletne baze znanja u sustavu za potporu pri odlučivanju [9]. Stoga je uloga ontologije u bazi znanja često ograničena na definiranje terminologije i/ili taksonomije, dok je pri rasuđivanju potrebna nadogradnja sustava sa alatima koji su u tom pogledu izražajni.

Bayesova mreža (engl. *Bayesian network*) pruža okosnicu u kojoj se može modelirati međusobna zavisnost svih varijabli u sustavu simultano u jednoj mreži. Koristeći samo jedan model Bayesove mreže moguće je izračunati očekivanja svih nepoznatih varijabli u sustavu u ovisnosti o vrijednostima poznatih varijabli u sustavu. Upravo zbog ove osobitosti Bayesova je mreža veoma prikladna za uporabu u sustavu za potporu pri

odlučivanju. Drugo veoma korisno svojstvo Bayesove mreže jest mogućnost učenja modela mreže iz sakupljenog skupa podataka. Zbog toga je u situacijama kada je skup podataka već dostupan moguće uz relativno male napore izgraditi model Bayesove mreže upotrebljiv u sustavu za potporu pri odlučivanju.

1.4 Ciljevi integracije ontologija i Bayesovih mreža

Deskriptivna logika slijedi *pretpostavku otvorenog svijeta*. Drugim riječima, njena semantika dozvoljava mogućnost da dio informacije u sustavu nedostaje (primjerice, ne zna se koliki je krvni tlak pacijenta jer još nije obavljeno mjerenje). Zbog toga se kod deskriptivne logike procesom rasuđivanja često ne dolazi do konačnih odluka (do konačne dijagnoze); one ostaju nepoznate. Laički govoreći, moglo bi se reći da sustav čeka na obavljanje potrebnih dijagnostičkih pretraga kako bi mogao donijeti konačnu odluku. Dok se mjerenje uistinu i ne dogodi, konačna odluka ostaje nepoznata.

U takvim situacijama bilo bi veoma korisno saznati:

- Koji zaključci su najvjerojatniji s obzirom na trenutno poznate informacije u sustavu, i
- Koja informacija u sustavu vodi ka konačnim zaključcima najefikasnije.

Deskriptivna logika ne pruža mogućnosti ispunjavanja ovih zadataka. Sa uvođenjem Bayesovih mreža u sustav one se mogu relativno jednostavno ispuniti. Stoga se u disertaciji predlaže sustav za predstavljanje znanja koji je utemeljen na integraciji ontologija i Bayesovih mreža.

Pri pronalaženju najvjerojatnijih ishoda i čimbenika koji su u danom trenutku najinformativniji poželjno je iskoristiti oba iskorištena formalizma za predstavljanje znanja u maksimalnom mogućem obujmu. U tom pogledu mogu se prepoznati sljedeće poželjne karakteristike integrirane baze znanja:

- Zaključke je potrebno donositi s obzirom na sve trenutno poznate podatke u sustavu. Primjerice, pri računanju vjerojatnosti pojedinih dijagnoza potrebno je uzeti u obzir sve poznate značajke promatranog pacijenta.
- Varijable veoma često imaju međusobne zavisnosti koje također valja uzeti u obzir. Primjerice, vrlo je dobro poznata činjenica da je krvni tlak koreliran sa starosnom dobi pacijenta. Stoga, ako je poznato da je promatrani pacijent u starijoj životnoj dobi postoji veća vjerojatnost da on ima i povišeni krvni tlak. Ovakve međusobne zavisnosti varijabli su u integriranom sustavu prikaza znanja obuhvaćene Bayesovom mrežom.
- Medicinske pretrage redovito obavljaju više od jednog mjerenja istovremeno, npr. mjerenje krvnog tlaka istovremeno mjeri i sistolički i dijastolički krvni tlak. Dapače, nezamislivo je da se ova dva mjerenja vrše odvojeno. Stoga je pri analizi

informativnosti važno percipirati medicinske pretrage kao nerazdvojive grupe elementarnih medicinskih mjerenja. Ovakvi strukturni opisi koncepata pomoću drugih koncepata u domeni osnovna su značajka znanja predstavljenog ontologijom.

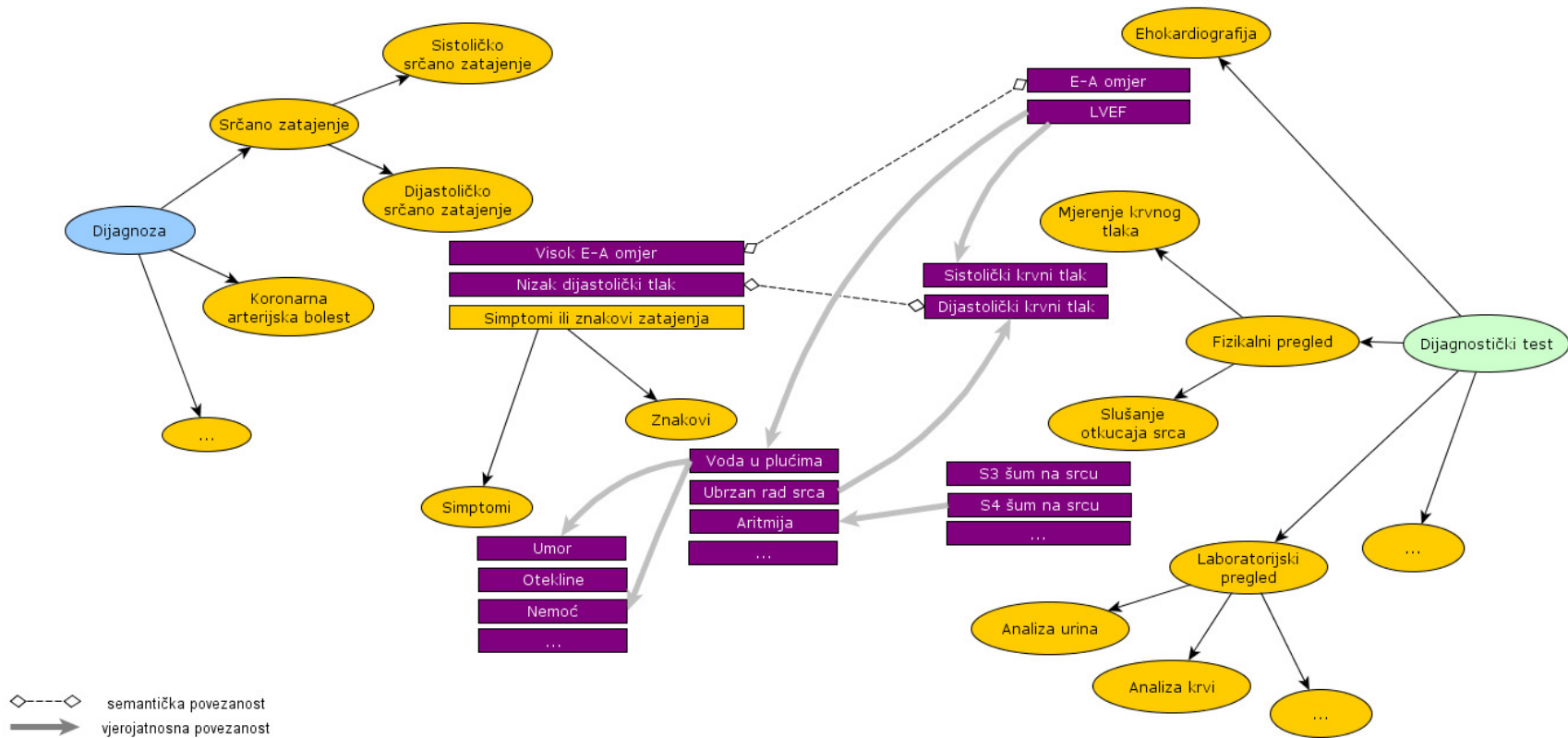
- Neka medicinska mjerenja se mogu provesti na više različitih medicinskih pretraga, npr. brzina otkucaja srca se može mjeriti i na fizičkom pregledu i na EKG-u. U takvim slučajevima informativnost medicinske pretrage ovisi o dobrobiti ostalih mjerenja koje su uključene u pretrage. Ovakvo znanje se također mora uzeti u obzir prilikom rasuđivanja.
- Znanje definirano u ontologiji može poprimiti i mnogo složenije i sveobuhvatnije oblike koje je također potrebno obuhvatiti u integriranom sustavu prikaza znanja (Slika 1.1). Primjerice, dijagnoza dijastoličkog srčanog zatajenja definirana je logičkim izrazima koji se referenciraju na simptome i znakove srčanog zatajenja. Samo simptoma srčanog zatajenja ima mnogo, pa stoga ontologija u sebi sadrži iscrpan skup individua srčanog zatajenja. Svaki pojedini element iz takvog skupa ima pridjeljenu listu medicinskih pretraga pomoću kojih se može otkriti. Stoga analiza informativnosti pri dijagnozi srčanog zatajenja mora predvidjeti i ovakve raznovrsne strukture i veze medicinskih koncepata definiranih u ontologiji kako bi pronašla najinformativniju pretragu koju promatrani pacijent treba izvršiti.

Okvirno gledano, može se reći da je za formalizaciju eksplicitnog ekspertnog znanja ključna ontologija (deskriptivna logika), a da je za formalizaciju neizrazitog znanja ključna Bayesova mreža. S obzirom da je Bayesove mreže moguće trenirati na postojećim medicinskim podacima, integrirani sustav se može promatrati i kao oblik integracije deklarativnog znanja sa znanjem dobivenim računalnim postupcima strojnog učenja.

Kako bi se ostvario integrirani sustav prikaza znanja pomoću ontologija i Bayesovih mreža koji posjeduje gore opisane karakteristike potrebno je:

- Definirati postupak semantičke integracije formalizama,
- Definirati postupak rasuđivanja s ciljem otkrivanja najvjerojatnijeg zaključka u sustavu, i
- Definirati postupak rasuđivanja s ciljem otkrivanja najinformativnijih čimbenika koji vode konačnom zaključku.

Kako bi se što ilustrativnije opisali ovi postupci, u disertaciji se kroz primjere demonstrira izgradnja baze znanja za potrebe dijagnoze srčanog zatajenja. Kako bi se opisao i postupak primjene rezultirajuće baze znanja u stvarnom medicinskom okruženju, analizira se i formalizam koji je razvijen za potrebe uključivanja činjeničnog znanja u sustav.



Slika 1.1. U integriranom sustavu ontološka struktura omogućava strukturnu i logičku definiciju pojmova, dok veze u Bayesovoj mreži realiziraju vjerojatnosne zavisnosti među konceptima. Zahvaljujući prikazanoj strukturi pri utvrđivanju dijagnoze nekom konkretnom pacijentu liječnik može ustvrditi koji je trenutno najinformativniji dijagnostički test kojeg pacijent može obaviti.

Gledano iz perspektive sustava za potporu pri odlučivanju u medicinskim domenama baza znanja koncipirana je na sljedeći način:

- Sustav za potporu pri odlučivanju donosit će eksplicitne zaključke temeljene na ekspertnom znanju kada su sve za to potrebne informacije prisutne u bazi znanja. Na ovaj način se sve važne odluke (dijagnoza, terapija, ...) temelje na postojećim eksplicitnim dokazima.
- Sustav za potporu pri odlučivanju donosit će zaključke probabilističke naravi kada u sustavu nema dovoljno informacije za donošenje konačne odluke. Nadalje, sustav će pronaći akciju (najčešće neki dijagnostički test) koji najefikasnije vodi ka konačnom zaključku. Na ovaj način se nesigurne odluke (sugerirani dijagnostički testovi, prognoze, vjerojatne dijagnoze...) temelje na izračunatim vjerojatnostima pojavljivanja.

U tom je pogledu predložena metodologija predstavljanja znanja usklađena sa općim poimanjem procesa donošenja odluka u medicini.

Organizacija doktorske disertacije je sljedeća. Poglavlje 2 daje pregled aktualnih srodnih istraživanja. Poglavlje 3 opisuje teorijske osnovice predstavljanja znanja i rasuđivanja u deskriptivnoj logici i u Bayesovim mrežama. Poglavlje 4 predstavlja metodologiju integracije ontologija i Bayesovih mreža te pripadajuće postupke rasuđivanja. Poglavlje 5 opisuje problematiku uključivanja činjeničnog znanja u integrirani sustav. Poglavlje 6 analizira neke funkcionalne značajke integriranog sustava kao što su trajanje rasuđivanja i pogreška učinjena pri rasuđivanju.

2 Srodna istraživanja

U ovom poglavlju opisana su istraživanja koja su vezana uz predloženu metodologiju predstavljanja znanja te općenito uz sustave za potporu pri odlučivanju u medicini. U prvom odjeljku načinjen je pregled postojećih sustava za potporu pri odlučivanju uz kraći kritički osvrt. U drugom odjeljku prikazani su neki dosadašnji pokušaji integracije ontologija i Bayesovih mreža u raznim primjenama. Treći odjeljak posvećen je problemu integracije baze znanja u stvarne medicinske informacijske sustave.

2.1 Postojeći sustavi za potporu pri odlučivanju

Danas postoji značajan broj primjera uporabe sustava za potporu pri odlučivanju u medicinskoj praktičnoj primjeni (ATHENA, Hepato Consult, HELP, ISABEL, ILIAD, LISA, PAIRS). S obzirom na važnost i raznolikost medicinske domene, pojavnost ovakvih sustava je još uvijek vrlo skromna. Metodološki oni su redovito bazirani na temeljnim principima općih i dobro poznatih formalizama za predstavljanje znanja:

- produkcijska pravila (engl. *production rules*),
- Bayesova logika i/ili Bayesove mreže,
- dijagrami toka (engl. *workflow*) i planovi,
- rasuđivanje zasnovano na pohranjenim slučajevima (engl. *case-based reasoning*) i
- neizrazita logika.

Neki postojeći sustavi temelje podršku pri odlučivanju već na samoj pohrani podataka u medicinskom informacijskom sustavu te na njihovoj naprednoj analizi i prikazu (HELP). Ovi medicinski sustavi usporedivi su sa sustavima za skladištenje podataka u području poslovne inteligencije (engl. *data warehousing, bussines intelligence*). Iako oni praktično ne koriste nikakve formalne metode prikaza znanja i/ili rasuđivanja, i oni se često u literaturi referenciraju kao sustavi za potporu pri odlučivanju. Iz istog su razloga takvi sustavi iz perspektive inteligentnih sustava manje zanimljivi za analizu.

Slika 2.1 daje pregled postojećih sustava za potporu pri odlučivanju. Sustavi su grupirani s obzirom na korištenu metodologiju prikaza znanja a unutar grupa su poslagani kronološki s obzirom na datum pojavljivanja. Mnogi su sustavi u kasnijim godinama doživjeli i unaprijeđene verzije. Primjerice, sustav APACHE je danas u svojoj četvrtoj verziji koja se pojavila 2005. godine. Sustavi koji su uistinu i zaživjeli u kliničkoj uporabi označeni su bijelom bojom (oni koji nisu označeni su sivom bojom).



Slika 2.1. Neke postojeće implementacije sustava za potporu pri odlučivanju. Bijelom bojom označeni su sustavi koji su upotrebljeni u u kliničkom okruženju.

Na Slici 2.1 može se uočiti nekoliko zanimljivosti. Kod produkcijskih pravila može se uočiti da se pravila koriste gotovo isključivo u specifičnim i uskim medicinskim problemima, što je sasvim očekivano s obzirom na izražajnost produkcijskih pravila kao formalizma.

Bayesove mreže se u pravilu koriste pri diferencijalnoj dijagnozi. To je također sasvim razumljivo jer su Bayesove mreže izuzetno prikladne za slučajeve kada treba pronaći dijagnozu koja najbolje pristaje uočenim simptomima bolesti. Pri dijagnosticiranju se redovito istovremeno posmatraju bolesti koje nominalno pripadaju raznovrsnim medicinskim domenama.

Rasuđivanje zasnovano na pohranjenim slučajevima se temelji na usporedbi promatranog pacijenta sa velikim brojem ranije evidentiranih pacijenata pri čemu se pokušavaju pronaći raniji slučajevi bolesti koji su po svojim značajkama najbliži promatranom pacijentu. Na ovaj način moguće je zaključivati u raznim smjerovima; najčešće su zastupljeni dijagnoza, prognoza i terapija.

Brojni pokušaji razvoja medicinskih sustava za potporu pri odlučivanju rezultirali su brojnim alatima specifično namijenjeni modeliranju smjernica (engl. *guideline modelling tools*, [12], [22], [25], [40], [46], [49], [51]) kao što su GLIF3, Arden syntax, EON, Asbru, PROforma, itd. Većina postojećih alata za modeliranje smjernica koriste ontologije kao formalizme za opis domene a vrlo često i za mnogo zahtjevnije zadaće. Primjerice, kod GLIF3 pristupa ontološki koncepti i relacije služe za opis medicinskih akcija pomoću dijagrama toka (engl. *workflow*). Osnovna ideja kod svih alata za modeliranje smjernica je koncentrirati se na medicinski specifične probleme i riješiti ih na prikladan način. Stoga su ovi alati vrlo prikladni u medicinskoj domeni ali teško funkcioniraju u nekim drugim domenama. Kroz dugi niz godina Arden syntax (temeljen na pravilima) je bio gotovo jedini ovakav alat uistinu korišten u praksi; danas je situacija ipak znatno unapređivala. Na Slici 2.1 istaknuti su neki sustavi utemeljeni na EON i PROforma alatima za modeliranje smjernica.

2.2 Postojeće ontologije u domeni srčanog zatajenja

Danas su dostupne neke inačice medicinskih ontologija, uglavnom terminološke ili taksonomske prirode. Neke od ontologija koje su primjenjive u domeni srčanog zatajenja su:

- Systematized Nomenclature of Medicine – Clinical Terms (Snomed CT),
- Unified Medical Language System (UMLS),
- Generalized Architecture for Languages, Encyclopedias and Nomenclatures in medicine (GALEN),

- Foundational Model of Anatomy (FMA),
- International Classification of Diseases (ICD),
- International Classification of Functioning, Disability and Health (ICF),
- Logical Observation Identifiers Names and Codes (LOINC),
- Medical Subjects Headings (MeSH),
- Medical Dictionary for Regulatory Activities (MedDRA),
- National Drug Code Directory, i
- ONIONS (ONtological Integration Of Naive Sources) Methodology.

Glavne poteškoće u praktičnoj primjeni ovih ontologija posljedica su njihove veličine. Također, u većini slučajeva one su u svojim domenama preopćenite da bi bile korisne u uskim i specifičnim domenama kao što je domena srčanog zatajenja. Ipak, kako bi se na globalnoj razini postigla jednoznačnost u terminološkom su dijelu ontologije srčanog zatajenja (u sklopu HEARTFAID platforme) medicinski koncepti referencirani na UMLS koncepte (Unified Medical Language System) uporabom UMLS CUI identifikatora (engl. *concept unique identifier* [1]).

2.3 Integracija ontologija i Bayesovih mreža

Integracija ontologija i Bayesovih mreža kao metoda predstavljanja znanja u jedan sustav također nije nova paradigma.

U [39] Bayesove mreže se koriste za automatsko prepoznavanje semantičkih relacija između koncepata u dvije različite ontologije, što omogućava njihovu integraciju u jednu funkcionalnu cjelinu. Ovaj je pristup primjenjen u domeni semantičkog weba gdje je problem semantičke povezanosti više ontologija izrazito naglašen.

U [50] se Bayesova mreža koristi za prepoznavanje međuzavisnosti raznih metodologija za povezivanje više ontologija. Na taj način se Bayesovom mrežom pokušava postići optimalna kombinacija raznih metodologija povezivanja ontologija, te na taj način postići kvalitetnija veza između dvije ontologije.

U [13] se znanje prikazano pomoću ontološke strukture koristi za automatizirano modeliranje Bayesove mreže. Kako ontologije opisuju koncepte u domeni i relacije među njima, postoji mogućnost da se to znanje iskoristi pri oblikovanju strukture Bayesove mreže.

U [52] se ontologija koristi za učenje Bayesove mreže i pri stvaranju strukture mreže i pri učenju mrežnih parametara (tj. učenju uvjetnih vjerojatnosti u čvorovima). To se postiže

tako da se uobičajeni proces učenja Bayesove mreže (koristeći postojeći skup podataka) podupre sa shemom bodovanja koja se temelji na ontološkom znanju.

U [33] se konceptualno znanje prikazano ontologijom integrira sa znanjem iz Bayesove mreže koje je dobiveno procesom učenja na postojećem skupu podataka. U [23] se ontologija koristi za rukovanje dokazima (evidencijama) u Bayesovoj mreži. U [53] se ontologija primjenjuje za integraciju heterogenih izvora podataka a Bayesova mreža za donošenje probablističkih odluka.

U [31] predlažu se sofisticirani algoritmi za korištenje probablističke deskriptivne logike koji su ispravni i kompletni. Iako se u pristupu ni u kojem trenutku ne referira eksplicitno na metodologiju Bayesovih mreža, rad je značajan utoliko što se detaljno analizira problem integracije vjerojatnosti i deskriptivne logike.

U [14] prikazan je pokušaj probablističke ekstenzije OWL jezika pomoću Bayesove mreže. To se postiže na način da se OWL jezik proširi konstruktima za prikazivanje vjerojatnosti (stvorena su nova polja unutar XML-a) te je dan skup transformacijskih pravila pomoću kojih se ovako nadopunjena ontologija prevodi u Bayesovu mrežu (kompletna struktura uključujući i tablice distribucije vjerojatnosti u čvorovima). Pri tome valja primjetiti da je ovim transformacijskim pravilima obuhvaćen manji podskup OWL jezika (čak manji i od OWL-Lite verzije). Metodologija prikazana u ovom radu se razlikuje od metodologije opisane u ovoj disertaciji u nekoliko točaka:

- Metodologija predložena u [14] stvara potpuno novu Bayesovu mrežu, dok se u predloženom metodologiji Bayesova mreža uči iz skupa podataka te se nakon toga tek nadopunjava novim čvorovima na temelju ontologije;
- U [14] se ni na koji način ne dotiče pitanje informacijske dobiti dok u predloženoj metodologiji informacijska dobit ima centralnu ulogu.

U medicinskoj domeni, [26] koristi ontologiju za polu-automatsku konstrukciju Bayesove mreže za dijagnosticiranje bolesti. U [54] se alat za modeliranje smjernica koji je utemeljen na medicinskim dijagramima toka (GLIF3) integrira sa vjerojatnostima dobivenim iz Bayesove mreže. Pri tome se dijagram toka koristi za prikaz slijeda medicinskih akcija koji pacijente provodi kroz unaprijed definirane medicinske scenarije. U trenutku kada se neka akcija u grafu treba izvršiti, pomoću Bayesove mreže analizira se utjecaj akcije na cijeli definirani proces. Slično kao i u ovoj disertaciji, uočava se potreba za rukovanjem neizrazitim znanjem u medicinskim sustavima za potporu pri odlučivanju.

Iz ovog kratkog pregleda postojećih metoda integracije ontologija i Bayesovih mreža može se zaključiti da je metodologija predložena u ovoj disertaciji principijelno drugačija od svih dosad postojećih. Krucijalni razlika koja se ističe u predloženoj metodologiji jest upravo korištenje *informacijske dobiti* pri donošenju odluka, dok se svi dosadašnji

pokušaji integracije fokusiraju gotovo isključivo na vjerojatnosti pojedinih tvrdnji. Upravo je zbog tih načelnih razilaženja u motivima i ciljevima integracije formalizama teško (ili nemoguće) napraviti bilo kakvu detaljniju usporedbu (prednosti, nedostaci i slično) predložene metodologije sa već postojećima.

2.4 Uključivanje činjeničnog znanja

Jedan od problema pri inegraciji sustava za potporu pri odlučivanju u stvarno medicinsko (kliničko) okruženje jest uključivanje stvarnih podataka u proces rasuđivanja. Danas se podaci o pacijentima u medicinskim institucijama u velikoj većini slučajeva pohranjuju u relacijskim bazama podataka. Stoga se i problem uključivanja činjeničnog znanja danas u velikoj mjeri svodi na integraciju sustava za odlučivanje sa konkretnom instancom relacijske baze podataka.

Problem preslikavanja podataka iz relacijske baze podataka u ontologiju se najčešće susreće u domeni semantičkog weba gdje postoji potreba za uključivanjem podataka pohranjenih u relacijskim bazama u aktivne i dinamičke stranice semantičkog weba. Ovaj problem se često označava pseudonimom „dubinski web“ (engl. *Deep web*). Istraživanje i razvoj upravo u toj domeni rezultiralo je nizom alata i jezika za integraciju relacijskih baza podataka sa ontologijama.

Od gotovih rješenja (formalizama) za integraciju ontologije i relacijske baze podataka (mahom iz domene semantičkog weba) najčešće se referenciraju:

- D2R MAP i proširene verzija D2R [4],
- R2O [6],
- KAON [48] i
- MAPONTO [2].

U inteligentnim informacijskim sustavima preslikavanja se često obavljaju i ad-hoc softverskim rješenjima (Observer [34], Pixel [20]) koja su oblikovana vrlo specifično prema potrebama i zahtjevima građenih sustava.

Neke postojeće inačice sustava podržavaju i heuristike koje pokušavaju proces preslikavanja učiniti (polu) automatskim, primjerice KAON-Reverse [48] i MAPONTO [2]. Pri tome se moguća preslikavanja generiraju na temelju već postojećih veza (polu automatski) ili na temelju sintaksnih/strukturnih sličnosti (automatski). Dakako, rezultate ovih heuristika uvijek valja dodatno validirati.

Postojeće implementacije mogu se ugrubo podijeliti u tri kategorije:

- I. Opis uključivanja pojedinih podataka iz baze (jedan po jedan podatak) u postojeću ontologiju. Kod ovog pristupa se zahtjeva precizna definicija preslikavanja svakog

pojedinih elementa ontologije pa se (uz više uloženi truda) mogu opisati jako specifična i složena preslikavanja.

- II. Definiranje generalnih principa za preslikavanje kompletne baze podataka u postojeću ontologiju. U ovom slučaju je definiranje preslikavanja jednostavnije, ali je teško (ili nemoguće) dobiti potrebne specifične oblike u ontologiji. Drugim riječima, pristup je često preopćenit.
- III. Stvaranje potpuno nove ontologije na temelju podataka u relacijskoj bazi. Ovaj pristup nije prikladan jer je struktura ontologije unaprijed zadana (definira bazu znanja).

Iz navedenog se može zaključiti da je prva navedena kategorija najbolje pristaje u sustavima za potporu pri odlučivanju. Iz istog razloga može se naslutiti da je i sam postupak uključivanja činjeničnog znanja u medicinskim sustavima za potporu pri odlučivanju složen.

Analiza srodnih istraživanja je pokazala da trenutno ne postoji rješenje koje bi zadovoljilo postavljene zahtjeve na sustav za potporu pri odlučivanju probabilističke naravi temeljen na integraciji ontologija i Bayesovih mreža te rad na ovoj disertaciji ima primjerenu motivaciju i bezuvjetno opravdanje.

3 Teorijske osnovice rasuđivanja

U ovom poglavlju opisuju se teorijske osnovice rasuđivanja u deskriptivnoj logici i u Bayesovim mrežama.

3.1 Deskriptivna logika

Deskriptivna logika (engl. *description logic*) [3] je naziv za porodicu formalizama za predstavljanje znanja u kojima se definiraju relevantni koncepti u domeni, njihova svojstva, relacije među konceptima, individue u domeni i relacije među individuama.

Deskriptivna logika je nastala kao sljedbenik *mreža sa strukturnim nasljeđivanjem* (engl. *structured inheritance networks*) koje su se razvile kao odgovor na glavne nedostatke *semantičkih mreža* (engl. *semantic networks*) i *okvira* (engl. *frames*). Osnovna razlika deskriptivne logike u usporedbi sa njenim prethodnicima jest da posjeduje semantiku koja je zasnovana na logici.

Veoma važno obilježje deskriptivne logike jest da pri rasuđivanju posjeduje svojstvo *terminacije* (odlučljivosti, odredivosti, engl. *decidability*), tj. da daje odgovor na svaki mogući upit u konačnom vremenu – za razliku od predikatne logike prvog reda za koju postoje upiti za koje se ne postiže odgovor u konačnom vremenu. Dakako, svojstvo terminacije ne implicira postizanje odgovora u realnom (ili prihvatljivom) vremenu, pa stoga u deskriptivnoj logici i *složenost* postupka rasuđivanja igra važnu ulogu.

Odlučljivost i *složenost* rasuđivanja uvelike ovisi o odabranoj izražajnosti formalizma – s jedne strane veoma izražajni oblici deskriptivne logike mogu imati veoma veliku složenost izvođenja (čak i do razine neodlučljivosti), a sa druge strane oblici sa slabijom izražajnom moći unatoč vrlo efikasnom izvođenju često ne daju dovoljno visok stupanj izražajnosti da bi se opisao željeni problem. Stoga je veoma važno pri dizajnu baze znanja postići kompromis koji će što bolje odgovarati promatranom problemu. Upravo su odlučljivost i složenost rasuđivanja za razne oblike deskriptivne logike i njena proširenja najčešće teme znanstvenog istraživanja u domeni.

3.1.1 Jezični oblici

Osnovni elementi jezika su *atomički koncepti* i *atomička svojstva*. Oni imaju ulogu imenovanja (npr. atomički koncepti *Pacijent*, *Simptom*, *Bolest* i atomička svojstva *jeStar*, *imaSimptom*, *kontraindicira*). Pomoću atomičkih koncepata i svojstava grade se složeniji

izrazi koristeći skup *jezičnih konstruktora*. Jezična izražajnost (i složenost izvođenja) izravno ovisi o dozvoljenim (odabranim) oblicima jezičnih konstrukata.

U općim primjerima se atomički koncepti po konvenciji označavaju velikim slovima A i B, složeni koncepti se označavaju slovima C i D, dok se svojstva označava slovima R i S.

Instanciranjem koncepata u deskriptivnoj logici nastaju *individue*. Primjerice, individua *marko* može biti *instanca* koncepta PREDAVAČ. Analogno tome, instanciranjem svojstava nastaju *relacije*. Tako npr. koncept OSOBA može posjedovati svojstvo BRAT, čijim se korištenjem može realizirati relacija između individua *marko* i *pero* (te time naznačiti da su oni braća).

Temeljni jezik deskriptivne logike jest *svojstveni jezik* (engl. *attributive language*) koji se skraćeno označava sa \mathcal{AL} . Svojstveni jezik deskriptivne logike definiran je prvi puta u [45] kao minimalni jezik upotrebljiv u praksi.

3.1.2 Svojstveni jezik \mathcal{AL}

Koncepti se u svojstvenom jeziku \mathcal{AL} opisuju slijedećom sintaksom:

C, D ->	A	(atomički koncept)
	\top	(univerzalni koncept)
	\perp	(prazni koncept)
	$\neg A$	(atomička negacija)
	$C \sqcap D$	(presjek)
	$\forall R.C$	(restrikcija na vrijednost)
	$\exists R.\top$	(reducirana uporaba egzistencijskog kvantifikatora)

Primjeri *atomičkih koncepata* su OSOBA, PREDAVAČ, PREDMET, STUDENT. Atomički koncepti imenuju pojmove iz stvarnosti i koriste se kao osnovni građevni elementi u deskriptivnoj logici.

Univerzalni koncept (engl. *top concept*) predstavlja koncept koji obuhvaća sve definirane individue u domeni. Drugim riječima, univerzalni koncept je *generalniji* od svih ostalih koncepata u domeni. *Prazni koncept* (engl. *bottom concept*) predstavlja koncept koji ne obuhvaćati niti jednu individuu u domeni. U interpretaciji skupovima prazni koncept predstavlja prazni skup.

Atomička negacija (engl. *atomic negation*) u interpretacijama predstavlja komplement skupa kojeg predstavlja promatrani atomički koncept. Pri tome formalno gledano u svojstvenom jeziku \mathcal{AL} nije dozvoljeno negirati složene koncepte: jedan znak negacije se smije nalaziti samo ispred jednog atomičkog koncepta (zbog toga i naziv *atomička*).

Međutim, u većini jezičnih oblika deskriptivne logike složena se negacija jednostavnim transformacijama pretvori u oblik sa isključivo atomičkim negacijama.

Presjek dvaju koncepata daje mogućnost definiranja presjeka dva skupa. Na taj se način može definirati da neka individua zadovoljava više koncepata istovremeno (npr. kada je individua istovremeno i predavač i student).

Univerzalni kvantifikator postavlja restrikciju na vrijednost relacije. U gore navedenom primjeru ($\forall R.C$) definira se skup individua koji zadovoljavaju sljedeći uvjet: sve relacije koje izlaze iz individue a nazivaju se R pokazuju na individue koje zadovoljavaju koncept C. Pri tome valja biti oprezan: u ovaj skup spadaju i one individue iz kojih uopće ne izlazi niti jedna relacija R.

Egzistencijski kvantifikator također predstavlja restrikciju na vrijednost relacije. Navedenim izrazom ($\exists R.\top$) definira se skup individua koje zadovoljavaju sljedeći uvjet: barem jedna relacija koja se naziva R izlazi iz individue. U \mathcal{AL} svojstvenom jeziku je upotreba egzistencijskog operatora ograničena samo na univerzalni koncept (\top). Drugim riječima, nije moguće postavljati dodatne uvjete na individue na koje relacija R pokazuje.

Evo nekoliko primejra uporabe jezika \mathcal{AL} .

$OSOBA \sqcap \check{Z}ENSKO$	(osoba ženskog spola)
$OSOBA \sqcap \neg\check{Z}ENSKO$	(osoba koja nije ženskog spola)
$OSOBA \sqcap \exists IMADIJETE.\top$	(osoba koja ima barem jedno dijete)
$OSOBA \sqcap \forall IMADIJETE.\check{Z}ENSKO$	(osoba kojoj su sva djeca ženskog spola)
$OSOBA \sqcap \forall IMADIJETE.\perp$	(osoba koja nema djece)

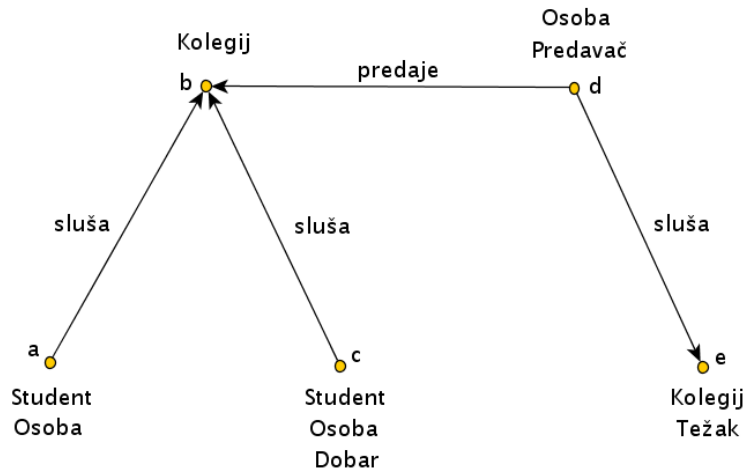
Valja primjetiti da \mathcal{AL} jezik ne dozvoljava uporabu *unije* između dva koncepta što je značajno semantičko ograničenje.

Jezik \mathcal{FL} nastaje ukidanjem atomičke negacije iz \mathcal{AL} jezika. Dodatnim ukidanjem egzistencijskog kvantifikatora nastaje jezik \mathcal{FL}_0 . Ovi jezici nemaju gotovo nikakvu primjenu u praksi, ali se često spominju u literaturi radi formaliziranja nekih drugih jezičnih oblika.

3.1.3 Interpretacija

Interpretacija daje potpuni opis svijeta kojeg promatramo. Kako svjetova može biti više, tako i interpretacija može biti više.

Formalnije, jedna interpretacija I sastoji se od nepraznog skupa Δ^I koji predstavlja domenu interpretacije te od funkcije preslikavanja koja svakom atomičkom konceptu A pridjeljuje skup $A^I \subseteq \Delta^I$ i svakom svojstvu R pridjeljuje binarnu relaciju $R^I \subseteq \Delta^I \times \Delta^I$. Slika 3.1 prikazuje primjer jedne interpretacije.



Slika 3.1. Primjer jedne interpretacije.

U ovom primjeru je interpretacija definirana skupom Δ^I koji sadrži elemente $\{a, b, c, d, e\}$, te preslikavanjem koji atomičkim konceptima A pridjeljuje skupove A^I , a svojstvima R pridjeljuje skupove R^I na način kako je to prikazano u Tablici 3.1 i Tablici 3.2:

Tablica 3.1. Interpretacija koncepata u primjeru kojeg prikazuje Slika 3.1.

Atomički koncept (A)	Skup u koji se koncept preslikava u interpretaciji (A^I)
STUDENT	$\{a, c\}$
OSOBA	$\{a, c, d\}$
PREDAVAČ	$\{d\}$
KOLEGIJ	$\{b, e\}$
TEŽAK	$\{e\}$

Tablica 3.2. Interpretacija svojstava u primjeru.

Svojstva (R)	Skup u koji se svojstvo preslikava u interpretaciji (R^I)
SLUŠA	$\{(a, b), (c, b), (d, e)\}$
PREDAJE	$\{(d, b)\}$

Kako je interpretacija potpuna slika svijeta, ono što nije navedeno u interpretaciji sigurno nije istinito. Tako je u ovoj konkretnoj interpretaciji izvjesno da individua d nije student

jer ne pripada skupu STUDENT^I . Kada bismo dodali informaciju da individua d uistinu jest STUDENT , tada bismo formalno gledajući stvorili sasvim novu interpretaciju (koja je različita od interpretacije I).

Koristeći interpretacije može se u potpunosti formalno definirati semantika svojstvenog \mathcal{AL} jezika:

$$\top^I = \Delta^I$$

$$\perp^I = \emptyset$$

$$(\neg A)^I = \Delta^I \setminus A^I$$

$$(C \sqcap D)^I = C^I \cap D^I$$

$$(\forall R.C)^I = \{a \in \Delta^I \mid \forall b. (a, b) \in R^I \rightarrow b \in C^I\}$$

$$(\exists R.\top)^I = \{a \in \Delta^I \mid \exists b. (a, b) \in R^I\}.$$

Riječima rečeno, interpretacija univerzalnog koncepta jest domena interpretacije, interpretacija praznog koncepta je prazan skup, interpretacija negiranog koncepta jest domena interpretacije umanjena za interpretaciju koncepta, interpretacija presjeka jest presjek interpretacija. Univerzalni i egzistencijski kvantifikatori imaju nešto složenije interpretacije. Univerzalnim kvantifikatorom opisuje se skup individua koji imaju svojstvo da su svim R relacijama povezani sa individuama iz interpretacije koncepta C . Egistencijalni kvantifikator (ovdje u ograničenom obliku jer se radi o \mathcal{AL} jeziku) opisuje skup individua koje su povezane s bilo kojom individuom relacijom R .

Tablica 3.3 prikazuje nekoliko primjera uporabe složenih konceptata i pregled skupova u koje se ti koncepti preslikavaju u interpretaciji prikazanoj na Slici 3.1.

Tablica 3.3. Primjeri složenih konceptata i njihove interpretacije.

Koncept (C)	Interpretacija (C ^I)
STUDENT \sqcap \exists SLUŠA. \top	{a, c}
\forall PREDAJE.TEŽAK	{a, b, c, e}
OSOBA \sqcap \exists PREDAJE. \top \sqcap \forall SLUŠA.TEŽAK	{d}

U drugom primjeru u tablici je zanimljivo da su obuhvaćene upravo one individue koje nemaju niti jednu relaciju PREDAJE.

Vrijedi primjetiti da u \mathcal{AL} jeziku ne postoji mogućnost da se definiraju izrazi pomoću relacija koje “ulaze u individuu”. Tako se npr. ne može opisati kolegij koji je slušan od

strane barem jednog studenta. Za takvu konstrukciju su potrebna *inverzna svojstva* koja nisu moguća u \mathcal{AL} jeziku, ali su moguća u nekim drugim proširenim oblicima jezika. Važno je biti svjestan ovakvih semantičkih ograničenja pri odabiru jezičnog obilka koji će se koristiti za modeliranje baze znanja.

Interpretacija i zaključivanje

U deskriptivnoj logici zaključci se donose neovisno o interpretaciji. Drugim riječima, iz postojećih definiranih logičkih izraza postupkom rasuđivanja se donose zaključci koji su valjani *u svim mogućim interpretacijama*. Ovo je jedno od temeljnih svojstava deskriptivne logike.

Primjerice, dva koncepta C i D su ekvivalentna ako je C^I jednak D^I u svim mogućim interpretacijama. Tako je izraz $\forall \text{SLUŠA.KOLEGIJ} \sqcap \forall \text{SLUŠA.TEŽAK}$ ekvivalentan izrazu $\forall \text{SLUŠA.(KOLEGIJ} \sqcap \text{TEŽAK)}$. Do ovakvog zaključka usitinu se i dolazi korištenjem poznatih algoritama za automatsko rasuđivanje u deskriptivnoj logici.

3.1.4 Porodica \mathcal{AL} jezika

U porodici \mathcal{AL} jezika se svojstveni jezik \mathcal{AL} proširuje novim elementima kako bi mu se povećala ekspresivnost. Na taj način nastaju razni jezični oblici.

Unija konceptata u deskriptivnoj logici označava se sa $C \sqcup D$; interpretacija unije jednaka je uniji interpretacija ($C^I \cup D^I$). Omogućavanje uporabe unije u deskriptivnoj logici se kod imenovanja jezičnog oblika označava velikim pisanim slovom \mathcal{U} , pa se jezik koji nastaje proširivanjem \mathcal{AL} jezika unijom označava sa $\mathcal{AL}\mathcal{U}$.

$$(C \sqcup D)^I = C^I \cup D^I$$

Uporaba *neograničenog egzistencijskog kvantifikatora* se u jezičnom obliku deskriptivne logike označava velikim slovom \mathcal{E} i omogućuje konstrukciju oblika $\exists R.C$. Interpretacija joj je slična interpretaciji sa ograničenom uporabom: opisuje se skup individua koji imaju svojstvo da su barem jednom R relacijom povezani sa individuum iz interpretacije koncepta C . Dakle, od ograničene uporabe se razlikuje u tome što je dozvoljena proizvoljna uporaba koncepta u doseg egzistencijskog kvantifikatora.

$$(\exists R.C)^I = \{a \in \Delta^I \mid \exists b. (a, b) \in R^I \wedge b \in C^I\}.$$

Negacija složenih konceptata se u imenovanju jezičnog oblika označava slovom \neg a omogućuje negaciju proizvoljno složenog koncepta, npr. $\neg(A \sqcap B)$. Interpretacija negiranog složenog koncepta jednaka je domeni umanjenoj za interpretaciju složenog koncepta.

$$(\neg C) = \Delta^I \setminus C^I$$

Semantički gledano, postoje jezici koji su međusobno ekvivalentni. Uzme li se u obzir da se unija može izraziti koristeći negaciju i presjek ($C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$), te da se egzistencijalni kvantifikator može izraziti koristeći negaciju i univerzalni kvantifikator ($\exists R.C \equiv \neg \forall R.\neg C$), postaje jasno da je jezik \mathcal{ALUE} ekvivalentan jeziku \mathcal{ALC} . Iz istog razloga se redovito pri imenovanju jezičnog oblika umjesto konstrukata \mathcal{UE} najčešće piše samo C .

Vrlo su korisne i neke skraćenice u deskriptivnoj logici koje pogoduju jednostavnijem izražavanju bez promjene semantike jezika:

$$\begin{aligned} C \rightarrow D &= \neg C \sqcup D && \text{(implikacija)} \\ C \leftrightarrow D &= (C \rightarrow D) \sqcap (D \rightarrow C) && \text{(bi-implikacija)} \\ \top &= A \sqcup \neg A && \text{(univerzalni koncept)} \\ \perp &= A \sqcap \neg A && \text{(prazan koncept)} \end{aligned}$$

Na primjer, pri opisivanju studenata koji slušaju samo teške kolegije moguće je koristiti ova dva ekvivalentna načina izražavanja:

- a) Student $\sqcap \forall$ sluša.(\neg Kolegij \sqcup Težak)
- b) Student $\sqcap \forall$ sluša.(Kolegij \rightarrow Težak)

Riječima rečeno, ako je nešto što Student sluša Kolegij onda je on Težak. Time se ne ograničava da Student sluša nešto drugo osim Kolegija; npr. moguće je da Student sluša i neki Tečaj (koji može ili ne mora biti Težak).

Slično vrijedi i za bi-implikaciju; tako je moguće reći da student od svih Kolegija voli samo one Teške a u isto vrijeme i da od svih Teških stvari voli samo Kolegije:

$$\text{Student } \sqcap \forall \text{voli.}(\text{Kolegij } \leftrightarrow \text{Težak})$$

Naravno, time se opet ne ograničava da Student voli i neke druge stvari (npr. Hranu ili Koncert), niti se ograničava njihov oblik (npr. Težak, Dobar i sl.).

Ovakve kratice koje služe samo za lakše izražavanje u jeziku ne treba miješati sa jezičnim konstruktima \sqsubseteq i \equiv koji se koriste pri definiciji pojmova u TBox-u (poglavlje 3.1.6).

Ukratko, ostali važniji jezični kostrukti koji se koriste u deskriptivnoj logici su:

- S – kratica za \mathcal{ALC} sa tranzitivnim svojstvima. Tranzitivna svojstva su svojstva koja se tranzitivno prenose nizom svojstava kao što su to PREDAK, POTOMAK, NADREĐEN itd. Primjerice, ako je Branko potomak od Ane, a Darko potomak od Branka, tada zbog tranzitivnosti relacije POTOMAK implicitno vrijedi i da je Darko potomak od Ane.

- \mathcal{H} – hijerarhija svojstava. Moguće je definirati jedno svojstvo kao podsvojstvo nekog drugog svojstva. Primjerice, može se naznačiti da je svojstvo VOZIPREMA podsvojstvo od svojstva IDEPREMA.
- \mathcal{N} – ograničenje kardinaliteta svojstva. Definira se minimalan i maksimalan broj relacija koji mogu izlaziti iz individue. Npr., moguće je definirati individue koje imaju najviše 2 brata ili najmanje 3 sestre.
- Q – kvalificirana ograničenja kardinaliteta svojstva. Ovaj konstrukt se koristi kada se želi pobrojati relacije sa određenom kodomenom, npr. kada se želi opisati individua kojoj su barem 2 brata studenti.
- \mathcal{F} – funkcionalna svojstva (svojstva kardinaliteta 1). Koriste se kada se želi naglasiti da npr. svojstvo OTACOD može poprimiti samo jednu individuu.
- I – inverzna svojstva. Koristi se kada se želi naglasiti da su neka dva svojstva komplementarna, npr. MANJIOD je međusobno inverzan sa VEĆIOD, SADRŽI sa JEDIOOD itd.
- ^(D) – uporaba podatkovnih vrijednosti. Npr. starost osobe u godinama, sistolički tlak, temperatura itd.
- O – korištenje imenovanih individua u logičkim izrazima. Ponekad je potrebno koristiti imenovane individue za opis koncepta, npr. skup individua koje slušaju predmete profesora Pavčevića.

Zanimljivo je spomenuti da jezik koji se koristi za izradu ontologija u domeni semantičkog weba (OWL-DL) ima semantičku ekspresivnost jezika $\mathcal{SHOIN}^{(D)}$.

3.1.5 Komponente baze znanja

Baza znanja koja se temelji na deskriptivnoj logici sastoji se od dvije osnovne komponente:

- TBox (T kao engl. *terminology*), koji sadrži terminologiju domene. Terminologija sadrži imena svih relevantnih konceptata i svojstava u domeni te njihove definicije i međusobne odnose;
- ABox (A kao engl. *assertion*), koji sadrži individue i relacije u domeni definirane koristeći pojmove iz TBox-a.

Svaka od ove dvije komponente ima i definirane svojstvene oblike rasuđivanja. Važno je naglasiti da iako je u deskriptivnoj logici granica između komponenti baze znanja postojana i jasno definirana, u OWL jeziku se ove dvije komponente ne razdvajaju nego je sve sadržano u jednoj jedinstvenoj bazi znanja.

3.1.6 TBox

TBox sadrži *terminološke aksiome* koji postavljaju tvrdnje o konceptima u domeni i o vezama između njih. U svom općem obliku jedan aksiom ima oblik:

$$C \sqsubseteq D.$$

Ovaj aksiom iznosi slijedeće (ekvivalentne) tvrdnje:

- koncept D generalniji od koncepta C;
- interpretacija koncepta C je podskup interpretacije koncepta D;
- ako individua zadovoljava koncept C, tada sigurno zadovoljava i koncept D.

TBox u praksi može sadržavati i tvrdnje oblika ekvivalencije:

$$C \equiv D.$$

Međutim, ekvivalencija ne proširuje semantiku TBox-a s obzirom da se jedna ekvivalencija može raspisati pomoću aksioma kao:

$$C \sqsubseteq D, C \supseteq D.$$

Proizvoljna interpretacija I zadovoljava aksiom $C \sqsubseteq D$ ako vrijedi da je C^I podskup od D^I .

Evo jednog primjera TBox-a:

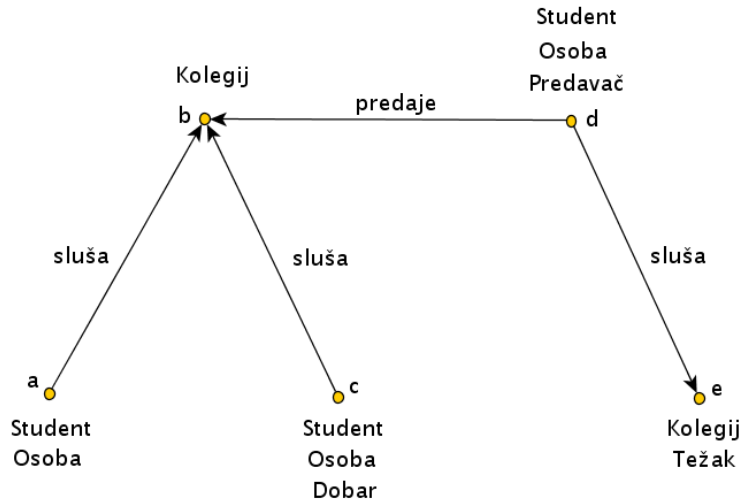
$$\text{PREDAVAČ} \equiv \text{OSOBA} \sqcap \exists \text{PREDAJE.KOLEGIJ}$$

$$\text{STUDENT} \equiv \text{OSOBA} \sqcap \exists \text{SLUŠA.KOLEGIJ}$$

U ovom primjeru TBox-a dane su definicije dva koncepta. Prva definicija govori da ako je neka individua osoba i postoji barem jedan kolegij kojeg predaje, tada je ta individua sigurno i predavač. Vrijedi i obrnuti smjer rasuđivanja: ako je neka individua predavač tada je ona sigurno i osoba i tada ona sigurno predaje barem jedan kolegij. Slično se može zaključiti i za drugu tvrdnju u navedenom TBox-u.

Na ovaj način TBox ograničava prostor prihvatljivih interpretacija. Samo neke od svih postojećih interpretacija će zadovoljiti specifikacije dane u TBoxu. Ako interpretacija I zadovoljava sve aksiome postavljene u TBox-u \mathcal{T} tada se za I kaže da je *model* od \mathcal{T} . Modela TBox-a također može biti (kao i interpretacija) beskonačno mnogo.

Primjer interpretacije koji je prikazan na Slici 3.1 *nije* model danog TBox-a jer individua d nije označena konceptom STUDENT. Kada se pridoda individui d i koncept STUDENT novodobivena interpretacija (Slika 3.2) jest model od predloženog TBox-a.



Slika 3.2. Interpretacija koja zadovoljava tvrdnje u TBox-u (model).

3.1.7 Oblici rasuđivanja u TBox-u

Postoji više oblika rasuđivanja vezanih za TBox komponentu. U ovom odjeljku dan je pregled nekih važnijih oblika.

Podrazumijevanje (engl. *subsumption*)

Kada je definiran neki proizvoljni TBox može se za neka dva pojma C i D zaključiti da li je (ili nije) jedan specifičniji (generalniji) od drugog. Koncept C je specifičniji od koncepta D s obzirom na TBox \mathcal{T} (piše se $C \sqsubseteq_{\mathcal{T}} D$) ako za sve modele I od \mathcal{T} vrijedi da je interpretacija od C podskup od interpretacije od D (tj. da je $C^I \subseteq D^I$).

Primjerice, za sljedeći TBox:

$$\text{PREDAVAČ} \equiv \text{OSOBA} \sqcap \exists \text{PREDAJE.KOLEGIJ}$$

$$\text{STUDENT} \equiv \text{OSOBA} \sqcap \exists \text{SLUŠA.KOLEGIJ}$$

mogu se donijeti sljedeći zaključci:

$$\text{PREDAVAČ} \sqsubseteq_{\mathcal{T}} \text{OSOBA}$$

$$\text{PREDAVAČ} \sqcap \exists \text{SLUŠA.KOLEGIJ} \sqsubseteq_{\mathcal{T}} \text{STUDENT}$$

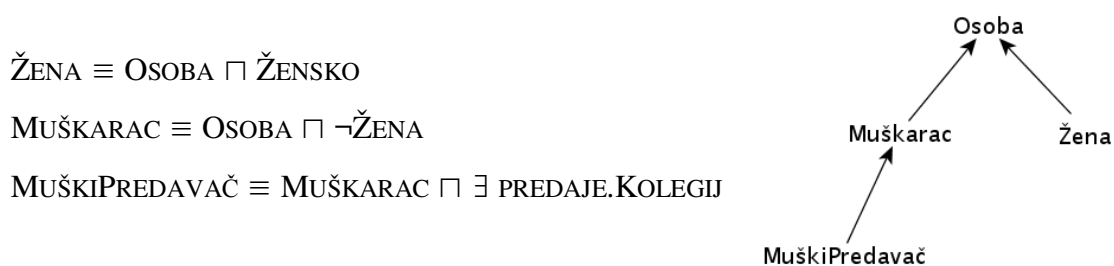
Prvi navedeni zaključak tvrdi da su u svakom mogućem modelu ovog TBox-a individue koje su obuhvaćene pojmom PREDAVAČ sigurno istovremeno i u pojmu OSOBA. Drugi zaključak tvrdi da su u svakom mogućem modelu zadanog TBox-a individue koje su u pojmu PREDAVAČ i koji slušaju neki KOLEGIJ sigurno istovremeno i u pojmu STUDENT.

Valja naglasiti da zadatak podrazumijevanja nije izvući sve ovakve moguće zaključke (dapače, moglo bi ih biti beskonačno mnogo), nego za jedan određeni upit o podrazumijevanju dva pojma potvrditi ili odbaciti podrazumijevanje među njima.

Klasifikacija (engl. *classification*)

Zadatak klasifikacije kao oblika rasuđivanja u deskriptivnoj logici jest organizirati sve definirane koncepte u TBox-u u hijerarhiju s obzirom na njihovu međusobnu specifičnost/generalnost.

Slika 3.3 prikazuje primjer jednostavnog TBox-a i izračunate hijerarhije koristeći postupak klasifikacije.



Slika 3.3. Primjer TBox-a i izračunate klasifikacije pojmova.

Važno je primjetiti da generirana hijerarhija nije nužno stablo, nego može biti i aciklički usmjereni graf. Primjerice, uvođenjem u hijerarhiju pojma PREDAVAČ, koji je specifičniji od pojma OSOBA a generalniji od pojma MUŠKIPREDAVAČ nastaje novi put od koncepta MUŠKIPREDAVAČ do koncepta OSOBA.

Ako se korištenjem relacije podrazumijevanja zatvori ciklus (npr. od dva, tri ili više pojmova) nužno slijedi da su svi pojmovi u ciklusu ekvivalentni. Stoga ciklusi u dobivenom grafu pri klasifikaciji nisu mogući.

Računanje klasifikacije može se svesti na više uzastopnih računanja podrazumijevanja.

Zadovoljivost (engl. *satisfiability*)

Neki koncept C je zadovoljiv s obzirom na TBox \mathcal{T} ako postoji bilo koji model I od \mathcal{T} za kojeg je C^I neprazan skup. Ako ne postoji niti jedan takav model, jasno je da u definiciji koncepta C postoji kontradikcija.

Kontradikcija u pojmu može postojati i kada nije definiran TBox, tj. sa praznim TBoxom. Na primjer, sljedeći koncept je kontradiktoran jer govori o studentu koji sluša samo ono što je lagano (tj. nije teško) a istovremeno barem nešto što sluša jest teško.

$$\forall \text{SLUŠA.}\neg\text{TEŽAK} \sqcap \exists \text{SLUŠA.TEŽAK}$$

Kada je TBox neprazan još je teže prostim okom uočiti kontradikciju:

$$\checkmark\text{ŽENA} \equiv \text{OSOBA} \sqcap \checkmark\text{ŽENSKO}$$

$$\text{MUŠKARAC} \equiv \text{OSOBA} \sqcap \neg\checkmark\text{ŽENA}$$

S obzirom na ovako definiran TBox izraz $\exists\text{PRIJATELJ.MUŠKARAC} \sqcap \forall\text{PRIJATELJ.}\checkmark\text{ŽENA}$ nije zadovoljiv. Poteškoća je što je disjunkcija između pojmova MUŠKARAC i ŽENA definirana posredno preko TBox-a, dok je u prethodnom primjeru bila eksplicitno dana u izrazu (TEŽAK, $\neg\text{TEŽAK}$).

3.1.8 ABox

ABox definira trenutno stanje svijeta u kojem se želi rasuđivati. U općem slučaju se u TBoxu definiraju pojmovi koji vrijede univerzalno i u svakom trenutku (kao što su OSOBA, STUDENT, PREDAVAČ), dok se u ABoxu definiraju individue koje predstavljaju jednu konkretnu sliku svijeta (kao što su *marko*, *ana*, *zemljopis*).

Sintaksa za izražavanje tvrdnji u ABoxu jest sljedeća:

$$a : C \quad (\text{individua } a \text{ zadovoljava koncept } C)$$

$$(a, b) : R \quad (\text{individue } a \text{ i } b \text{ su povezane relacijom } R)$$

Interpretacija I preslikava individue iz ABox-a u elemente skupa Δ^I . Interpretacija zadovoljava *jednu* tvrdnju iz ABoxa \mathcal{A} ako vrijedi sljedeće:

$$a : C \quad \text{ako} \quad a^I \in C^I$$

$$(a, b) : R \quad \text{ako} \quad (a^I, b^I) \in R^I$$

Da bi interpretacija I bila *model* definiranog ABoxa \mathcal{A} ona mora zadovoljiti *sve* tvrdnje iz \mathcal{A} .

Primjer jednog ABoxa može izgledati ovako:

$$\textit{marko} : \text{STUDENT}$$

$$\textit{zemljopis} : \text{KOLEGIJ}$$

$$\textit{zemljopis} : \text{TEŽAK}$$

$$(\textit{marko}, \textit{zemljopis}) : \text{SLUŠA}$$

Važno je uočiti da ABox za razliku od interpretacije daje *nepotpunu sliku* svijeta. Tako npr. gore opisani ABox *ne* implicira sljedeću tvrdnju:

$$\textit{marko} : \forall\text{SLUŠA.TEŽAK}$$

Naime, moguće je da Marko u nekim interpretacijama (modelima) sluša i neke druge kolegije od kojih neki i nisu teški (npr. likovna kultura), iako to nije eksplicitno navedeno

u ABox-u. Dakle, jedan ABox može imati više različitih modela, a da bi se mogao donijeti valjani zaključak on mora vrijediti za sve moguće modele ABox-a. Prema tome, slično kao i kod TBox-a, svaka tvrdnja u ABox-u umanjuje broj interpretacija koji može biti i model.

Ovakav oblik izražavanja ključan je za deskriptivnu logiku i temelji se na *pretpostavci otvorenog svijeta*. Pod pojmom *otvoreni svijet* se podrazumijeva da svijet nije u potpunosti opisan tvrdnjama danim u bazi znanja te da je moguće da postoji i informacija koja u bazi znanja nedostaje.

3.1.9 Oblici rasuđivanja u ABox-u

Slično kao i za TBox, i za ABox postoje zaključci koji se mogu donijeti neovisno o interpretaciji. Ovdje je prikazan pregled najvažnijih oblika rasuđivanja kod ABox-a.

Konzistentnost (engl. *consistency*)

Konzistentnost je oblik rasuđivanja koji u bazi znanja pronalazi kontradikcije. Kako bi se ustvrdilo postoji li u bazi znanja definiranoj TBox-om \mathcal{T} i ABox-om \mathcal{A} kontradikcija valja utvrditi postoji li barem jedna interpretacija koja je model i za \mathcal{A} i za \mathcal{T} .

Tablica 3.4. prikazuje primjer baze znanja u kojem je ABox nekonzistentan sa TBox-om. U prikazanom primjeru su u TBox-u definirani odnosi pojmova a u ABox-u nepotpuna slika svijeta. Ovako zadana baza znanja je nekonzistentna jer je za individuu *dumbo* definirano da je tamno sive boje (u linijama 2 i 3) i da je svaka njegova boja svijetlo siva (u liniji 4). To se kosi sa tvrdnjom u TBox-u koja tvrdi da nešto ne može biti istovremeno i svijetlo i tamno sivo (linija 4).

Tablica 3.4. Primjer nekonzistentne baze znanja.

ABox \mathcal{A}	TBox \mathcal{T}
<i>dumbo</i> : SISAVAC	
<i>siva23</i> : TAMNOSIVA	SLON \equiv
(<i>dumbo</i> , <i>siva23</i>) : JEBOJE	\equiv SISAVAC \sqcap \exists DIO TIJELA.SURLA \sqcap \forall JEBOJE.SIVA
<i>dumbo</i> : \forall JEBOJE.SVIJETLOSIVA	SIVA \equiv SVIJETLOSIVA \sqcup TAMNOSIVA
<i>surla14</i> : SURLA	$\perp \equiv$ SVIJETLOSIVA \sqcap TAMNOSIVA
(<i>dumbo</i> , <i>surla14</i>) : DIO TIJELA	

Provjera instanci (engl. *instance checking*)

Provjera instanci je postupak rasuđivanja koji za zadane TBox \mathcal{T} , ABox \mathcal{A} , promatranu individuu a i promatrani koncept C provjerava da li individua a pripada konceptu C u svakom mogućem modelu od \mathcal{A} i \mathcal{T} . Kada je to ispunjeno piše se:

$$\mathcal{A}, \mathcal{T} \models a : C \quad (\text{riječima: iz } \mathcal{A} \text{ i } \mathcal{T} \text{ slijedi da individua } a \text{ sigurno zadovoljava } C)$$

Ako se iz primjera prikazanog Tablicom 3.4 izbaci linija 2 ($siva23 : TAMNOSIVA$), tada baza znanja prestaje biti nekonsistentna pa postaje smisleno donositi zaključke pomoću postupka provjere instanci. U tom slučaju može se zaključiti da individua *dumbo* pripada konceptu $SLON$ ($\mathcal{A}, \mathcal{T} \models dumba : SLON$) jer zadovoljava izraz $SISAVAC \sqcap \exists DIOTIJELA.SURLA \sqcap \forall JEBOJE.SIVA$ i to zbog sljedećih činjenica:

- *dumbo* jest $SISAVAC$ (linija 1),
- *dumbo* ima dio tijela koji je $SURLA$ (*surla14*, linije 5 i 6),
- svaka boja *dumba* je $SVIJETLOSIVA$ a zbog toga ujedno i $SIVA$ (linija 4).

Iz ovog primjera vidimo da doneseni zaključci nisu bili baš očiti na prvi pogled – dapače, porastom broja tvrdnji u bazi znanja postaje nezamislivo donositi zaključke bez automatiziranog postupka rasuđivanja.

3.1.10 Posredno rasuđivanje

Mnogi oblici rasuđivanja se jednostavnim transformacijama mogu svesti jedan na drugi. Pokazuje se da je dovoljno implementirati algoritam za samo jedan oblik rasuđivanja (npr. za zadovoljivost ili podrazumijevanje) – ostali oblici rasuđivanja mogu se računati posredno. Većina današnjih implementacija računa zadovoljivost.

Podrazumijevanje – zadovoljivost

Podrazumijevanje se može svesti na zadovoljivost na sljedeći način:

- $C \sqsubseteq_{\mathcal{T}} D$ akko $C \sqcap \neg D$ nije zadovoljiv s obzirom na \mathcal{T} .

Riječima rečeno, ako se želi dokazati da je D generalniji od C s obzirom na postojeće definicije u TBox-u, formira se novi izraz oblika $C \sqcap \neg D$ te se ispituje njegova zadovoljivost. Ako je izraz zadovoljiv tada se može zaključiti da D nije generalniji od C , a ako nije zadovoljiv tada D jest generalniji od C .

Primjerice, ako se želi pokazati da je u TBoxu pojam $OSOBA$ generalniji od pojma $STUDENT$ (tj. da $STUDENT \sqsubseteq_{\mathcal{T}} OSOBA$), tada se formira novi složeni pojam $STUDENT \sqcap \neg OSOBA$ i provjerava njegova zadovoljivost. Intuitivno je jasno da je ovaj pojam predstavlja prazan skup, tj. da bi netko mogao biti $STUDENT$ on mora biti u prvom redu $OSOBA$. Ako se dokaže (koristeći zadovoljivost) da u nekom proizvoljnom modelu može postojati individua koja je istovremeno i $STUDENT$ i nije $OSOBA$ tada bi se pokazalo da

OSOBA nije generalniji pojam od pojma STUDENT. Isto tako dokaže li se da takva individua ne može postojati u nijednom modelu, dokazano je da je pojam OSOBA generalniji od pojma STUDENT.

Moguće je i obratno rasuđivanje; zadovoljivost se može svesti na podrazumijevanje na sljedeći način:

- C je zadovoljiv s obzirom na \mathcal{T} ako ne vrijedi da $C \sqsubseteq_{\mathcal{T}} \perp$.

Riječima rečeno, ako se želi dokazati pomoću podrazumijevanja da je C zadovoljiv izraz s obzirom na TBox \mathcal{T} , provjerava se da li je u svim mogućim modelima koncept C specifičniji od praznog koncepta. Ako se to uspije dokazati, tada se zaključuje da C nije zadovoljiv; u protivnom se zaključuje da C jest zadovoljiv koncept.

Kako prazni koncept po definiciji ne može sadržavati niti jednu individuu, ako je C specifičniji od njega tada niti C ne može sadržavati niti jednu individuu. Na taj način se izravno zaključuje i o zadovoljivosti koncepta C .

Konzistentnost – provjera instanci

Računanje konzistentnosti se može svesti na provjeravanje instanci na sljedeći način:

- ABox \mathcal{A} je konzistentan s obzirom na TBox \mathcal{T} ako $\mathcal{A}, \mathcal{T} \models a : \perp$.

Želi li se doznati da li je ABox \mathcal{A} konzistentan sa TBox-om \mathcal{T} ispituje se da li je neka proizvoljna individua a nužno instanca praznog koncepta. Ako se to uspije pokazati tada se zaključuje da je sustav nekonzistentan, u protivnom se zaključuje da je sustav konzistentan.

Ovaj način svodenja rasuđivanja je valjan jer se nekonzistentnost ABox-a i TBox-a odražava na način da svaka individua u bazi znanja nužno postaje instanca praznog koncepta.

Provjera instanci se svodi na konzistentnost na sljedeći način:

- $\mathcal{A}, \mathcal{T} \models a : C$ ako $\mathcal{A} \cup \{a : \neg C\}$ nije konzistentan.

Ako se želi pokazati da individua a zasigurno jest instanca koncepta C tada se pokušava pokazati da suprotan slučaj stvara nekonzistentnost. Individui a se dodaje koncept $\neg C$ i pokušava se pokazati da je takav slučaj nekonzistentan. Ako se to uspije to može značiti samo da individua doista jest instanca koncepta C . S druge strane, ako je takav sustav konzistentan tada je jasno da je moguć model u kojem je a instance od $\neg C$, dakle a nije nužno instanca od C u svim mogućim interpretacijama.

Primjerice, želi li se pokazati da je individua *marko* instanca koncepta STUDENT, pridjeljuje mu se koncept \neg STUDENT te se pokušava dokazati nekonzistentnost takve baze znanja.

Zadovoljivost - konzistentnost

Zadovoljivost koncepta može se izračunati posredno koristeći računanje konzistentnosti na sljedeći način:

- C je zadovoljiv s obzirom na TBox \mathcal{T} akko $a : C$ je konzistentan.

Stvaranjem jedne nove individue a i dokazivanjem da pridjeljivanje koncepta C toj individui ne izaziva nekonzistentnost zaključuje se je koncept C zadovoljiv (time se ujedno dobije i primjer jedne individue koja ga zadovoljava). Ako opisani postupak izaziva nekonzistentnost baze znanja jedini uzrok tome može biti nezadovoljivost izraza C .

Suprotan smjer (svođenje konzistentnosti na zadovoljivost) se za \mathcal{ALC} jezični oblik može izvesti uz nešto složeniju proceduru. Da bi se dokazalo da je ABox \mathcal{A} konzistentan prvo se \mathcal{A} proširi koristeći pravila proširenja ABoxa:

$$a : (C \sqcap D) \quad \rightarrow \quad a : C, a : D$$

$$a : \forall R.C, (a, b) : R \quad \rightarrow \quad b : C$$

U slučaju *unije* (disjunkcije) stvara se replika ABox-a pri čemu svaki novodobiveni ABox zadovoljava po jedan koncept iz promatrane unije. Obavezno je i da se koncepti koji su definirani u TBox-u raspišu na elementarne koncepte (procesom *ekspanzije* definicije koncepata se raspisuju dok je to moguće). Pri tome je važno da je TBox acikličan, tj. da ne postoje zatvoreni krugovi u definicijama pojmova. Kada je proces proširivanja gotov, za svaki se dobiveni ABox $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k$, i svaku individuu a provjerava da li je zadovoljiva konjunkcija pojmova $\{C \mid a : C \in \mathcal{A}_i\}$.

Svođenje podrazumijevanja na provjeru instanci

Podrazumijevanje se može svesti na provjeru instanci:

- $C \sqsubseteq_{\mathcal{T}} D$ akko $\{a : C\}, \mathcal{T} \models a : D$

Kada se želi utvrditi da li je C specifičniji pojam od D tada se stvara novi ABox u kojem postoji samo individua a koja je instanca pojma C , te se pokušava ustvrditi da li iz takve baze znanja nužno slijedi da je individua a u svim modelima instanca pojma D . Ako je to istina, zaključuje se da je pojam C uzrokovao svrstavanje individue a u pojam D , što izravno znači i da je pojam C specifičniji od D .

3.1.11 Semantike otvorenog i zatvorenog svijeta

Deskriptivna logika slijedi *pretpostavku otvorenog svijeta* (engl. *open world assumption*). Otvoreni svijet u rasuđivanju dozvoljava da određena informacija u bazi znanja potencijalno nedostaje kao i mogućnost nedokučenog odgovora na postavljeno pitanje.

Takva semantika oprečna je *pretpostavci zatvorenog svijeta* (engl. *closed world assumption* [21]) koja podrazumijeva da je informacija u bazi znanja kompletna. Kod zatvorenog svijeta se podrazumijeva da tvrdnje koje u bazi nisu navedene (eksplicitno) ili se do njih ne može doći rasuđivanjem (implicitno) nisu istinite – ovaj oblik zaključivanja se još naziva i negacija kao neuspjeh (engl. *negation as failure*) ili skraćeno NF. Kod zatvorenog svijeta svako postavljeno pitanje dobiva odgovor (ne postoji mogućnost neodlučnosti).

3.1.12 Deskriptivna logika i predikatna logika prvog reda

Semantiku deskriptivne logike može se smatrati podskupom semantike predikatne logike prvog reda (iako ne uvijek, to ne vrijedi za neke posebne oblike deskriptivne logike). Atomički koncepti deskriptivne logike mogu se predstaviti unarnim predikatima a atomičke relacije binarnim predikatima predikatne logike:

atomički koncept A	\Leftrightarrow	unarni predikat P_A
atomička relacija R	\Leftrightarrow	binarni predikat P_R

Složeni koncepti deskriptivne logike prevode se u izraze predikatne logike transformacijom φ na sljedeći način:

$$\begin{aligned}
 \varphi^x(A) &= P_A(x) \\
 \varphi^x(\neg C) &= \neg \varphi^x(C) \\
 \varphi^x(C \sqcap D) &= \varphi^x(C) \wedge \varphi^x(D) \\
 \varphi^x(C \sqcup D) &= \varphi^x(C) \vee \varphi^x(D) \\
 \varphi^x(\exists R.C) &= \exists y. P_R(x, y) \wedge \varphi^y(C) \\
 \varphi^x(\forall R.C) &= \forall y. P_R(x, y) \rightarrow \varphi^y(C)
 \end{aligned}$$

Primjerice, evo kako izgleda transformacija izraza $\forall \text{SLUŠA.}(\text{TEŽAK} \sqcap \text{ZANIMLJIV})$ u predikatnu logiku prvog reda:

- korak 1. $\forall y P_{\text{sluša}}(x, y) \rightarrow \varphi^y(\text{TEŽAK} \sqcap \text{ZANIMLJIV})$
korak 2. $\forall y P_{\text{sluša}}(x, y) \rightarrow P_{\text{Težak}}(y) \wedge P_{\text{Zanimljiv}}(y)$

U prisustvu TBox-a \mathcal{T} primjenjuje se sljedeća transformacija:

$$\varphi(\mathcal{T}) = \forall x. \bigwedge_{D \sqsubseteq E \in \mathcal{T}} \varphi^x(D) \rightarrow \varphi^x(E)$$

Semantika TBox-a jest koncipirana na način da svaka sadržana tvrdnja mora vrijediti za svaku individuu. Zbog toga se u predikatnoj logici prvog reda stvara jedna konjunkcija implikacija koja mora vrijediti za sve postojeće individue (pa stoga $\forall x$ na početku izraza).

U prisustvu ABox-a individue a se zamjenjuju konstantama c_a te se izvršavaju sljedeće transformacije:

$$\begin{aligned}\varphi(a : C) &= \varphi^x(C)[c_a] \\ \varphi((a, b) : R) &= P_R(c_a, c_b) \\ \varphi(\mathcal{A}) &= \bigwedge_{\beta \in \mathcal{A}} \varphi(\beta)\end{aligned}$$

Iz izloženog je vidljivo da je transformacija iz deskriptivne logike u predikatnu logiku prvog reda vrlo jednostavna (dok je transformaciju u drugom smjeru rijetko moguće izvesti). Evidentno je da je zapis u deskriptivnoj logici puno koncizniji i jasniji nego u predikatnoj logici, što predstavlja temeljnu prednost deskriptivne logike nad predikatnom logikom prvog reda. Dakako, jasno je i da je ekspresivnost u deskriptivnoj logici umanjena u odnosu na onu u predikatnoj logici.

3.1.13 Tableau postupak rasuđivanja

Tableau [3], [45] je ispravan i kompletan algoritam za rasuđivanje u deskriptivnoj logici koji rješava problem zadovoljivosti. U svojoj osnovnoj verziji tableau je primjenjiv na \mathcal{ALC} jezični oblik a već postoje i proširenja na mnoge ostale jezične oblike deskriptivne logike.

Tableau za \mathcal{ALC}

Da bi odgovorio na pitanje da li je \mathcal{ALC} koncept C zadovoljiv, tableau pokušava konstruirati model koji će zadovoljiti sve uvjete definirane u C . Ako u tome (ne) uspije, tableau odgovara da koncept jest (nije) zadovoljiv.

U početku se zadani koncept C pretvori u ekvivalentan koncept C_0 koji je zapisan u negacijskoj normalnoj formi dobivenoj transformacijom pomoću De Morganovih pravila:

$$\begin{aligned}\neg(C \sqcap D) &\rightsquigarrow \neg C \sqcup \neg D \\ \neg(C \sqcup D) &\rightsquigarrow \neg C \sqcap \neg D \\ \neg\neg C &\rightsquigarrow C \\ \neg\forall R.C &\rightsquigarrow \exists R.\neg C \\ \neg\exists R.C &\rightsquigarrow \forall R.\neg C\end{aligned}$$

Model kojeg tableau algoritam konstruira jest stablo u kojem je svaki čvor x označen labelom $\mathcal{L}(x)$ koja sadrži skup koncepata koji taj čvor zadovoljava. Ako je bilo koji koncept D element labele $\mathcal{L}(x)$, to zapravo znači da je u građenom modelu čvor x instanca koncepta D . Bridovi u stablu označavaju se imenima relacija.

Model se inicijalizira jednim čvorom x_0 s pripadnom labelom $\mathcal{L}(x_0) = \{C_0\}$. Zatim se u proizvoljnom redosljedu primjenjuju *tableau pravila*:

\sqcap -pravilo:

ako je $C_1 \sqcap C_2 \in \mathcal{L}(x)$
postavi $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$

\sqcup -pravilo:

ako je $C_1 \sqcup C_2 \in \mathcal{L}(x)$
postavi $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C\}$ za neki $C \in \{C_1, C_2\}$

\exists -pravilo:

ako je $\exists S.C \in \mathcal{L}(x)$
tada stvori novi cvor y sa $\mathcal{L}(\langle x, y \rangle) = \{S\}$ i $\mathcal{L}(y) = \{C\}$

\forall -pravilo:

ako je $\forall S.C \in \mathcal{L}(x)$ i postoji y koji je S -sljedbenik od x
postavi $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$

Pravila se primjenjuju na građenom stablu sve dok mu ona dodaju novu informaciju. Stablo sadrži *trivijalnu kontradikciju* (engl. *clash*) ako labela bilo kojeg čvora sadrži koncepte $\{A, \neg A\}$ ili prazni koncept (Slika 3.4). Stablo je *kompletno* ako se više niti jedno pravilo ne može primjeniti na njega (Slika 3.5).

Tableau će odgovoriti da je koncept C zadovoljiv ako sagradi kompletno stablo. Ako u tome ne uspije, odgovoriti će da koncept nije zadovoljiv.

Drugo pravilo u skupu tableau pravila je nedeterminističko, tj. postoji više mogućnosti kako ga se može primjeniti. Tableau pokušava sa jednim izborom i nastavlja graditi stablo sve dok se ne postigne kompletno stablo ili dok ne dođe do prve trivijalne kontradikcije. U potonjem slučaju, algoritam se vraća na poziciju “pogrešne odluke” i pokušava sa sljedećom mogućom opcijom (engl. *backtracking*).

Zbog nedeterminizma u drugom pravilu složenost algoritma se drastično povećava. Za sve disjunkcije koje se pojave u labelama stabla mora se pronaći odgovarajuća kombinacija odabira kako bi stablo bilo kompletno. Tu se otvara mogućnost za strategije odabira mogućnosti pri disjunkciji.

Za slučaj kada koncept C nije zadovoljiv, potrebno je biti siguran da u cijelom mogućem prostoru pretraživanja ne postoji niti jedno kompletno stablo – stoga tableau algoritam vrši potpuno pretraživanje prostora. Zbog ovakvih slučajeva postoje razne metode optimizacije koje smanjuju prostor pretraživanja prepoznavanjem već pretraženih struktura.

Primjeri rasuđivanja

Neka se želi doznati da li su zadovoljivi sljedeći izrazi:

$$\begin{aligned} & A \sqcap \exists R.B \sqcap \forall R.\neg B \\ & A \sqcap \exists R.B \sqcap \forall R.(\neg B \sqcup \exists S.E) \end{aligned}$$

- složenost algoritma u PSpace.

Kod dodatnih proširenja tableau algoritma na druge jezične oblike deskriptivne logike složenost izvođenja u pravilu dodatno raste.

Proširenje za TBox

TBox, kao što je definirano u odjeljku 3.1.6, sadržava skup tvrdnji oblika:

$$C \sqsubseteq D$$

Za primjer može se pogledati odnos koncepta STUDENT i koncepta OSOBA.

- STUDENT \sqsubseteq OSOBA
- STUDENT \rightarrow OSOBA
- \neg STUDENT \sqcup OSOBA

Ove tri tvrdnje su ekvivalentne. Tvrdnja *a*) govori da je koncept OSOBA generalniji od koncepta STUDENT. Iz toga slijedi da su svi studenti nužno istovremeno i osobe (tvrdnja *b*)). Tvrdnja *c*) je samo raspisani oblik tvrdnje *b*), a govori da za sve individue na svijetu vrijedi da je ona osoba ili da nije student. Drugim riječima, ne postoji nešto što je istovremeno i student i nije osoba.

Iz ovoga slijedi i proširenje tableau algoritma za TBox [3]. Za svaki čvor u stablu tableau algoritma mora vrijediti svaka tvrdnja iz TBox-a. Zbog toga se svim čvorovima u stablu za svaku tvrdnju iz TBox-a u labelu dodaje koncept (nakon pretvorbe u negacijsku normalnu formu):

$$\neg C \sqcup D$$

Slijedi da se tableau algoritam za TBox dobiva proširivanjem \mathcal{ALC} tableau algoritma jednim dodatnim pravilom:

$$\begin{aligned} & \mathcal{T}\text{-pravilo:} \\ & \text{ako je } C_1 \sqsubseteq C_2 \in \mathcal{T} \\ & \text{postavi } \mathcal{L}(x) = \mathcal{L}(x) \cup \{NNF(\neg C_1 \sqcup C_2)\} \end{aligned}$$

Ovo pravilo se primjenjuje *na svaki čvor u stablu*. Poteškoća kod ovog pravila jest što ono unosi drastično velik broj disjunkcija u stablo. Kako smo već spomenuli, pravilo disjunkcije (drugo tableau pravilo) je nedeterminističko i unosi veliku složenost u algoritam. Tableau algoritam s proširenjem na TBox ima složenost 2NExpTime [3].

Optimizacije

Lako se uočava da će svaki novo-stvoreni čvor u stablu jako sličiti roditelju, s obzirom da imaju mnoštvo koncepata u labelama dospjelih iz TBox-a. U većini slučajeva će ovo rezultirati neprestanim generiranjem čvorova djece (neprekidni lanci istih čvorova). Za

rješavanje ovog problema koristi se tehnika detekcije ciklusa koja se naziva *blokiranje* (engl. *blocking*) [3]. Bez blokiranja tableau algoritam za TBox gubi svojstvo terminacije, tj. u nekim (mnogim) slučajevima ne završava.

Kada god se pojavi situacija da čvor x ima jača ograničenja od bilo kojeg svog potomka y , čvor y i svi njegovi potomci se *blokiraju*. Na blokiranim čvorovima se zaustavlja primjena tableau pravila – uzima se da je podgrana stabla od čvora y zadovoljiva.

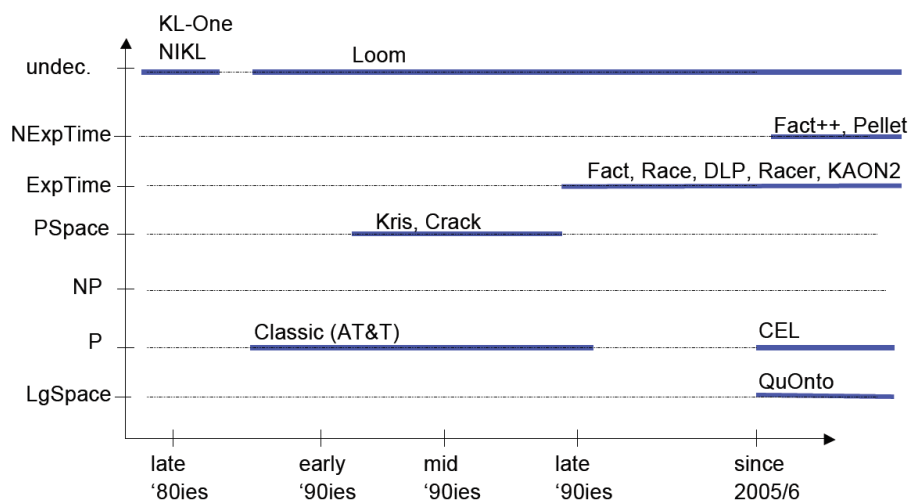
Problem kod upotrebe blokiranja jest samo detektiranje da je labela jednog čvora zapravo podskup labela drugog čvora zbog potencijalno velikog broja čvorova u stablu. Kao posljedica postupak blokiranja dodatno povećava složenost izvođenja (ali je i nužan za osiguranje terminacije algoritma, što je mnogo važnije).

Osim blokiranja postoje i mnoge druge metode optimizacije tableau algoritma. Pregled tehnika se može naći u [58].

3.1.14 Postojeći sustavi za rasuđivanje

Postojeće implementacije sustava za rasuđivanje u deskriptivnoj logici značajno variraju po ekspresivnosti, brzini izvođenja (čak i do razine neodlučljivosti), jeziku implementacije itd.

Slika 3.6 prikazuje pregled najpoznatijih sustava za rasuđivanje kroz godine s obzirom na njihovu složenost izvođenja. Sa slike je evidentno da je upravo složenost izvođenja usko grlo primjene deskriptivne logike u široj uporabi. Danas se može reći da su najzastupljenije implementacije u praktičnoj uporabi *Pellet*, *Fact++* i *Racer*.



Slika 3.6. Implementacije sustava za rasuđivanje u deskriptivnoj logici.

3.2 Bayesove mreže

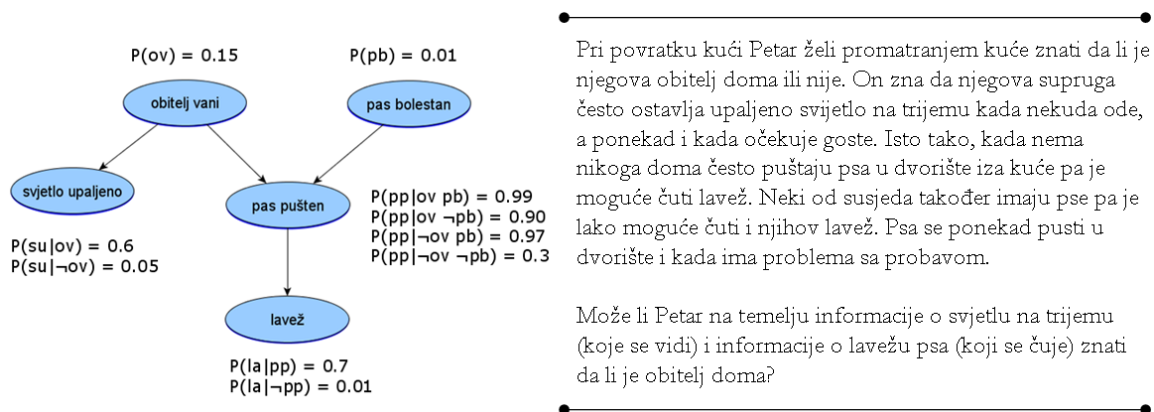
Bayesova mreža (engl. *Bayesian network*) je probabilistički grafički model kojim se prikazuju zavisnosti među varijablama. Sama mreža je usmjereni aciklički graf u kojem čvorovi predstavljaju varijable a bridovi njihove međuzavisnosti. Na temelju Bayesove mreže moguće je izračunati očekivanja (vjerojatnosti) svih nepoznatih varijabli u sustavu na temelju već poznatih varijabli (dokaza) [10].

Velika većina događaja vezanih uz istraživanje na Bayesovim mrežama dogodila se već u osamdesetim godinama [37] (engl. *belief networks, causal networks, influence diagrams*) kada se formirala opće prihvaćena struktura mreže, razvili mnogi alati za rasuđivanje i kada su se Bayesove mreže već počele koristiti u raznim primjenama. U devedesetima se pojavljuju i brojni algoritmi za učenje Bayesovih mreža iz skupova podataka. Bayesove mreže i danas predstavljaju model primjenjiv u računalnim sustavima u raznim područjima ljudskog djelovanja.

Kako bi se definirala jedna Bayesova mreža potrebno je definirati:

- Čvorove u mreži (varijable u problemu),
- Moguće ishode svih čvorova (vrijednosti koje varijable mogu poprimiti),
- Bridove u mreži (povezanost varijabli),
- Združene distribucije vjerojatnosti ishoda u svakom pojedinom čvoru ovisno o njegovim roditeljima u mreži.

Za čvorove koji nemaju roditelje potrebno je definirati samo *a priori* očekivanja njihovih ishoda. *A priori* očekivanja ishoda čvorova koji imaju roditelje definirana su posredno preko pripadnih tablica združene distribucije vjerojatnosti i preko *a priori* očekivanja njihovih roditelja. Stoga je suvišno u definiciji mreže eksplicitno navoditi *a priori* vjerojatnosti ishoda za čvorove koji imaju roditelje.



Pri povratku kući Petar želi promatranjem kuće znati da li je njegova obitelj doma ili nije. On zna da njegova supruga često ostavlja upaljeno svjetlo na trijemu kada nekuda ode, a ponekad i kada očekuje goste. Isto tako, kada nema nikoga doma često puštaju psa u dvorište iza kuće pa je moguće čuti lavež. Neki od susjeda također imaju pse pa je lako moguće čuti i njihov lavež. Psa se ponekad pusti u dvorište i kada ima problema sa probavom.

Može li Petar na temelju informacije o svjetlu na trijemu (koje se vidi) i informacije o lavežu psa (koji se čuje) znati da li je obitelj doma?

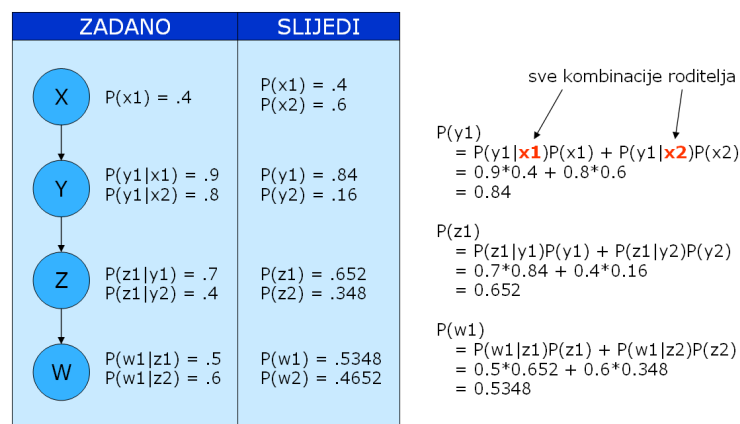
Slika 3.7. Primjer Bayesove mreže.

Primjer Bayesove mreže prikazan je na Slici 3.7. Uz svaki čvor definirana je pripadna združena distribucija vjerojatnosti.

3.2.1 Računanje *a priori* očekivanja čvorova

A priori očekivanje ishoda u Bayesovoj mreži podrazumijeva očekivanje ishoda koje vrijedi kada nije poznat dokaz niti za jedan čvor u mreži.

Slika 3.8 prikazuje jedan primjer računanja *a priori* očekivanja u Bayesovoj mreži. Čvor *X* ima *a priori* vrijednosti očekivanja eksplicitno zadane (jer nema roditelja). Ostalim čvorovima ona se računa iz pripadnih združenih distribucija vjerojatnosti i *a priori* očekivanja ishoda roditelja u mreži.



Slika 3.8. Primjer računanja *a priori* očekivanja čvorova u Bayesovoj mreži.

Iz postupka koji je prikazan na Slici 3.8. vidi se da se računanje obavlja slijedno niz mrežu (s obzirom na smjer bridova u mreži). Da bi se izračunalo *a priori* očekivanje nekog čvora nužno je prethodno izračunati *a priori* očekivanja ishoda svih njegovih roditelja.

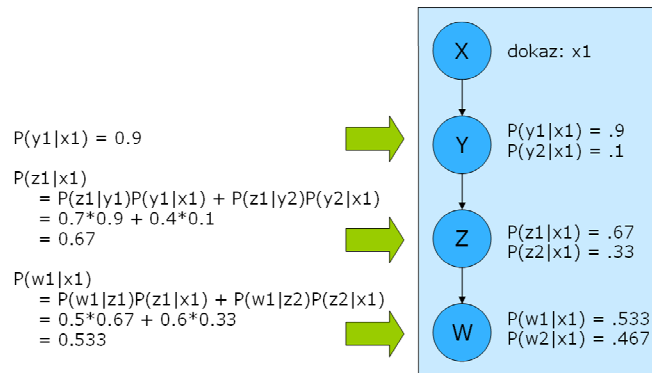
Pri računanju *a priori* očekivanja jednog čvora prolazi se iterativno kroz sve moguće kombinacije ishoda njegovih roditelja. U prikazanom primjeru svi čvorovi imaju najviše jednog roditelja gdje svaki roditelj ima po dva ishoda – stoga ni broj pribrojnika u napisanim izrazima nije velik pa ni složenost računanja nije veliko. U općem slučaju važno je znati da broj kombinacija ishoda roditelja raste kombinatoričkom složnošću s obzirom na broj roditelja promatranog čvora.

Izračunate vrijednosti *a priori* očekivanja temelj su i za izračunavanje očekivanja u prisustvu dokaza. Stoga je korisno unaprijed izračunati sva *a priori* očekivanja u mreži.

3.2.2 Propagacija dokaza unaprijed kroz mrežu

Kada se u mreži pojave dokazi za neke čvorove, očekivanja svih ostalih čvorova se mijenjaju. Kako bi se izračunale vrijednosti tih očekivanja koriste se *aksiomi vjerojatnosti*. Pri tome je informaciju o postojanju dokaza potrebno propagirati i *unaprijed* kroz mrežu (od roditelja prema djeci) i *unatrag* kroz mrežu (od djece prema roditeljima).

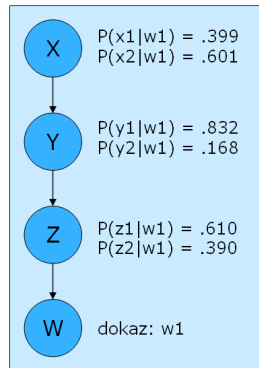
Slika 3.9 prikazuje primjer propagacije dokaza *unaprijed* kroz mrežu. Pri tome se u jednom čvoru koriste *a priori* očekivanja roditelja čvora i pripadna tablica združene distribucije vjerojatnosti.



Slika 3.9. Primjer propagacije dokaza unaprijed kroz mrežu. Poznavanje ishoda varijable X utječe na očekivanja ishoda čvorova Y, Z i W.

3.2.3 Propagacija dokaza unatrag kroz mrežu

Poznavanje ishoda nekog proizvoljnog čvora u mreži utječe i na očekivanja ishoda njegovih roditelja. U tom se slučaju može govoriti o propagaciji dokaza *unatrag* kroz mrežu.



Slika 3.10. Primjer propagacije dokaza unatrag kroz mrežu. Poznavanje ishoda varijable W utječe na očekivanja ishoda čvorova Z, Y i X.

Slika 3.10 prikazuje primjer propagacije dokaza unatrag kroz Bayesovu mrežu. U mreži je poznat dokaz o ishodu čvora W . Na temelju tablica združenih distribucija vjerojatnosti moguće je izračunati očekivanja ishoda svih ostalih čvorova u mreži. Pri tome se dokaz propagira unatrag kroz mrežu.

Očekivanja ishoda se prvo računaju za čvor Z koristeći Bayesov teorem:

$$P(z1|w1) = \frac{P(w1|z1)P(z1)}{P(w1)} = \frac{0.5 * 0.652}{0.5348} = 0.6096$$

Kako bi se izračunala tražena vrijednost bilo je potrebno znati:

- *A priori* očekivanja ishoda za čvorove Z i W , koja su izračunata ranije (odjeljak 3.2.1) i
- Vrijednost $P(w1|z1)$ koja je definirana u mreži (nije ju potrebno računati).

Sada se informacija o dokazu na čvoru W prosljeđuje i na čvor Y (preko čvora Z). Ponovno je potrebno upotrijebiti Bayesov teorem:

$$P(y1|w1) = \frac{P(w1|y1)P(y1)}{P(w1)}$$

U ovom izrazu poznate su sve potrebne vrijednosti osim $P(w1|y1)$. Ona se može izračunati koristeći aksiome vjerojatnosti:

$$\begin{aligned} P(w1|y1) &= P(w1|z1)P(z1|y1) + P(w1|z2)P(z2|y1) \\ &= 0.5 * 0.7 + 0.6 * 0.3 \\ &= 0.53 \end{aligned}$$

S ovom vrijednosti postaje moguće izračunati očekivanja za čvor Y :

$$P(y1|w1) = \frac{0.53 * 0.84}{0.5348} = 0.832$$

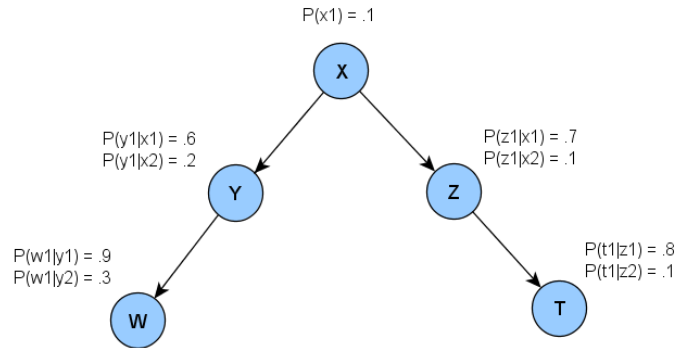
Sljedeći čvor na kojemu se izračunavaju očekivanja ishoda je čvor X . Kako bi se to omogućilo prethodno je potrebno izračunati vrijednosti:

- $P(w1|y2)$ te
- $P(w1|x1)$.

Iz opisanog je jasno da je propagacija unatrag u općem slučaju nešto zahtjevnija od propagacije unaprijed.

3.2.4 Propagacija unaprijed i unatrag

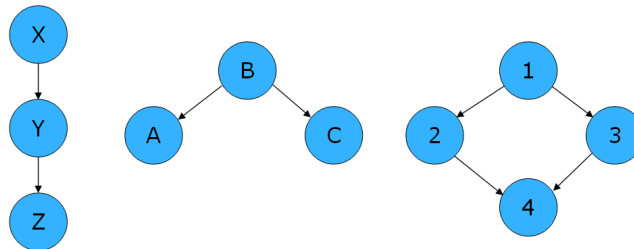
Sasvim je realna situacija u kojoj se propagacija dokaza događa kombinirano unaprijed i unatrag kroz mrežu. Slika 3.11 prikazuje primjer Bayesove mreže u kojoj se postavljanjem dokaza na čvor T događa propagacija dokaza unatrag kroz mrežu do čvora X a zatim i unaprijed kroz mrežu do čvora W .



Slika 3.11. Primjer kombinirane propagacije dokaza kroz mrežu. Poznavanje ishoda čvora T utječe na očekivanja ishoda svih ostalih čvorova u mreži.

3.2.5 Međusobna zavisnost čvorova

Prisutnost dokaza na nekim čvorovima u mreži izravno utječe na međusobnu zavisnost preostalih čvorova u mreži. Kada bi se predvidjela međusobna zavisnost čvorova u mreži (s obzirom na strukturu mreže i s obzirom na postojeće dokaze), moglo bi se uštedjeti vrijeme pri izračunu očekivanja ishoda čvorova. Naime, ako je poznato da su neka dva čvora u mreži međusobno nezavisna, tada pri postavljanju dokaza na jedan čvor nije potrebno trošiti vrijeme na izračunavanje očekivanja drugog čvora.



Slika 3.12. Primjeri Bayesovih mreža koji su indikativni u određivanju međuzavisnosti čvorova

Slika 3.12 prikazuje neke strukture Bayesove mreže koje ilustriraju međusobne zavisnosti čvorova u Bayesovoj mreži.

U prvom primjeru mreže na Slici 3.12 u situaciji kada nije poznat niti jedan dokaz čvorovi X i Z međusobno su zavisni: poznavanje jednog čvora utječe na očekivanje drugog čvora jer među njima postoji put preko čvora Y . Ako se otkrije dokaz za čvor Y tada čvorovi X i Z prestaju biti zavisni jer među njima nema više otvorenog puta. Piše se:

$$I \langle X, Y, Z \rangle.$$

Riječima rečeno, kada je poznat dokaz za čvor Y čvorovi X i Z su međusobno nezavisni.

U drugom primjeru na Slici 3.12 situacija je slična; razlika je u redefiniciji odnosa roditelj-djete između dva čvora. Promatrajući međusobnu zavisnost čvorova u ovoj mreži može se primjetiti da su čvorovi A i C bez poznatih dokaza u mreži međusobno zavisni a u prisutnosti dokaza na čvoru B međusobno nezavisni. Može se pisati:

$$I \langle A, B, C \rangle.$$

U trećem primjeru na slici 3.12 promatra se zavisnost čvorova 1 i 4 . Između njih postoje dva puta, preko čvora 2 i preko čvora 3 . Kako bi se postigla nezavisnost čvorova 1 i 4 potrebno je dokazima „presjeći“ oba puta među njima:

$$I \langle 1, 2, 4 \rangle - \text{NE vrijedi,}$$

$$I \langle 1, \{2, 3\}, 4 \rangle - \text{vrijedi.}$$

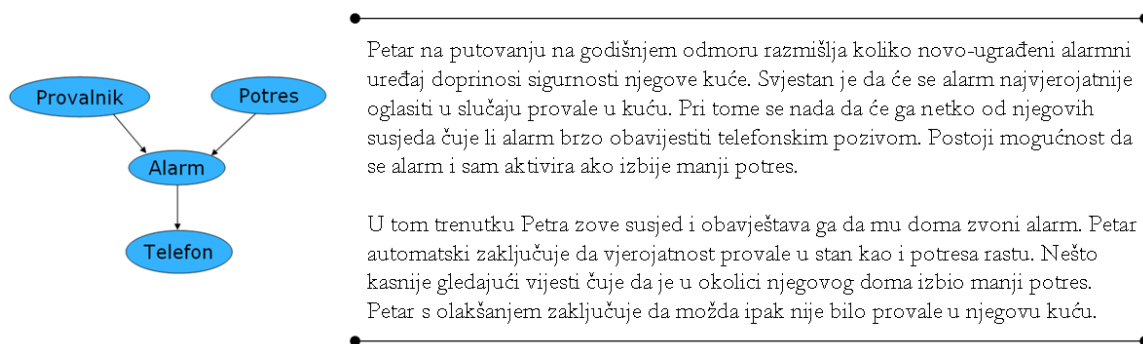
Po prikazanim primjerima mogao bi se steći dojam da je zavisnost između čvorova definirana jedino postojanjem dokaza na putevima među njima. Međutim, postoji slučaj kojeg je potrebno zasebno razmotriti.

Slika 3.13 prikazuje primjer u kojem međusobna zavisnost čvorova nije ovisna isključivo o čvorovima na putevima među njima. U prikazanom primjeru vrijede tvrdnje:

$$I \langle \text{Provalnik}, \{\}, \text{Potres} \rangle - \text{vrijedi}$$

$$I \langle \text{Provalnik}, \text{Alarm}, \text{Potres} \rangle - \text{NE vrijedi}$$

$$I \langle \text{Provalnik}, \text{Telefon}, \text{Potres} \rangle - \text{NE vrijedi}$$



Slika 3.13. Postoje slučajevi strukture u kojima međusobna zavisnost čvorova nije ovisna samo o putevima među njima.

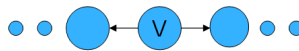
U interpretaciji riječima, bez poznavanja ikakvih dokaza u mreži varijable *provalnik* i *potres* međusobno su nezavisne. Međutim, ako se sazna dokaz za varijablu *alarm* one postaju zavisne (slično vrijedi i za varijablu *telefon*).

3.2.6 Algoritam d-separacije

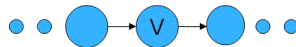
Algoritam d-separacije (engl. *d-separation*) je relativno jednostavan algoritam pomoću kojega se može u linearnom vremenu izračunati međusobna zavisnost čvorova.

Algoritam se sastoji od pravila koja otkrivaju da li je neki put između dva čvora blokiran ili nije. Postoji ukupno tri pravila:

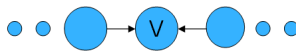
- I. Put između dva čvora je blokiran kada na njemu postoji čvor V za kojeg postoje dokazi a u kojem su lukovi puta okrenuti jedan drugom repovima:



- II. Put između dva čvora je blokiran kada na njemu postoji čvor V za kojeg postoje dokazi a u kojem su lukovi puta okrenuti jedan drugom glava prema repu:



- III. Put između dva čvora je blokiran kada na njemu postoji čvor V za kojeg ne postoje dokazi niti za bilo kojeg njegovog sljedbenika a u kojem su lukovi okrenuti jedan prema drugome glavama.



Pravila I i II govore da je neki put blokiran ako na njemu postoji dokaz. Pravilo III opisuje složeni slučaj zavisnosti među čvorovima (Slika 3.13).

Algoritam d-separacije moguće je pokretati već i prilikom dodavanja novih dokaza u sustav (ili pri brisanju dokaza) kako bi se unaprijed pripremila podloga za efikasnije rasuđivanje.

3.2.7 Algoritmi rasuđivanja

Opisani postupci propagacije dokaza kroz mrežu unaprijed i unatrag više su teoretske prirode jer se mogu primjeniti samo u slučajevima:

- kada je prisutan samo jedan dokaz u mreži i
- kada u mreži ne postoje višestruki putevi.

Za slučajeve višestrukih dokaza i višestrukih puteva u mreži može koristiti neki od poznatih algoritama za rasuđivanje u Bayesovoj mreži. Algoritme za rasuđivanje u Bayesovoj mreži generalno se može razdijeliti na egzaktne i stohastičke [56]:

- egzaktni:
 - *clustering* algoritam (Lauritzen, Spiegelhalter 1988), uz njegova poboljšanja (Jensen 1990, Dawid 1992),
 - *Pearl's message passing* algoritam (Pearl 1986),
- stohastički (uzorkovanje):
 - uzorkovanje unaprijed (*forward sampling*, Henrion 1988),
 - *maximum likelihood sampling* (Fung, Chang 1990),
 - *self-importance sampling* (Shachter, Peot 1990),
 - *heuristic-importance sampling* (Shachter, Peot 1990),
 - uzorkovanje unatrag (*backward sampling*, Fung, del Favero 1994),
 - *AIS-BN* (*adaptive-importance sampling*, Cheng, Druzdzel 2000),
 - *EPIS-BN* (*estimated posterior importance sampling*, Yuan, Druzdzel 2003).

U općem slučaju su egzaktni algoritmi po prirodi složeniji (a neki nemaju ni dostupan otvoreni kod). S druge strane, korištenje uzorkovanja ostavlja korisniku mogućnost odabira između trajanja izvođenja i točnosti rasuđivanja što može biti korisno u nekim primjenama. Dodatna pozitivna karakteristika uzorkovanja je ta što se uzorkovanje bez većih poteškoća može distribuirati među više računala na način da svako računalo zasebno generira jednu količinu uzoraka. Ovakva situacija danas je iznimno povoljna s obzirom na postojanje i dostupnost velikih računalnih mreža (npr. CRO GRID).

U ovoj doktorskoj disertaciji predlaže se predstavljanje znanja zasnovano na ontologijama i Bayesovim mrežama, pri čemu je (gledano sa probabilističke strane) u predloženoj metodologiji moguće koristiti bilo koji navedeni algoritam za rasuđivanje u Bayesovim mrežama. Međutim, pokazuje se da je zbog složenosti integriranog sustava mogućnost kompromisa pri uzorkovanju presudna. Naime, smanjenje točnosti rasuđivanja omogućuje donošenje zaključaka u realnom vremenu što je iznimno korisna osobina. Stoga se u sljedećem odjeljku daje uvod u osnovna načela uzorkovanja u Bayesovim mrežama.

3.2.8 Uzorkovanje

Na temelju Bayesove mreže moguće je generirati proizvoljno velik umjetan skup podataka (uzoraka) koji će statistički slijediti sve zavisnosti opisane u mreži. Pri tome je postupak generiranja uzoraka moguće uskladiti i sa svim poznatim dokazima u mreži. Ako je skup uzoraka dobiven uzorkovanjem dovoljno velik može se upotrijebiti za računanje očekivanja ishoda svih čvorova u mreži (tj. za rasuđivanje).

U Tablici 3.5 prikazan je primjer skupa uzoraka koji je generiran koristeći Bayesovu mrežu na Slici 3.13. Skup uzoraka u ovom primjeru sadrži ukupno četiri uzorka (četiri redka). Na temelju vrijednosti u uzorcima može se donijeti zaključak o očekivanjima ishoda varijabli kako je prikazano na dnu tablice. Očekivanje pojedinog ishoda jednako je proporciji (zastupljenosti) promatranog ishoda u skupu uzoraka.

Tablica 3.5. Primjer rezultata uzorkovanja za Bayesovu mrežu prikazanu na Slici 3.13.

Provalnik	Potres	Alarm	Telefon
ne	ne	ne	da
ne	da	da	da
da	ne	da	ne
ne	da	da	ne

zaključak: $P(\text{Provalnik}) = 0.25$ $P(\text{Potres}) = 0.5$ $P(\text{Alarm}) = 0.75$ $P(\text{Telefon}) = 0.5$

Kako bi se povećala točnost postupka rasuđivanja potrebno je povećati broj generiranih uzoraka.

Uzorkovanje unaprijed

Najjednostavniji algoritam za uzorkovanje jest algoritam *uzorkovanja unaprijed*. Kod uzorkovanja unaprijed se ishod određenog čvora (u promatranom uzorku) određuje slučajnim odabirom na temelju ranije generiranih ishoda njegovih roditelja u mreži. Stoga je prije samog postupka generiranja uzoraka potrebno poslagati čvorove u niz po topološkom redosljedju tako da se niti jedan čvor u nizu ne pojavljuje prije svojih roditelja. Pseudokod algoritma uzorkovanja je sljedeći:

Parametri:

N – broj čvorova u mreži

M – broj uzoraka

$E = \{E_1, E_2, \dots, E_N\}$ – skup dokaza u mreži

Algoritam:

posloži čvorove topološki da se dobije niz $X = \{X_1, X_2, \dots, X_N\}$

```
za  $m = 1..M$  { // generiranje M uzoraka
    za  $i = 1..N$  { // prođi kroz sve čvorove u mreži
        uzorkuj  $x_{mi}$  na temelju  $P(X_i | \text{roditelji}(X_i))$ 
        ako je  $x_{mi} \neq E_i$  { // ako se ne slaže sa dokazom
            odbaci i ponovo generiraj cijeli uzorak  $m$ 
        }
    }
}
```

Ako se u postupku generiranja jednog uzorka dobiveni ishod na nekom čvoru ne slaže sa pripadnim postavljenim dokazom tada se kompletan uzorak odbacuje te se umjesto njega pokušava ponavljanjem postupka načiniti novi uzorak. Dakle, *nije* dovoljno u uzorak unaprijed postaviti poznatu vrijednost dokaza jer se tada vrijednosti za preostale čvorove u mreži (čvorove bez dokaza) ne generiraju u ovisnostima o dokazima.

Kao primjer izvođenja prikazanog algoritma može se opisati postupak generiranja jednog uzorka na temelju Bayesove mreže prikazane na Slici 3.7 kada u mreži nisu poznati dokazi:

$P(ov) = 0.15$ -> slučajnim odabirom dobije se “**da**”

$P(pb) = 0,01$ -> slučajno odabran “ne”

$P(sulov) = 0.6$ -> slučajno odabran “da”

$P(pplov, \neg pb) = 0.9$ -> slučajno odabran “da”

$P(lalpp) = 0.7$ -> slučajno odabran “ne”

Dobiveni uzorak: $X_m = \{da, ne, da, da, ne\}$

U prikazanom primjeru odabrana vrijednost varijable *ov* (u prvoj liniji) u daljnjem postupku generiranja uzorka direktno utječe na odabir vrijednosti varijabli *su* i *pp*, a time i indirektno na varijablu *la*. Slične zavisnosti mogu se ustvrditi i za ostale čvorove u mreži. Ponavljanjem istog postupka može se dobiti proizvoljno velik skup uzoraka.

Najveća poteškoća kod uzorkovanja unaprijed jest eliminacija velikog broja uzoraka zbog neslaganja s dokazima. Stoga izvođenje uzorkovanja unaprijed traje u mnogim slučajevima neprihvatljivo dugo.

Može se zaključiti da uzorkovanje unaprijed predstavlja tek valjani matematički model generiranja uzoraka neprimjenjiv u praksi. Ipak, mnogi drugi algoritmi koji su korisni i u praktičnim primjenama samo su modificirani oblici uzorkovanja unaprijed.

Maximum likelihood uzorkovanje

Ovaj algoritam modificira algoritam uzorkovanja unaprijed tako da izbjegava generiranje loših uzoraka (pa se stoga niti jedan uzorak ne odbacuje). U postupku uzorkovanja se na čvorovima koji imaju postavljen dokaz automatski prepisuje vrijednost dokaza ali se pri tome dobiveni uzorak uzima sa određenom težinom. Pri tome vrijedi:

- Kada postoji samo jedan dokaz u mreži težina uzorka jednaka je vjerojatnosti odabira dokaza,
- Kada je dokaza u mreži više težine se za promatrani uzorak množe,
- Težine se moraju uzimati u obzir pri računanju očekivanja ishoda.

Tablica 3.6 prikazuje primjer generiranog skupa uzoraka pomoću *maximum likelihood* uzorkovanja za Bayesovu mrežu prikazanu slikom 3.13.

Tablica 3.6. Primjer skupa uzoraka koji se dobije *maximum likelihood* algoritmom uzorkovanja. Svaki generirani uzorak ima dodijeljenu težinu.

Provalnik	Potres	Alarm	Telefon	težina
ne	ne	ne	da	0.2
ne	da	da	da	0.1
da	ne	da	da	0.8
ne	da	da	da	0.1

$$\begin{aligned}
 P(\text{Provalnik}) &= & P(\text{Potres}) &= & P(\text{Alarm}) &= & P(\text{Telefon}) &= 1 & \Sigma &= 1.2 \\
 &= 0.8 / 1.2 & &= 0.2 / 1.2 & &= 1.0 / 1.2 & & \text{- dokaz -} & & \\
 &= 0.667 & &= 0.167 & &= 0.833 & & & &
 \end{aligned}$$

Pseudokod *maximum likelihood* algoritma uzorkovanja je sljedeći:

Parametri:

N – broj čvorova u mreži

M – broj uzoraka

E = {E₁, E₂, ..., E_N} – skup dokaza u mreži

Algoritam:

poslaži čvorove topološki da se dobije niz X = {X₁, X₂, ..., X_N}

```
za m = 1..M {           // generiranje M uzoraka
    wm = 1                // početna težina svakog uzorka je 1
    za i = 1..N {        // prođi kroz sve čvorove u mreži
        ako postoji Ei {
            xmi = Ei           // prepisi vrijednost dokaza
            wm *= P(Xi | roditelji(Xi)) // prilagodi težinu promatranog uzorka
        }
        inače uzorkuj xmi na temelju P(Xi | roditelji(Xi))
    }
}
```

Iz prikazanog je algoritma vidljivo da je za slučaj kada nema postavljenih dokaza u mreži *maximum likelihood* algoritam uzorkovanja identičan algoritmu uzorkovanja unaprijed.

U sljedećem poglavlju predlaže se metodologija predstavljanja znanja zasnovana na integraciji ontologija i Bayesovih mreža. U predloženoj metodologiji rasuđivanje je utemeljeno na osnovnim načelima rasuđivanja oba promatrana formalizma. U prvom su planu rasuđivanje pomoću tableau algoritma (kada u sustavu postoji dovoljno informacije za donošenje konačnih odluka) te algoritam uzorkovanja u Bayesovoj mreži (kada u sustavu nije prisutna sva potrebna informacija za donošenje konačnih odluka).

4 Integracija ontologija i Bayesovih mreža

U ovom poglavlju opisana je metodologija integracije ontologija i Bayesovih mreža. U početku se prikazuju osnovna načela izgradnje baze znanja pomoću ontologije te se daje uvod u problem nedostajućih vrijednosti pri donošenju odluka. Na taj način pokazuje se osnovna motivacija za integraciju sa Bayesovom mrežom. Zatim se kroz nekoliko primjera prikazuje oblik rezultata koji se očekuje od integriranog sustava. Na koncu se definira i sam postupak integracije te postupak rasuđivanja u integriranom sustavu.

4.1 Ontologije u sustavu za potporu pri odlučivanju

U ovom odjeljku se opisuje primjena ontologije za dizajn baze znanja. Pri tome će kao primjer poslužiti pojednostavljeni problem dijagnoze srčanog zatajenja. Opisuje se način na koji se ontologija koristi za opis domene srčanog zatajenja i za opis medicinskih pravila za dijagnozu srčanog zatajenja. Opisana metodologija ujedno daje i uvod u problem nedostajućih vrijednosti pri rasuđivanju na ontologijama.

4.1.1 Primjer problema

Pri dijagnozi srčanog zatajenja jedna od ključnih točaka je i dijagnosticiranje dva postojeća oblika srčanoga zatajenja: sistoličko i dijastoličko srčano zatajenje. Pri tome su mogući slučajevi kada pacijent ima istovremeno i sistoličko i dijastoličko zatajenje. Dijagnoza srčanog zatajenja i oblika srčanog zatajenja se obavlja pregledom pacijenta uz korištenje nekih dijagnostičkih testova (ehografija, EKG, rendgen prsnog koša itd.). Pri tome postoje neka medicinska dijagnostička pravila:

Dijastoličko srčano zatajenje je prisutno

ako pacijent ima neke od simptoma ili znakova srčanog zatajenja, i

ako je visoka razina omjera E/A, i

ako pacijent ima nizak dijastolički tlak.

Sistoličko srčano zatajenje se dijagnosticira na dva načina:

ako su prisuni neki simptomi ili znakovi srčanog zatajenja i

ako je niska razina ejekcijske frakcije lijeve klijetke srca

ili

ako su prisuni neki simptomi ili znakovi srčanog zatajenja i

ako je ubrzan ritam rada srca i

ako je nizak sistolički krvni tlak.

Simpomi srčanog zatajenja su: ubrzano lupanje srca, bol u prsima, otekli zglobovi, zadihanost, slabost, vrtoglavica, nemoć,

Znakovi srčanog zatajenja su: voda u plućima, treći i četvrti zvuk srca, povećanje volumena srca, povećanje volumena jetre....

Ova pravila predstavljaju medicinsko ekspertno znanje koje je potrebno integrirati u sustav za potporu pri odlučivanju. Ukoliko se u tome uspije, sustav bi morao moći na temelju uvida u aktualno stanje pacijenta samostalno primijeniti unešeno ekspertno znanje i donositi medicinski relevantne zaključke.

Korištenje pravila je uobičajeno u današnjim ekspertnim sustavima (medicinskim a i šire) [7], [15], [42]. Dakako, u mnogim situacijama ne postoje općenita pravila za dijagnozu – u takvim slučajevima nije moguće izgraditi ekspertni sustav temeljen na pravilima. Stoga je u ovom trenutku uvedena pretpostavka da postoji nekakvo eksplicitno ekspertno znanje koje se može formalizirati u ovom obliku. Kao što je ranije navedeno, ekspertno znanje se najčešće može naći u raznim stručnim priručnicima (smjernicama) ili se stječe direktno razgovorima sa ekspertima.

4.1.2 Opis domene pomoću ontologije

Jedna od osnovnih svojstava ontologije jest da pruža mogućnost opisa domene problema. Svrha opisa domene pomoću ontologije jest da se definiraju koncepti u domeni i relacije među njima, da se omogući strukturni pregled područja i da se eksplicitno izreknu pretpostavke koje su načinjene u domeni. Kada se načini opis domene postat će moguće izražavati ekspertno znanje (pravila) koristeći definirane koncepte u ontologiji. Domena gore opisanog problema jest srčano zatajenje; ontologija koja opisuje domenu srčanog zatajenja mora opisati sve koncepte koji se tiču srčanog zatajenja i relacije među njima [27], [38].

Teško je definirati jasna pravila pomoću kojih bi se trebali organizirati koncepti u nekoj domeni. Postoji mnogo različitih pristupa opisivanja, pri čemu neki specifičan pristup može biti prikladniji u svojem predviđenom okruženju a manje prikladan u drugom. Ipak, bez obzira na odabrani pristup izgradnje ontologije, važno je da se opis domene izrekne jednoznačno i jasno. Na koncu, poželjno je da je rezultirajuća ontologija intuitivno razumljiva barem od strane eksperata u domeni [28].

Snažan napredak u domeni semantičkog weba u novije vrijeme rezultirao je i brojnim novim ontološkim jezicima i alatima za opisivanje ontologija. Danas se može reći da je u među ontološkim jezicima u uporabi dominantan upravo OWL (engl. *Web Ontology Language*) te alat za uređivanje ontologija Protégé-OWL. Osim njihove popularnosti,

vrlo je važna komponenta i dostupnost alata te postojanje velikog broja razvijenih modula koji su u slobodnoj uporabi (u prvom redu su to razni programi koji nadograđuju Protégé-OWL alat). Stoga se u nastavku disertacije za opisivanje ontološke strukture koristi formalizam ontološkog jezika OWL (koji je zasnovan na semantici deskriptivne logike) te alat za oblikovanje ontologija Protégé-OWL [57]. Prikazani pristup dizajna baze znanja pomoću OWL ontologije je u jednom dijelu opisan i u [42].

U ontologiji srčanog zatajenja centralni su koncepti PATIENT i CHARACTERISTIC. Oni će omogućiti opis pojedinih pacijenata pomoću njihovih karakteristika. Koncept OWL:THING u OWL jeziku ekvivalentan je univerzalnom konceptu u deskriptivnoj logici (\top).

PATIENT \sqsubseteq OWL:THING

CHARACTERISTIC \sqsubseteq OWL:THING

Kako su simptomi srčanog zatajenja ujedno i karakteristike pacijenta jasno je da će koncept HFSYMP TOM biti definiran kao podklasa koncepta CHARACTERISTIC. Isto vrijedi i za znakove srčanog zatajenja.

HFSYMP TOM \sqsubseteq CHARACTERISTIC

HFSIGN \sqsubseteq CHARACTERISTIC

Sada se mogu navesti svi znakovi i simptomi srčanog zatajenja. Moguće je dakako, da neka individua bude istovremeno i znak i simptom srčanog zatajenja (koncepti nisu nužno međusobno disjunktni). Liste znakova i simptoma srčanog zatajenja moguće je i naknadno nadopunjavati sa novim elementima.

heartRateHigh : HFSYMP TOM

heartRateHigh : HFSIGN

anginaPectoris : HFSYMP TOM

SBPlow : HFSIGN

peripheralEdema : HFSYMP TOM

SBPhigh : HFSIGN

dyspnea : HFSYMP TOM

pulmonaryEdema : HFSIGN

...

...

U ontologiji valja također navesti kakve sve dijagnostičke pretrage pacijent može obaviti te što se sve može tim pretragama otkriti. Pojedini nalaz na nekom testu se također može smatrati karakteristikom pojedinog pacijenta.

TESTFINDING \sqsubseteq CHARACTERISTIC

decelerationHigh : ECHOFINDING

CHESTXRAYFINDING \sqsubseteq TESTFINDING

decelerationLow : ECHOFINDING

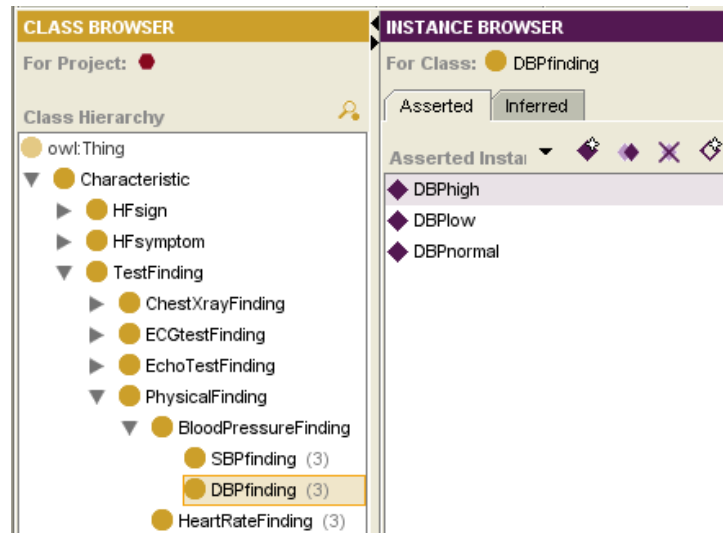
ECHOFINDING \sqsubseteq TESTFINDING

EAratioLow : ECHOFINDING

...

...

Na ovaj način je opisan dio domene srčanog zatajenja koji je malen ali dovoljno opširan za rješavanje gore opisanog pojednostavljenog oblika dijagnosticiranja. Trenutno stanje ontologije u Protégé alatu prikazuje Slika 4.1. Sa lijeve strane slike prikazani su gore opisani koncepti posloženi u hijerarhiju klasa, dok desna strana prikazuje sve navedene individue koje su instance trenutno odabrane klase (PHYSICALFINDING). Brojke u zagradama pokraj imena konceptata predstavljaju trenutni broj individua koje su instance pripadnog koncepta.



Slika 4.1. Opis domene srčanog zatajenja pomoću ontologije.

4.1.3 Opis pacijenta u ontologiji

Pojedini slučajevi pacijenata se također opisuju kao individue (koje su instance klase PATIENT). Pri tome se sve karakteristike pojedinog pacijenta opisuju relacijama na već postojeće individue u klasi CHARACTERISTIC. Evo dva primjera instanci pacijenata:

(marko, DBPlow) : HASCHARACTERISTIC

(marko, EARatioHigh) : HASCHARACTERISTIC

(marko, anginaPectoris) : HASCHARACTERISTIC

(pero, EARatioNormal) : HASCHARACTERISTIC

(pero, SBPlow) : HASCHARACTERISTIC

Kako se ovdje radi o opisu Abox-a dani opisi slijede pretpostavku otvorenog svijeta, što znači da je moguće da obje gore opisane instance pacijenta imaju još neke karakteristike koje nisu eksplicitno navedene. Tako npr. nije sasvim jasno ima li Marko visoku ili nisku razinu krvnih zrnaca; to je informacija koja u sustavu trenutno nedostaje. Takav je scenarij moguć ako pacijent još nije obavio laboratorijsku pretragu krvi.

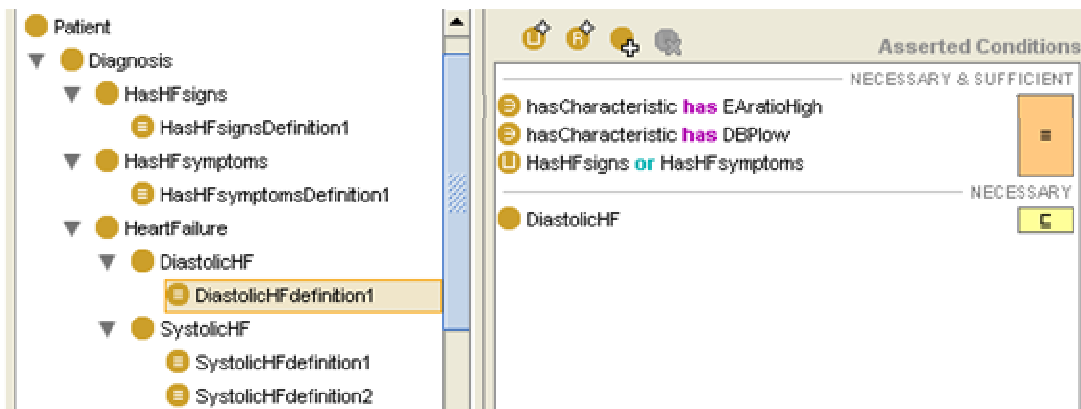
Važno je primjetiti da je upravo opis domene omogućio opisivanje vrlo složenih medicinskih situacija u kojima se pacijenti nalaze jednoznačno i na vrlo jednostavan način.

4.1.4 Opis znanja logičkim izrazima

Opis domene daje i podlogu za opis dijastoličkog i sistoličkog oblika srčanog zatajenja pomoću deskriptivne logike. Znanje se opisuje pomoću logičkih izraza:

$$\text{HASHFSIGN} \equiv \exists \text{HASCHARACTERISTIC.HFSIGN}$$
$$\text{HASHFSYMPATOM} \equiv \exists \text{HASCHARACTERISTIC.HFSYMPATOM}$$
$$\text{DIASTOLICHF} \equiv$$
$$(\text{HASHFSIGN} \sqcup \text{HASHFSYMPATOM})$$
$$\sqcap \exists \text{HASCHARACTERISTIC.EARATIOHIGH}$$
$$\sqcap \exists \text{HASCHARACTERISTIC.DBFLOW}$$
$$\text{SYSTOLICHF} \equiv$$
$$(\text{HASHFSIGN} \sqcup \text{HASHFSYMPATOM})$$
$$\sqcap \exists \text{HASCHARACTERISTIC.SBFLOW}$$
$$\sqcap \exists \text{HASCHARACTERISTIC.HEARTRATEHIGH}$$
$$\text{SYSTOLICHF} \equiv$$
$$(\text{HASHFSIGN} \sqcup \text{HASHFSYMPATOM})$$
$$\sqcap \exists \text{HASCHARACTERISTIC.LVEFLOW}$$

Ovako je definiran jedan način dijagnosticiranja dijastoličkog srčanog zatajenja te dva načina dijagnosticiranja sistoličkog srčanog zatajenja. Trenutno stanje ontologije u Protégé alatu prikazuje Slika 4.2. Lijeva strana slike prikazuje hijerarhiju klasa, dok desna strana prikazuje logički izraz koji definira označenu klasu u hijerarhiji klasa (dijastoličko srčano zatajenje).



Slika 4.2. Definiranje logičkih izraza za srčano zatajenje u Protégé-u.

S obzirom na definirane izraze i definirana stanja pacijenata, sigurno je da će se u fazi rasuđivanja pacijent *marko* svrstati u klasu DIASTOLICHF. Međutim, još uvijek se pacijent *pero*, iako ima karakteristiku *EAratioNormal*, neće svrstati u komplement klase DIASTOLICHF (što bi značilo da pacijent nema dijastoličko srčano zatajenje). Naime, u ontologiji još nigdje nije naznačeno da pacijent ne može istovremeno imati i *EAratioHigh* i *EAratioNormal*. Dakle, valja za sve pojmove koji su međusobno disjunktni to u ontologiji i naglasiti. Evo primjera:

$$\text{FINDOUTEARATIOHIGH} \equiv \exists \text{HASCHARACTERISTIC.EAratioHigh}$$

$$\text{FINDOUTEARATIOLOW} \equiv \exists \text{HASCHARACTERISTIC.EAratioLow}$$

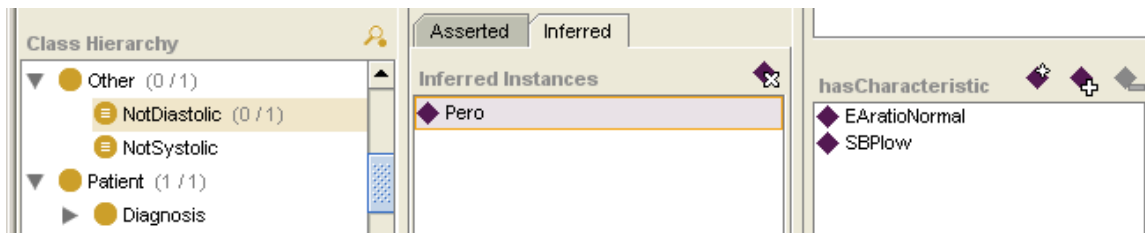
$$\text{FINDOUTEARATIONORMAL} \equiv \exists \text{HASCHARACTERISTIC.EAratioNormal}$$

$$\text{FINDOUTEARATIOHIGH} \sqcap \text{FINDOUTEARATIOLOW} \sqsubseteq \perp$$

$$\text{FINDOUTEARATIOHIGH} \sqcap \text{FINDOUTEARATIONORMAL} \sqsubseteq \perp$$

$$\text{FINDOUTEARATIONORMAL} \sqcap \text{FINDOUTEARATIOLOW} \sqsubseteq \perp$$

Riječima rečeno, ako bilo koja individua ima istovremeno dvije od tri mogućnosti za E/A omjer (visok, nizak, normalan) ontologija je *nekonzistentna*. Osim što ove tvrdnje omogućavaju pronalaženje nekonzistentnosti ontologije, one omogućavaju i prepoznavanje kada neki pacijent sigurno ne zadovoljava neki logički izraz. Primjerice, sada je moguće zaključiti da individua *pero* ni u kojem slučaju nije instanca koncepta DIASTOLICHF. Kako bi se uvjerali u to, može se načiniti novi koncept NOTDIASTOLICHF jednostavno negacijom koncepta DIASTOLICHF te pogledati pripada li individua *pero* u ovaj koncept. Rezultat rasuđivanja prikazuje Slika 4.3. Brojke u zagradama pokraj definicije koncepta predstavljaju broj ručno pridjeljenih instanci u konceptu i broj instanci pridjeljenih u koncept rasuđivanjem.



Slika 4.3. Svrstavanje individua pacijenata u klase.

Za slučaj sistoličkog zatajenja srca postoje dvije različite definicije. Pretpostavi li se da su to *jedine* dvije definicije pomoću kojih se dolazi do dijagnoze sistoličkog srčanog zatajenja, ispravno je zaključiti da pacijent nema sistoličko zatajenje srca ako pacijent sa sigurnošću ne može zadovoljiti niti jednu od dvije definicije (tj. da sigurno zadovoljava komplemente obe definicije). Da bi se takvo ponašanje ostvarilo i u ontologiji dovoljno je napisati da:

$$\text{SYSTOLICHF} \equiv \text{SYSTOLICDEFINITION1} \sqcup \text{SYSTOLICDEFINITION2}$$

$$\text{NOTSYSTOLIC} \equiv \neg\text{SYSTOLICHF}$$

Kada se za sve karakteristike pacijenta korištene u definicijama definiraju disjunktni skupovi (*visok, nizak, normalan*), procesom rasuđivanja se dobivaju rezultati prikazani u Tablici 4.1.

Tablica 4.1. Rezultati rasuđivanja za za neke specifične primjere pacijenata.

Karakteristike pacijenta	Klase u koje se svrstava
LVEFLOW, ANGINAPECTORIS	SYSTOLICDEFINITION1, SYSTOLICHF
HEARTRATEHIGH, SBPLOW, ANGINAPECTORIS	SYSTOLICDEFINITION2, SYSTOLICHF
LVEFNORMAL, SBPNORMAL	NOTSYSTOLICHF
ANGINAPECTORIS	niti SYSTOLIC, niti NOTSYSTOLICHF

Vrlo je zanimljiv slučaj u posljednjem retku Tablici 4.1. Pacijent koji nema poznate vrijednosti za *LVEF*, *SBP* i *heartRate* nije svrstan niti u klasu SYSTOLICHF a niti u njen komplement. To je sasvim logičan zaključak, s obzirom da nema dovoljno podataka da se donese zaključak; pacijentu se ne može odrediti dijagnoza prije nego što se obave za dijagnozu potrebni testovi.

4.2 Motivacija za integraciju sa Bayesovom mrežom

Nedostatak deskriptivne logike (a stoga i OWL ontologije) jest što u slučajevima nedostajućih vrijednosti ne postoji mogućnost otkrivanja najvjerojatnijeg zaključka niti akcije koja će najefikasnije voditi ka konačnom zaključku. Iako je ponekad moguće izravno iz definiranog logičkog izraza pročitati varijable koje definiraju logički izraz, u općem slučaju situacija izgleda mnogo složenije.

U medicinskoj domeni su situacije u kojima nedostaju određeni podaci mnogo češći nego situacija sa poznatim svim mogućim podacima (npr. pri prvom posjetu liječniku poznato je vrlo malo informacije o pacijentu). U takvim slučajevima je vrlo korisno znati koje su akcije (tipično medicinski testovi) najinformativnije. Kao primjer može se pogledati sljedeći logički izraz:

$$\begin{aligned} \text{DIASTOLICHF} &\equiv \\ &(\text{HASHFSIGN} \sqcup \text{HASHFSYMPTOM}) \\ &\sqcap \exists \text{HASCHARACTERISTIC.EARATIOHIGH} \\ &\sqcap \exists \text{HASCHARACTERISTIC.DBLOW} \end{aligned}$$

Isčitavanjem ovog izraza može se zaključiti da je pri dijagnozi dijastoličkog srčanog zatajenja korisno znati ima li pacijent neke znakove i simptome srčanog zatajenja, koja je razina E/A omjera i koliki je krvni tlak pacijenta (u mirovanju). U stvarnom svijetu liječnik je svjestan da su problemi sa razinom E/A prisutni kod oko 5% populacije pacijenata, da je nizak dijastolički tlak prisutan kod oko 20% populacije, te da se neki oblik znakova i simptoma srčanog zatajenja javlja kod oko 40% populacije. S tim u vidu i s obzirom na definirani logički izraz jasno je da mjerenje E/A omjera vodi konačnoj odluci kod većeg dijela populacije.

U ovom slučaju može se reći da je mjerenje E/A razine *informativnije* od mjerenja dijastoličkog krvnog tlaka ili provjere znakova ili simptoma srčanog zatajenja.

Međutim, prije ovakvog zaključka neki čimbenici se također moraju razmotriti:

- Analiza informativnosti se mora provesti u ovisnosti o trenutno poznatim podacima promatranog pacijenta,
- Varijable veoma često imaju međusobne zavisnosti (npr. krvni tlak i starosna dob pacijenta),
- Medicinske pretrage redovito obavljaju više od jednog mjerenja istovremeno (npr. sistolički i dijastolički krvni tlak),
- Neka medicinska mjerenja se mogu provesti na više različitih medicinskih pretraga (npr. brzina otkucaja srca se može mjeriti i na fizičkom pregledu i na EKG-u),

Medicinski koncepti korišteni u logičkim izrazima su definirani preko drugih koncepata u ontologiji i relacija među njima (ontološka struktura). Primjerice, postoje mnogi različiti znakovi srčanog zatajenja i simptomi srčanog zatajenja; svaki pojedini element iz takvog skupa ima pridjeljenu i listu medicinskih pretraga pomoću koje se može otkriti. Analiza informativnosti mora predvidjeti i ovakve raznovrsne strukture medicinskih koncepata definiranih u ontologiji.

U mediciskoj praksi veoma često i neki drugi (ne-medicinski) faktori također igraju važnu ulogu, npr. cijena obavljana pretrage, trajanje pretrage, dostupnost mediciskog uređaja, hitnost, itd. Iako mjera informativnosti pretrage ne uključuje ovakve čimbenike, moguće je načiniti težinsku kombinaciju (ili neki drugi oblik) ovih čimbenika kako bi se stvorila jedna unifirmna skala korisnosti medicinske pretrage. Međutim, ovakva analiza nije obrađena u sklopu ove disertacije; ona analizira samo informativnost medicinskih pretraga u čistom medicinskom smislu.

4.2.1 Ciljevi integracije

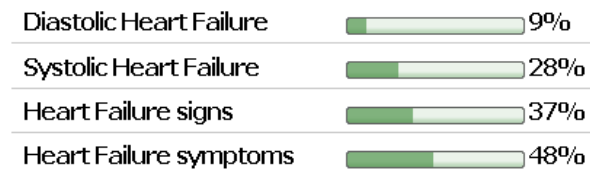
Shodno opisanom problemu mogu se definirati dva generalna cilja koji se žele postići integracijom ontologije i Bayesove mreže. To su:

- Računanje očekivanja ciljnih koncepata definiranih u ontologiji i
- Računanje informacijske dobiti (informativnosti) dijagnostičkih testova definiranih u ontologiji.

4.2.2 Očekivanja ciljnih koncepata

Kada se koncepti definirani logičkim izrazima u ontologiji integriraju sa Bayesovom mrežom postat će moguće izračunati koja su očekivanja tih koncepata s obzirom na sve trenutno postojeće činjenice (dokaze) u sustavu. Primjerice, moći će se izračunati koliko je očekivanje dijastoličkog srčanog zatajenja u situacijama kada nema dovoljno dokaza za konačnu odluku s obzirom na sve dosad poznate značajke pacijenta.

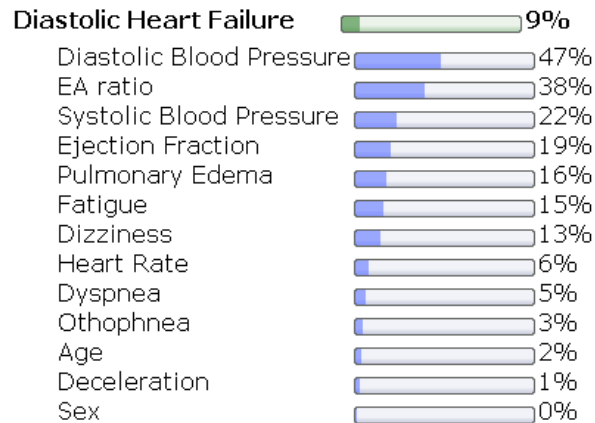
Zanimljivo je i međusobno uspoređivati dobivene vrijednosti očekivanja ciljnih koncepata. Slika 4.4 prikazuje primjer izračunatih rezultata očekivanja.



Slika 4.4. Očekivanja ciljnih koncepata pri dijagnozi srčanog zatajenja.

Ovakav način usporedbe očekivanja vrlo je sličan medicinskom procesu diferencijalne dijagnoze: uspostavljanje dijagnoze u slučajevima kada simptomi upućuju na više različitih bolesti. Pojmovi prikazani na slici nisu nužno međusobno disjunktivi.

Samom integracijom pojmova u Bayesovu mrežu se ujedno dobiva i mogućnost otkrivanja informativnosti pojedinih medicinskih elementarnih mjerenja. Slika 4.5 prikazuje primjer u kojem su informativnosti pojedinih pretraga poslagane po iznosu informativnosti.



Slika 4.5. Informacijska dobit dijagnostičkih čvorova za ciljni koncept DIASTOLIC HF za pacijenta koji nije obavio još nijednu medicinsku pretragu.

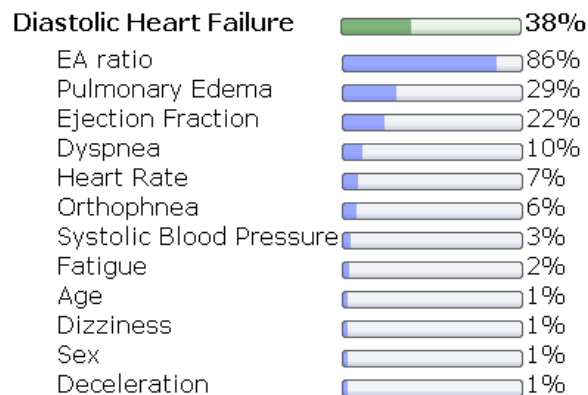
Na Slici 4.5 zelenom je bojom označena vrijednost očekivanja ciljnog ishoda a plavom bojom vrijednost informativnosti pojedinih pretraga. Iz slike se vidi da je za konačnu dijagnozu dijastoličkog srčanog zatajenja najinformativnije načiniti pretragu mjerenja dijastoličkog krvnog tlaka i E-A omjera u srcu. To je i očekivano s obzirom da su ta dva pojma izravno navedena u definiciji dijastoličkog srčanog zatajenja u ontologiji. Nadalje, iako sistolički tlak nije izravno naveden u definiciji dijastoličkog zatajenja, on i dalje nosi određenu količinu informacije s obzirom da je posredno povezan sa ciljnim konceptom preko dijastoličkog krvnog tlaka. Kada liječnik sazna iznos sistoličkog krvnog tlaka, dobiva i nešto jasniju predodžbu dijastoličkog krvnog tlaka pa stoga i dijastoličkog srčanog zatajenja.

4.2.3 Slijed odluka temeljen na informacijskoj dobiti

Nakon što je dobio obavijest da očekivanje dijastoličkog srčanog zatajenja iznosi 9%, liječnik saznaje i da je mjerenje dijastoličkog tlaka po tom pitanju najinformativnija pretraga. Ako liječnik uistinu i obavi mjerenje dijastoličkog krvnog tlaka, moguća su dva odvojena scenarija.

Prvi scenarij je da pacijent nema nizak krvni tlak što otklanja mogućnost postojanja dijastoličkog srčanog zatajenja (zbog definicije logičkog izraza). Pokrenuvši postupak rasuđivanja u ontologiji pacijent se svrstava u klasu NOTDIASTOLICHF.

Drugi scenarij je da pacijent ima nizak dijastolički krvni tlak, što povećava očekivanje dijastoličkog srčanog zatajenja na 38%. Kako bi došao do konačnog odgovora liječnik bi mogao obaviti i neke dodatne pretrage. Pri tome se opet može izvršiti analiza informativnosti pretraga. Slika 4.6 prikazuje rezultate analize informativnosti za promatranog pacijenta nakon što je obavio pretragu dijastoličkog krvnog tlaka.



Slika 4.6. Analiza informativnosti nakon izmjerenog niskog krvnog tlaka.

Nakon izmjerenog dijastoličkog krvnog tlaka, mjerenje E-A omjera dobiva posebno na važnosti (informativnost je skočila sa 38% na 86%). To se dogodilo zbog toga što se uklonila dotada najinformativnija pretraga (dijastolički krvni tlak), te ostaje jedna varijabla manje za promatrati kako bi se došlo do konačnog odgovora.

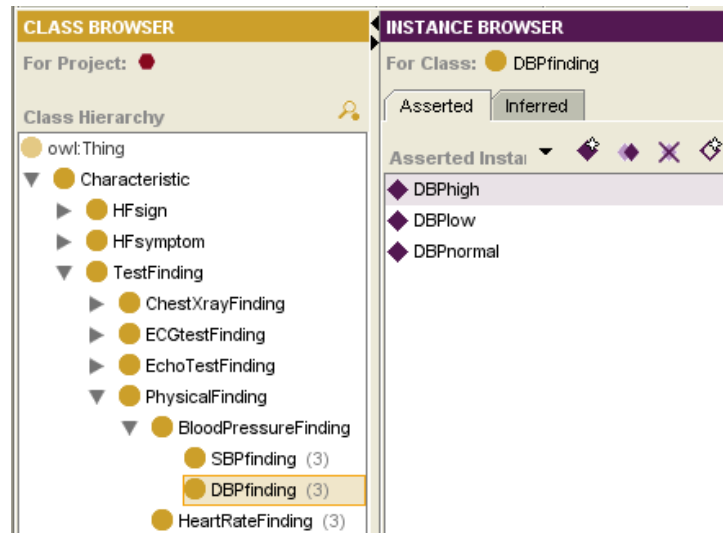
Slika 4.6 također pokazuje da mjerenje ejekcijske frakcije nosi vrlo visoku informaciju (22%). S obzirom da je E-A omjer koreliran sa ejekcijskom frakcijom porast informativnosti jednog mjerenja prati porast informativnosti drugog.

Nadalje, vidi se da mjerenje sistoličkog krvnog tlaka sada ne nosi više tako mnogo informacije (3%) kao što je nosilo dok dijastolički krvni tlak nije bio poznat (22%). To je također očekivan rezultat s obzirom da je varijabla preko koje sistolički tlak utječe na konačan ishod razotkrivena (dijastolički tlak).

4.2.4 Informacijska dobit grupiranih pretraga

Valja primjetiti da u stvarnom svijetu postoje mjerenja koja se nikada ne obavljaju odvojeno. Gotovo je nemoguće izmjeriti sistolički a da se pri tome ne izmjeri i dijastolički krvni tlak.

Ontologija pruža mogućnost da se sve moguće pretrage organiziraju u grupe kako se one doista i obavljaju u stvarnom svijetu. Primjerice, jasno je da se mjerenjem krvnog tlaka mjere istovremeno sistolički tlak, dijastolički tlak i brzina otkucaja srca; da se na EKG-u može ustanoviti brzina otkucaja srca, aritmija i razni šumovi na srcu; da se ehokardiografijom mjeri ejectionna frakcija, E-A omjer i povećan volumen srca itd. U ontologiji se pretrage jednostavno grupiraju svrstavanjem individua u klase kao što je prikazano na Slici 4.7.



Slika 4.7. Ontologija pruža mogućnost jednostavnog organiziranja pretraga u klase.

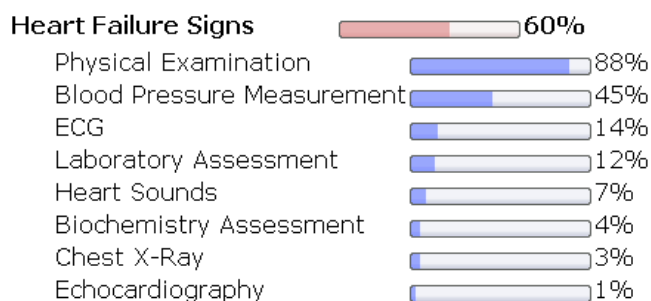
Desna strana Slike 4.7 prikazuje koje se sve informacije mogu otkriti mjerenjem krvnog tlaka. Osim što se mjerenje tlaka može gledati kao samostalna pretraga, ono je istovremeno i sastavni dio jedne veće (opsežnije) pretrage koja se naziva fizikalni pregled (Slika 4.7 lijevo). Na ovaj način moguće je pretrage u ontologiji vrlo jednostavno organizirati u grupe i podgrupe. Dakako, moguće je da jedna pretraga bude obavljena na dva potpuno različita pregleda; primjerice, ubrzan rad srca može se otkriti i na fizičkom pregledu i na EKG-u. Grupe pretraga nisu nužno međusobno disjunktne.

Tablica 4.2 prikazuje elementarne pretrage u svakoj od definiranih grupa u ontologiji. Sa lijeve strane u tablici navedeno je ime grupirane pretrage a sa desne strane popis svih pretraga koji je sačinjavaju. Pretrage navedene u desnom stupcu a koje također sadrže podpretrage napisane su u kurzivu.

Kada se učini analiza informativnosti svih definiranih grupiranih pretraga za ciljni pojam HFSIGNS može se saznati koja je u tom pogledu pretraga najinformativnija. Slika 4.8 prikazuje poredak medicinskih pretraga po informativnosti za navedeni ciljni koncept.

Tablica 4.2. Medicinske grupirane pretrage koje se definiraju u ontologiji.

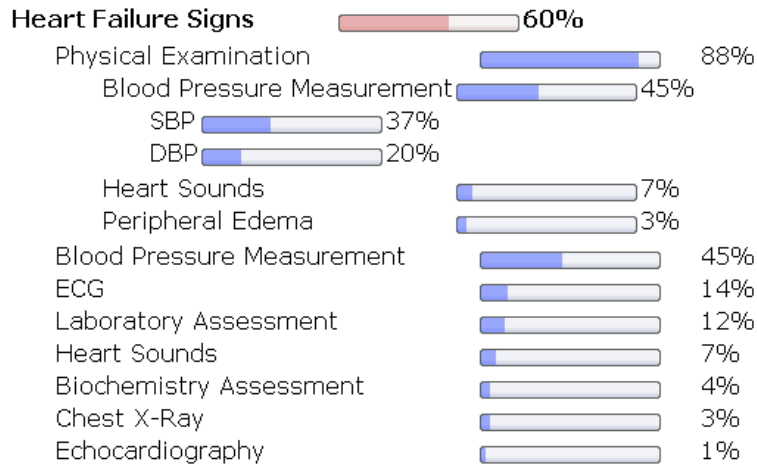
Grupirana pretraga	Elementarne pretrage
Physical examination	Age, Sex, Dyspnea, Fatigue, Othophnea, Dyspnea, Peripheral Edema, Pulmonary Rales, Chest Pain, Arrhythmia, Bradycardia, Tachycardia, Heart Rate, Jugular Vein Congestion, <i>Heart Sounds, Blood Pressure Measurement</i>
Blood pressure measurement	Systolic Blood pressure, Diastolic blood pressure
Heart Sounds	S3 heart sound, S4 heart sound
ECG	Heart rate, Q waves, ST depression, T wave, <i>Conduction</i>
Conduction	LBBB, RBBB, PQ duration, QRS duration, QT duration
Chest X-ray	Pulmonary edema, Cardiothoracic ratio
Echocardiography	Left ventricular ejection fraction, E-A ratio, Akinesis, Dyskinesis, Hypokinesis, Deceleration, Contractility
Laboratory Assessment	BNP, <i>Biochemistry, Complete Blood Count</i>
Biochemistry	Glucose, Creatinine, Creatinine clearance, Potassium, Sodium, Urea, Cholesterol, TSH, ALT, AST
Complete Blood Count	RBC count, WBC count, Hemoglobine, Hematocrite



Slika 4.8. Informativnosti grupiranih pretraga za ciljni pojam HFSIGNS.

Na Slici 4.8 je zanimljivo primjetiti da je pretraga mjerenja krvnog tlaka jedna od podpretraga koja se vrši unutar standardnog fizikalnog pregleda, te stoga mora imati i manju informativnost od potpunog fizikalnog pregleda.

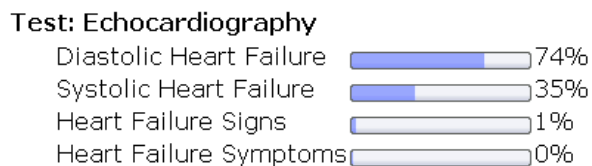
S obzirom da su testovi definirani ontološkom strukturom moguće je za svaki test pogledati koji njegovi podtestovi nose najveći dio informacije. Slika 4.9 prikazuje analizu informativnosti podtestova fizičkog pregleda, a unutar nje i analizu podtestova mjerenja krvnog tlaka.



Slika 4.9. Pregled informativnosti testova po strukturi ontologije.

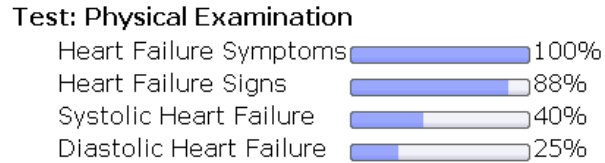
Učinak jedne pretrage na sve ciljne pojmove

Sada je moguće i sumarno vidjeti na koji način obavljanje jedne grupirane pretrage doprinosi svim definiranim ciljnim pojmovima. Slika 4.10 prikazuje koliko se informacije stječe obavljanjem pretrage ehokardiografije za svaki pojedini ciljni pojam. Na slici se vidi da ehokardiografija nosi značajnu informaciju pri dijagnozi dijastoličkog i sistoličkog srčanog zatajenja, ali i da ne igra značajnu ulogu pri razotkrivanju znakova i simptoma srčanog zatajenja.



Slika 4.10. Utjecaj (informativnost) ehokardiografije na sve definirane ciljne pojmove u ontologiji.

Na isti način se mogu obaviti analize za sve ostale grupirane pretrage definirane u ontologiji. Slika 4.11 prikazuje informativnost fizičkog pregleda na ciljne pojmove u ontologije. Fizički pregled ima značajnu ulogu za otkrivanje znakova i simptoma srčanog zatajenja, te nešto manju ali ne i beznačajnu ulogu u dijagnozi dijastoličkog i sistoličkog srčanog zatajenja.



Slika 4.11. Utjecaj fizičkog pregleda na ciljne pojmove definirane u ontologiji.

Kada se liječniku prikažu ovakve strukture izračunatih informativnosti opet je njemu prepuštena odluka koja će se pretraga (ili slijed) uisitnu i obaviti. Sustav za donošenje odluka nema mogućnost samostalnog donošenja odluka ali ima značajne mogućnosti da u takvim odlukama pripomogne.

4.3 Izgradnja i uporaba Bayesove mreže

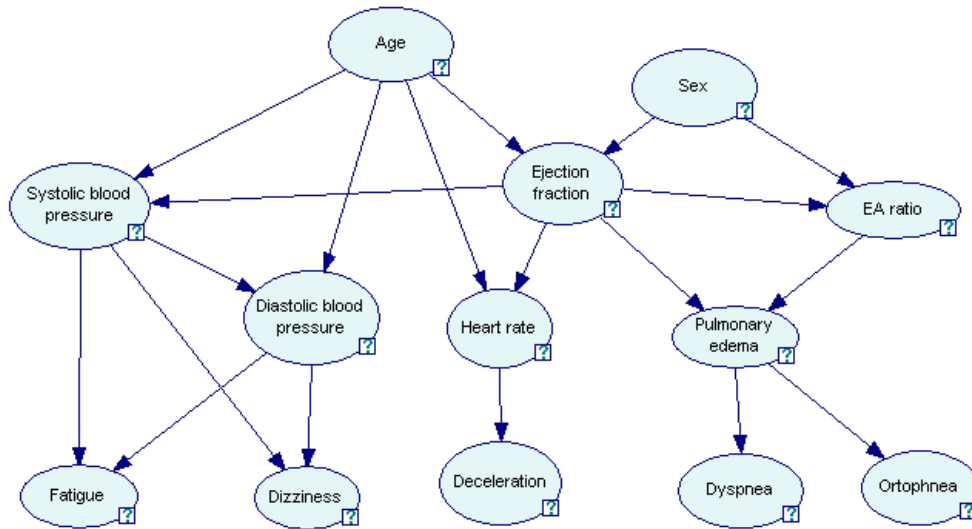
Bayesova mreža pruža prikladan okvir za modeliranje zavisnosti među varijablama. Zavisnosti se modeliraju pomoću mrežne strukture i skupa uvjetnih vjerojatnosti. Za definirati jednu Bayesovu mrežu potrebno je definirati čvorove (ili varijable, npr. LVEF), moguće vrijednosti koje pojedini čvor može poprimiti (npr. visok, nizak, normalan), veze između čvorova i vrijednosti uvjetnih vjerojatnosti u čvorovima.

U ovoj doktorskoj disertaciji korišten je alat za uređivanje Bayesovih mreža GeNIe&SMILE [56]. SMILE (engl. *Structural Modeling, Inference and Learning Engine*) sadrži implementaciju algoritama za izgradnju, rasuđivanje i učenje na Bayesovim mrežama dok GeNIe (engl. *Graphical Network Interface*) pruža grafičko korisničko sučelje za jednostavniju izgradnju i evaluaciju Bayesovih mreža. Za potrebe ove doktorske disertacije ovaj je alat jednako pogodan kao i mnogi drugi alati [59] (npr. Hugin [60]) a odabran je zbog njegove dostupnosti, jednostavnosti korištenja iz programskog koda i velikog broja ugrađenih algoritama za rasuđivanje.

Bayesovu mrežu je moguće konstruirati (1) ručno kroz postupke sakupljanja znanja u interakciji sa ekspertima iz domene ili (2) automatski postupkom strojnog učenja iz dostupnih skupova podataka. Pri tome je u strojnom učenju moguće odvojeno učiti strukturu i uvjetne vjerojatnosti u čvorovima.

Pristup strojnim učenjem je posebno prikaldan u medicinskoj domeni gdje je često veoma teško eksplicitno izreći ekspertno znanje, dok sa druge strane već postoji velika količina dostupnih medicinskih skupova podataka koji su sakupljeni za potrebe istraživanja. Osim toga, pristizanjem novih podataka i novih instanci pacijenata znanje u mreži se može dodatno modificirati i poboljšavati. U slučaju potpune promjene radne sredine (primjerice ako se sustav preseli u drugu državu ili se počne primjenjivati na drugačijoj ciljnoj populaciji pacijenata) moguće je načiniti novu mrežu učenjem na novom (pripadnom) skupu podataka.

Slika 4.12 prikazuje provizorni primjer Bayesove mreže koja je sagrađena ručno (bez uporabe algoritama strojnog učenja) a koja služi u ovoj disertaciji samo kao primjer za demonstraciju metodologije. Mreža definira ovisnosti među nekim ključnim pojmovima u domeni srčanog zatajenja.



Slika 4.12. Bayesova mreža koja prikazuje povezanost između nekih ključnih pojmova u domeni srčanog zatajenja.

Dakako, na Slici 4.12 je prikazana samo sturktura Bayesove mreže. Ono što nije prikazano na slici su ishodi koje svaki pojedini čvor u mreži može poprimiti i tablice uvjetnih vjerojatnosti definirane u svakom pojedinom čvoru. Primjerice, za čvor SEX je definirano da može poprimiti vrijednosti {MALE, FEMALE}, čvor EJECTION FRACTION može poprimiti vrijednosti {NORMAL, LOW}, itd. Tablica 4.3 prikazuje primjer tablice uvjetnih vjerojatnosti definirane za čvor EJECTION FRACTION.

Tablica 4.3. Definicija uvjetnih vjerojatnosti u čvoru EJECTION FRACTION.

		AGE	YOUNG		OLD	
		SEX	MALE	FEMALE	MALE	FEMALE
EJECTION FRACTION	NORMAL	0.85	0.9	0.6	0.65	
	LOW	0.15	0.1	0.4	0.35	

Tablica 4.3 je zapravo sažeti oblik zapisa sljedećih tvrdnji:

$$P(\text{EF} = \text{NORMAL} \mid \text{YOUNG, MALE}) = 0.85$$

$$P(\text{EF} = \text{NORMAL} \mid \text{YOUNG, FEMALE}) = 0.9$$

$$P(\text{EF} = \text{NORMAL} \mid \text{OLD, MALE}) = 0.6$$

...

Na temelju definicije čvorova mogu se izračunati *a priori* vjerojatnosti svih vrijednosti svih čvorova u mreži. Tako se u ovom konkretnom slučaju izračunaju sljedeće vrijednosti:

Čvor EJECTION FRACTION

$$P(\text{NORMAL}) = 0.705 \quad P(\text{LOW}) = 0.295$$

Čvor DIASTOLIC BLOOD PRESSURE

$$P(\text{HIGH}) = 0.393 \quad P(\text{NORMAL}) = 0.412 \quad P(\text{LOW}) = 0.195$$

Čvor HEART RATE

$$P(\text{NORMAL}) = 0.564 \quad P(\text{HIGH}) = 0.436$$

...

Ove vjerojatnosti predstavljaju vrijednosti s kojima liječnik može očekivati ishode pojedine varijable bez da zna ikakve podatke o trenutnom pacijentu kojeg razmatra (*a priori*). Drugim riječima, kada se pojavi neki potpuno novi pacijent o kojemu liječnik još apsolutno ništa ne zna, liječnik očekuje da će taj pacijent u 70% slučajeva imati normalnu razinu ejekcijske frakcije, da će u 40% slučajeva imati povišeni dijasolički krvni tlak te da će u 56% slučajeva imati normalnu brzinu otkucaja srca.

Ako u međuvremenu liječnik sazna još neke konkretne podatke o pacijentu tada se očekivanja ishoda čvorova za ovog konkretnog pacijenta mijenjaju. Primjerice, ako se zna da je pacijent osoba muškog spola i u podmakloj životnoj dobi očekivanja ishoda čvorova su:

Čvor EJECTION FRACTION

$$P(\text{NORMAL}) = 0.6 \quad P(\text{LOW}) = 0.4$$

Čvor DIASTOLIC BLOOD PRESSURE

$$P(\text{HIGH}) = 0.47 \quad P(\text{NORMAL}) = 0.305 \quad P(\text{LOW}) = 0.225$$

Čvor HEART RATE

$$P(\text{NORMAL}) = 0.44 \quad P(\text{HIGH}) = 0.56$$

Dakle, ovisno o trenutno poznatim informacijama o pacijentu, liječnik saznaje koja su očekivanja ostalih (trenutno nepoznatih) karakteristika pacijenta.

4.3.1 Učenje Bayesove mreže iz skupa podataka

HEARTFAID konzorciju je ljubaznošću organizacije talijanskih liječnika ANMCO (L'Associazione Nazionale Medici Cardiologi Ospedalieri) dostupan skup podataka o pacijentima koji su imali problema sa srcem, a u prvom redu sa srčanim zatajenjem. Skup uključuje oko 18,000 pacijenata iz talijanskih bolnica. Svaki pacijent je opisan sa 84 atributa od kojih su neki sakupljeni iz nalaza pacijenata (krvni tlak, ejekcijska frakcija, E/A omjer...), a neki opisuju događaje kod pacijenta (životni vjek, kvaliteta života, hospitalizacije, prenosnice, transplantacije...). Ovaj skup podataka daje osnovu za učenje Bayesove mreže iz podataka. Učenje Bayesove mreže u okviru ove disertacije je također obavljen u programskom paketu GeNIe&SMILE.

Osnovna poteškoća koja se pojavljuje prilikom učenja Bayesove mreže je problem nedostajućih vrijednosti u skupu podataka. Ovaj je problem općenito dobro poznat u područjima strojnog učenja i dubinske analize podataka a najčešći pristupi rješavanju ovog problema su:

- brisanje instanci sa nedostajućim vrijednostima (ili nekim postotkom nedostajućih vrijednosti),
- brisanje atributa sa nedostajućim vrijednostima (ili nekim postotkom nedostajućih vrijednosti),
- nadomještanje sa najčešćom (ili nekom očekivanom) vrijednosti u skupu,
- nedostajući podatak poprima jedan oblik vrijednosti atributa (npr. sa {Da, Ne} se prelazi na {Da, Ne, Nepoznato}),
- pogađanje vrijednosti na temelju ostalih značajki instance (engl. *educated guess*).

Dobro je primjetiti da u ovom konkretnom slučaju skupa podataka (medicinska domena) svaka nedostajuća vrijednost može nositi informaciju. Naime, sama činjenica da neki test na pacijentu nije obavljen daje za naslutiti da liječnik nije smatrao da je potrebno izvršiti taj test, te shodno tome pacijent ima vrlo visoku vjerojatnost normalne vrijednosti testa. Stoga bi se nedostajuća vrijednost mogla zamijeniti i sa normalnom vrijednosti pojedinog testa (a ne sa najčešćom ili očekivanom vrijednosti). Kako bi se ovakva pretpostavka mogla uvesti, potrebno je poznavati način na koji je provedeno samo sakupljanje podataka.

S obzirom da je skup podataka dovoljno velik, najjednostavnije i najefikasnije je primjeniti postupak uklanjanja atributa i instanci sa nedostajućim vrijednostima iz skupa podataka. Za početak je potrebno izbaciti *attribute* koji nemaju nikakvu informaciju u sebi i shodno tome samo smetaju u analizi. Primjerice, svaki pacijent ima svoj ID kojeg je nužno izbaciti iz skupa podataka. Zatim je potrebno iz skupa podataka izbaciti attribute sa velikim postotkom nedostajućih vrijednosti. Prema vlastitom nahođenju granicu za postotak popunjenosti atributa možemo postaviti na 50%. Dakle, svi atributi koji imaju

imaju više od 50% nedostajućih vrijednosti se brišu. Ovo je relativno razumna mjera s obzirom da nekolicina atributa ima popunjenost manju od 10%; takvi atributi teško mogu doprinijeti kvaliteti strojnog učenja.

Nakon toga se iz skupa podataka izbacuju sve instance koje imaju visok postotak nedostajućih vrijednosti (također po osobnom nahođenju granica se postavlja na maksimalno 4 nedostajuće vrijednosti). Ovo je nešto drastičnija mjera s obzirom da se na ovaj način uklanja velik dio potencijalno korisnih podataka. Umjesto izbacivanja instanci moglo se obaviti i pogađanje vrijednosti na temelju ostalih značajki instance ili postavljanje na najčešću, očekivanu, ili normalnu vrijednost. Međutim, izbacivanje instanci s nedostajućim vrijednostima je djelomično opravdano s obzirom da i poslije izbacivanja ostaje relativno velik broj pacijenata za izvođenje pokusa (~4200 prijmjera sa 30 atributa).

Nakon izbacivanja instanci još preostaje jedan manji dio nedostajućih vrijednosti u skupu. Kako bi se potpuno eliminirale nedostajuće vrijednosti koristit ćemo nadomještanje sa normalnom vrijednošću (u medicinskom smislu). Primjerice, za pacijente koji nemaju izmjeren krvni tlak pretpostavlja se normalna vrijednost krvnog tlaka.

GeNIe - [cleaned.csv]

File Edit View Data Tools Window Help

Tree View

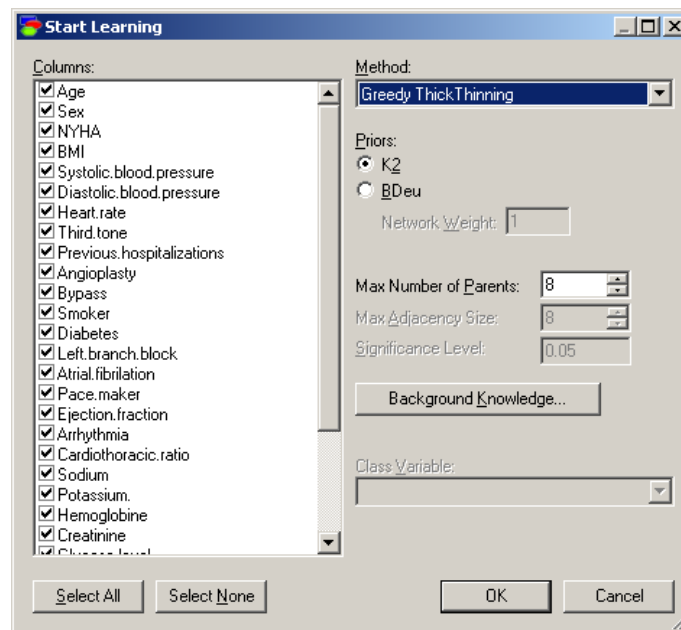
	Age	Sex	NYHA	BMI	Systolic blood pressure	Diastolic blood pressure	Heart rate	Third tone	Previous ho...
▶	Old	M	2	High	High	Normal	Normal	no	None
	Young	M	1	High	Normal	Normal	Normal	no	None
	Old	F	2	Normal	Normal	Normal	Low	yes	OneToThree
	Young	F	3	High	Normal	Normal	Normal	no	OneToThree
	Old	M	3	Normal	Normal	Low	Low	no	None
	VeryOld	M	2	High	Normal	Normal	Low	no	None
	Young	M	2	High	Normal	Normal	Normal	no	OneToThree
	Young	M	2	High	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	1	Normal	Normal	Normal	Normal	no	OneToThree
	Young	M	1	High	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	2	High	Normal	Normal	Normal	no	OneToThree
	VeryOld	F	3	Normal	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	2	Low	Normal	Normal	Normal	yes	OneToThree
	Old	M	2	Normal	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	1	Normal	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	2	Normal	Normal	Normal	Normal	no	OneToThree
	Old	M	3	High	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	2	Normal	Normal	Normal	Normal	yes	OneToThree
	Old	F	2	Normal	Normal	Normal	Normal	no	OneToThree
	VeryOld	M	2	High	High	Normal	Normal	no	None
	Old	M	1	Normal	Normal	Normal	Normal	no	None
	VeryOld	M	3	High	Normal	Normal	Normal	yes	OneToThree
	VeryOld	M	3	Normal	Normal	Normal	Normal	no	None
	Old	M	3	Low	Normal	Normal	Normal	no	OneToThree

Row 1 of 6171

Slika 4.13 Skup podataka pripremljen za učenje Bayesove mreže.

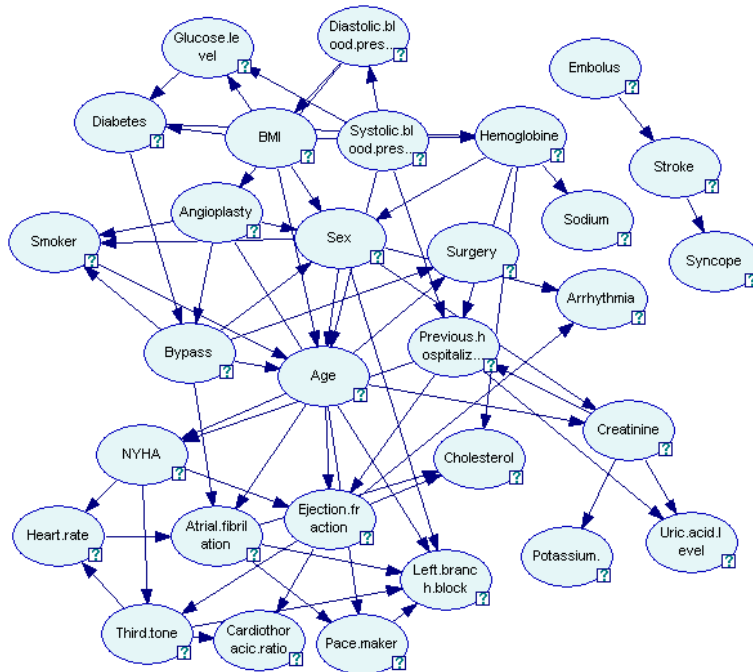
S obzirom da se Bayesova mreža može učiti samo iz diskretnih atributa potrebno je provesti i diskretizaciju podatkovnih vrijednosti u skupu podataka. Diskretizaciju je moguće provesti na način da se svaki atribut podijeli na intervale tako da svaki interval dobije razmjernan dio instanci pacijentata. Međutim, mnogo je razumnije granice postaviti na nivoe koje imaju medicinskog smisla. Primjerice, numerička vrijednost BMI (engl. *body mass index*) se diskretizira prema granicama 18 i 25 na vrijednosti niske, normalne i visoke razine. Kada su podaci na ovaj način pripremljeni moguće je učitati ih u GeNIe programski paket. Slika 4.13. prikazuje učitani pripremljeni skup podataka.

Na koncu preostaje još i izbor metode učenja Bayesove mreže. U GeNIe paketu nude se četiri implementirane metode (engl. *Greedy Thick Thinning, PC, Naive, Naive with feature selection*) gdje je potrebno za svaku metodu još dodatno specificirati parametre pojedine metode učenja (Slika 4.14).

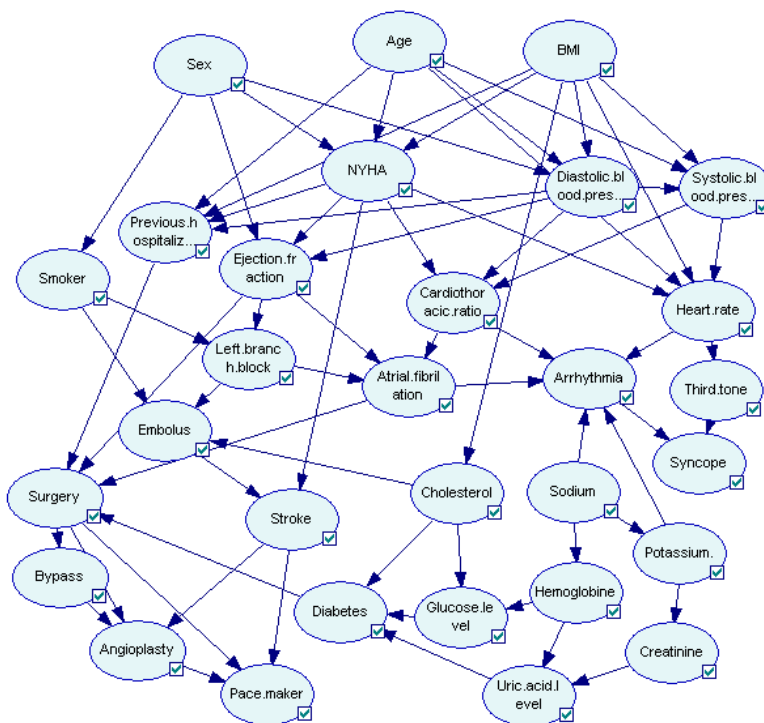


Slika 4.14. Odabir parametara pri učenju Bayesove mreže iz skupa podataka.

Na slici 4.15 prikazan je primjer mreže koji se dobije procesom učenja. Najveća poteškoća kod mreže dobivene strojnim učenjem je neintuitivnost strukture mreže te teška ili nemoguća medicinska interpretacija samog rezultata. Stoga je mnogo prihvatljivije dopustiti medicinskim ekspertima konstrukciju strukture Bayesove mreže (ili barem popravak strukture dobivene strojnim učenjem). Nakon toga se tablice uvjetnih vjerojatnosti mogu naučiti odvojeno procesom učenja iz skupa podataka. Slika 4.16 prikazuje strukturu Bayesove mreže koja je izgrađena ručno kako bi se postigla jasnija medicinska interpretacija a kojoj su postupkom strojnog učenja pridodane tablice uvjetnih vjerojatnosti u čvorovima.



Slika 4.15. Struktura Bayesove mreže dobivena procesom učenja često je neintuitivna i ne daje mogućnost medicinske interpretacije.



Slika 4.16. Često je prihvatljivije prepustiti medicinskim ekspertima izgradnju strukture (ili barem dijela) mreže a vjerojatnosne zavisnosti naučiti iz sakupljenog skupa podataka.

4.4 Povezivanje formalizama

U ovom odjeljku se opisuje konkretan postupak povezivanja dvaju opisanih metodologija predstavljanja znanja u jednu funkcionalnu cjelinu.

4.4.1 Ostvarivanje semantičke veze

Prvi korak u integraciji jest *identifikacija semantički identičnih koncepata* korištenih u ontologiji i Bayesovoj mreži. Primjerice, nužno je uočiti koncept visokog sistoličkog krvnog tlaka i u ontologiji i u Bayesovoj mreži te poveznicom naznačiti da se radi o identičnom konceptu.

Kako bi se ovo postiglo logički izrazi u ontologiji (deskriptivnoj logici) se mapiraju na ishode čvorova u Bayesovoj mreži. Primjeri mapiranja su prikazani u Tablici 4.4.

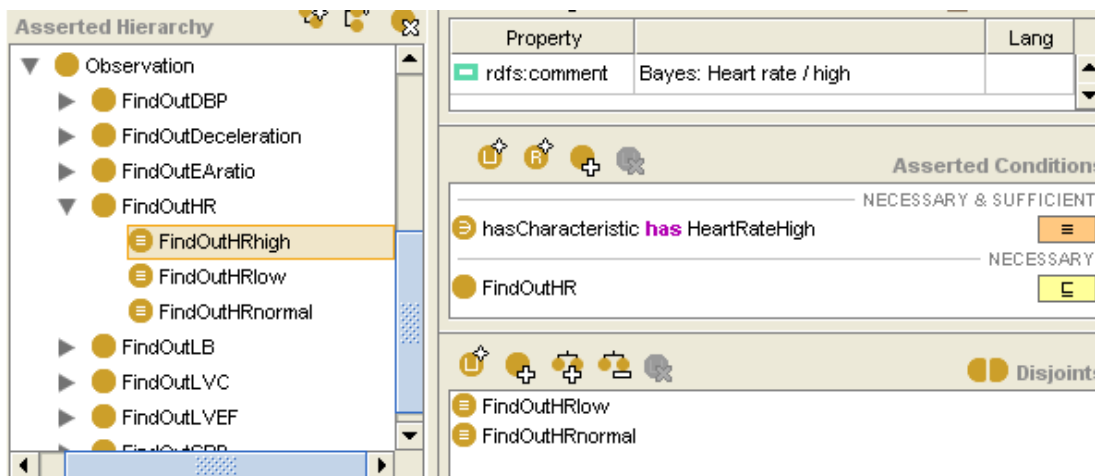
Tablica 4.4. Primjer identifikacije semantički identičnih koncepata u ontologiji i Bayesovoj mreži.

Ontologija (definicija koncepta)	Bayesova mreža (Čvor / Ishod)
\exists HASCHARACTERISTIC.EAratioHigh	EA RATIO / HIGH
\exists HASCHARACTERISTIC.EAratioLow	EA RATIO / LOW
\exists HASCHARACTERISTIC.EAratioNormal	EA RATIO / NORMAL
...	...

Teoretski gledano, bilo koji logički izraz se može mapirati na bilo koji par čvor/ishod u Bayesovoj mreži. Međutim, važno je biti svjestan da su ishodi u čvorovima Bayesove mreže nužno disjunktne (implicitna pretpostavka Bayesove mreže). Nadalje, ako su čvorovi unutar ontologije na neki način semantički povezani, ta veza neće biti održana unutar Bayesove mreže. Mapiranje samo pronalazi semantički identične koncepte u ontologiji i Bayesovoj mreži.

Slika 4.17 prikazuje implementirani način na koji se ontologija povezuje na Bayesovu mrežu u Protégé-OWL alatu. Na slici se vidi da je koncept FINDOUTHRHIGH povezan sa ishodom čvora u Bayesovoj mreži tako da mu je u opis dodan niz znakova „Bayes: Heart rate / high“. Pri tome je „Heart rate“ ime čvora u mreži a „high“ ishod u čvoru Bayesove mreže.

Nadalje, na slici se vidi i definicija koncepta FINDOUTHRHIGH pomoću logičkog izraza. Pri dnu slike vidi se i lista koncepata s kojim je navedeni koncept disjunktan.

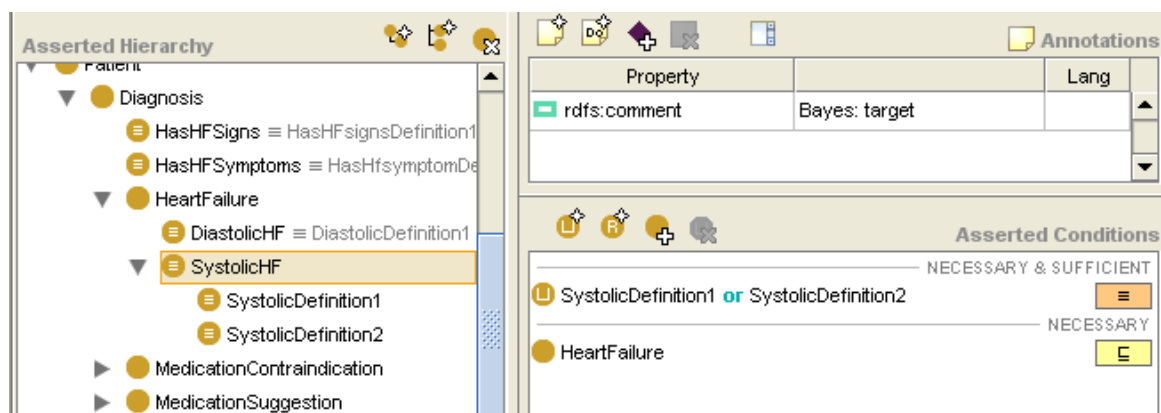


Slika 4.17. Ostvarivanje semantičkih poveznica između ontologije i Bayesove mreže u Protégé alatu.

Opisano prepoznavanje semantički ekvivalentnih koncepata ima potencijala da se obavi (polu) automatski; u ovoj disertaciji se ograničavamo samo na ručno obavljanje opisanog mapiranja.

4.4.2 Integracija logičkih izraza u Bayesovu mrežu

Ontologija sadrži neke koncepte koji su od posebne važnosti u sustavu za potporu pri odlučivanju. Takve koncepte možemo nazvati *ciljni koncepti*. U primjeru dijagnoze srčanog zatajenja može se u ontologiji naznačiti da su ciljni koncepti sljedeći: HASHFSIGNS, HASHFSYMPTOMS, DIASTOLICHF, SYSTOLICHF. Dakle, radi se o konceptima koji su ključni za dijagnozu srčanog zatajenja.



Slika 4.18. Označavanje ciljnih pojmova u ontologiji u Protégé alatu.

Na Slici 4.18 prikazan je način obilježavanja ciljnih koncepata u ontologiji u Protégé alatu. U opisu koncepta SYSTOLICHF je unešen niz „Bayes: target“. Isto je učinjeno i za preostala 3 ciljna koncepta.

Kada su označeni ciljni koncepti u ontologiji oni se mogu automatski preslikati u Bayesovu mrežu. Dakle, drugi korak integracije jest *preslikavanje definicija ciljnih koncepata* u Bayesovu mrežu.

Proces preslikavanja započinje *ekspanzijom koncepta* [3]. Ekspanzija koncepta se obavlja na način da se koncepti koji su korišteni u definiciji ciljnog koncepta iterativno zamjenjuju svojim definicijama sve dok je to moguće. Kao primjer može se pogledati definicija koncepta DIASTOLICHF:

$$\begin{aligned} \text{DIASTOLICHF} &\equiv \\ &(\text{HASHFSIGN} \sqcup \text{HASHFSYMPATOM}) \\ &\sqcap \exists \text{HASCHARACTERISTIC.EARATIOHIGH} \\ &\sqcap \exists \text{HASCHARACTERISTIC.DBLOW} \end{aligned}$$

U početku ekspanzije se koncepti HASHFSIGN i HASHFSYMPATOM zamjenjuju svojim definicijama pa slijedi:

$$\begin{aligned} \text{DIASTOLICHF} &\equiv \\ &(\exists \text{HASCHARACTERISTIC.HFSIGN} \sqcup \exists \text{HASCHARACTERISTIC.HFSYMPATOM}) \\ &\sqcap \exists \text{HASCHARACTERISTIC.EARATIOHIGH} \\ &\sqcap \exists \text{HASCHARACTERISTIC.DBLOW} \end{aligned}$$

U ovom trenutku načini se pretpostavka da ontologija sadži kompletan opis domene. Drugim riječima, moguće je pretpostaviti da koncept HFSIGN sadrži listu svih mogućih znakova srčanog zatajenja (analogno za koncept HFSYMPATOM). U općem slučaju ovo nije nužno istinita pretpostavka (jer ontologija slijedi pretpostavku otvorenog svijeta), ali u ovom konkretnom slučaju pretpostavka djeluje razumno.

Uz uvedenu pretpostavku moguće je koncepte korištene u definiciji ciljnog koncepta zamijeniti listom pripadajućih instanci. Tako se koncepti HFSIGN i HFSYMPATOM proširuju na pripadajuće liste instanci.

Kada je proces ekspanzije završen moguće je integrirati dobiveni prošireni koncept u Bayesovu mrežu. Ovaj je proces implementiran na način da se u proširenim logičkim izrazima obave sva moguća preslikavanja iz Tablice 4.4 te se na taj način dijelovi logičkog izraza zamjene adekvatnim čvorovima i ishodima iz Bayesove mreže. Tako se prošireni logički izraz za dijastoličko srčano zatajenje transformira u sljedeći logički izraz koji je sagrađen od čvorova i ishoda Bayesove mreže:

```

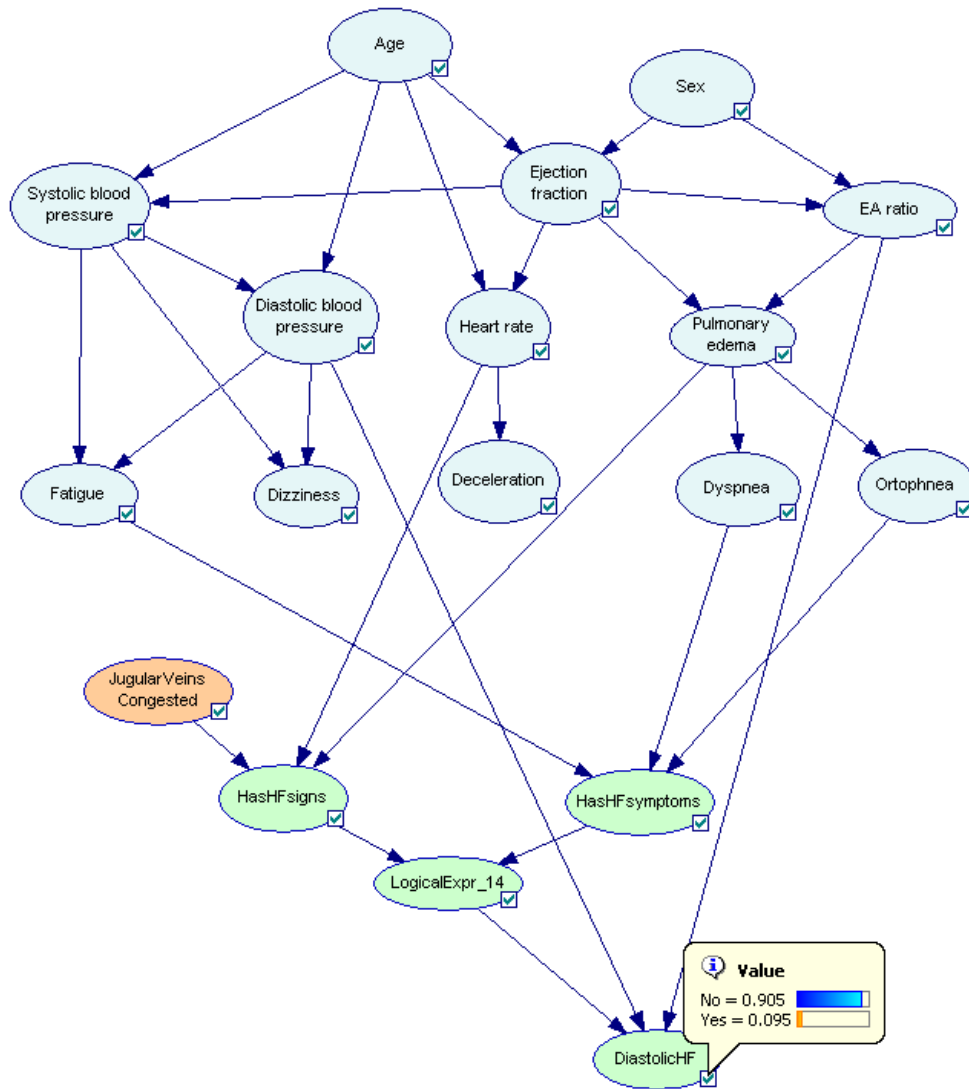
(DIASTOLIC HF / YES) ≡
(
  (
    (HEART RATE / HIGH)
    OR (PULMONARY EDEMA / PRESENT)
    OR (JUGULAR VEINS CONGESTED / YES)
  )
  OR
  (
    (FATIGUE / PRESENT)
    OR (DYSPNEA / PRESENT)
    OR (ORTOPHNEA / PRESENT)
  )
)
AND (EA RATIO / HIGH)
AND (DIASTOLIC BLOOD PRESSURE / LOW)

```

Na temelju ovog logičkog izraza se automatski u Bayesovu mrežu dodaje novi čvor koji se nazove DIASTOLIC HF. Pri tome se po potrebi u mreži stvaraju novi među-čvorovi koji predstavljaju korištene logičke među-izraze. Slika 4.19 prikazuje način na koji je koncept DIASTOLIC HF integriran u Bayesovu mrežu.

Plavi čvorovi na Slici 4.19 predstavljaju mrežu koja je prethodno zadana ručno. Ovaj dio mreže je ostao nepromijenjen.

Zeleni čvorovi u mreži su nastali transformacijom logičkih izraza iz ontologije. Oni su direktno ili indirektno povezani sa čvorovima o kojima ovisi njihova istinsna vrijednost. Tako je čvor DIASTOLIC HF povezan sa čvorovima DIASTOLIC BLOOD PRESSURE, EA RATIO i LOGICALEXP_14. Tablica uvjetnih vjerojatnosti nekog zelenog čvora gradi se računanjem tablice istinitosti zadanog logičkog izraza. U Tablici 4.5 prikazana je definicija čvora DIASTOLIC HF ovisno o njegovim roditeljima.



Slika 4.19. Ugrađivanje pojma dijastoličkog zatajenja srca koristeći definiciju pojma zadanu logičkim izrazima u ontologiji.

Tablica 4.5. Definicija uvjetnih vjerojatnosti u čvoru DIASTOLICHF.

EA RATIO		NORMAL						HIGH					
		DIASTOLIC BP		NORMAL		LOW		HIGH		NORMAL		LOW	
SIGNS OR SYMPTOMS		NO	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES
DIASTOLICHF	NO	1	1	1	1	1	1	1	1	1	1	1	0
	YES	0	0	0	0	0	0	0	0	0	0	0	1

Narančasti čvor na Slici 4.19 (JUGULARVEINSCONGESTED) predstavlja pojam koji se koristi u logičkom izrazu u ontologiji, a za kojeg ne postoji odgovarajući čvor u mreži. Naime, za očekivati je da će postojati pojmovi u ontologiji koji nisu uzeti u obzir pri konstrukciji Bayesove mreže. Za takve slučajeve se pri preslikavanju definicije u Bayesovu mrežu dodaje novi čvor za kojeg se definira njegovo a priori očekivanje. Ta se vrijednost pri konstrukciji ontologije definira pokraj definicije pojma (npr. kraj instance *jugularVeinsCongested* u ontologiji stoji da se pojavljuje u ukupno 10% populacije pacijenata). Na taj način se pojmovi za koje je poznata međusobna ovisnost povezuju međusobnim vezama u Bayesovoj mreži, dok se za pojmove za koje nije jasna relacija sa ostalim pojmovima samo definira njihova a priori vjerojatnost.

4.4.3 Pretpostavke pri integraciji

U opisanom procesu integracije ontologije i Bayesove mreže uvedene su neke pretpostavke koje omogućuju uspješno povezivanje formalizama. Osnovna poteškoća je što deskriptivna logika ima značajno drugačiju semantiku od Bayesovih mreža te nije moguće bez gubitaka integrirati ciljne pojmove iz ontologije u Bayesovu mrežu.

Prva pretpostavka koje je uvedena pri integraciji formalizama jest da ontologija sveobuhvatno definira domenu koju opisuje. Ako ontologija uistinu definira kompletno znanje o domeni koje postoji, tada je moguće neke koncepte u ontologiji zamijeniti listama pripadajućih instanci tog koncepta. Primjerice, na ovaj se način koncept HFSIGN pri ekspanziji zamjenjuje listom instanci znakova srčanog zatajenja. U općem slučaju kod deskriptivne logike zbog pretpostavke otvorenog svijeta nije moguće uvesti ovu zamjenu, ali u ovom konkretnom slučaju primjene (integracija formalizama) opisana zamjena djeluje kao prihvatljiv i razuman korak.

Sljedeća pretpostavka koja je uvedena jest da su ciljni koncepti eksplicitno definirani logičkim izrazima u ontologiji. Ova se pretpostavka može percipirati i kao smjernica za dizajn baze znanja: osoba koja modelira ontologiju bi trebala ciljne koncepte eksplicitno definirati. U općem slučaju kod ontologija to nije nužno slučaj: koncepte je u deskriptivnoj logici moguće definirati i *implicitno* zadanom strukturom klasa, podklasa i/ili logičkom definicijom ostalih koncepata u ontologiji. Ova je pretpostavka potrebna kako bi se ciljni logički izrazi mogli integrirati u mrežu (procesom ekspanzije). Definicije koncepata koje su zadane implicitno (strukturom ontologije) stoga neće biti ispravno obuhvaćene dobivenom Bayesovom mrežom.

Bayesova je mreža na neki način vjerojatnosno proširenje propozicijske logike. Svaki čvor u mreži sadrži tablicu uvjetnih vjerojatnosti koji definira očekivanje promatranog čvora. Analogno tome, u propozicijskoj logici svaki logički izraz može se predstaviti tablicom istinitosti koji definira njegovu istinosnu vrijednost. Razlika između tablica istinitosti u propozicijskoj logici i tablica distribucija vjerojatnosti Bayesovim mrežama

je ta što su kod propozicijske logike dozvoljene samo vrijednosti logičke istine i neistine (tj. vrijednosti 0 i 1) dok je kod tablica združenih vjerojatnosti moguće koristiti i vrijednosti između 0 i 1 (realni brojevi). Međutim, deskriptivna logika podržava i jezične konstrukte koji se ne mogu adekvatno predstaviti nikakvim konstruktima u propozicijskoj logici (primjerice, \exists i \forall). Stoga se pri integraciji uvodi pretpostavka (ograničenje) da se svaki izraz u sintaksi deskriptivne logike može adekvatno preslikati u jedan ishod čvora Bayesove mreže (kao što su ostvarene poveznice u Tablici 4.4)

4.5 Rasuđivanje

U ovom odjeljku opisuje se postupak rasuđivanja u sustavu u kojem je integrirana ontologija sa Bayesovom mrežom. Nadalje, definira se pojam informativnosti čimbenika te se predstavlja postupak računanja informativnosti kod grupiranih pretraga. Postupak rasuđivanja je u jednom dijelu opisan i u [43].

4.5.1 Otkrivanje očekivanja ishoda

Kada je koncept u potpunosti integriran u Bayesovu mrežu moguće je jednim od standardnih postupaka rasuđivanja u Bayesovoj mreži izračunati koja su očekivanja njegovih ishoda.

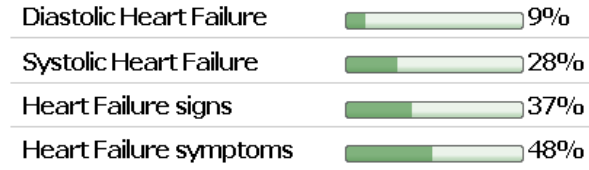
Primjerice, kada je na ovaj način u mrežu integriran koncept DIASTOLICHF moguće je izračunati vrijednosti očekivanja za njegove ishode (YES i NO):

Čvor DIASTOLICHF

$$P(\text{NO}) = 0.905 \quad P(\text{YES}) = 0.095$$

Ove brojke kazuju da je a priori očekivanje da pacijent o kojemu se ne zna niti jedan podatak ima dijastoličko zatajenje srca jednako 10%. Drugim riječima, kada se pojavi novi pacijent kod liječnika, liječnik smatra da taj pacijent najvjerojatnije nema dijastoličko zatajenje srca (90%). Kada bi liječnik saznao još pokoji podatak o tom pacijentu, primjerice da je pacijent muškarac u starijoj dobi, unošenjem tih novih saznanja u sustav mogao bi saznati da je očekivanje dijastoličkog srčanog zatajenja za tog pacijenta poraslo na iznos od 14%.

Kada su koncepti iz ontologije integrirani u mrežu moguće je izračunati i njihova očekivanja. Na taj način mogu se očekivanja ciljnih koncepata iz ontologije i međusobno uspoređivati. Slika 4.20 prikazuje dobivena očekivanja svih ciljnih koncepata u primjeru dijagnoze srčanog zatajenja.



4.20 Očekivanja ciljnih koncepata u primjeru dijagnoze srčanog zatajenja.

4.5.2 Otkrivanje informativnosti čimbenika

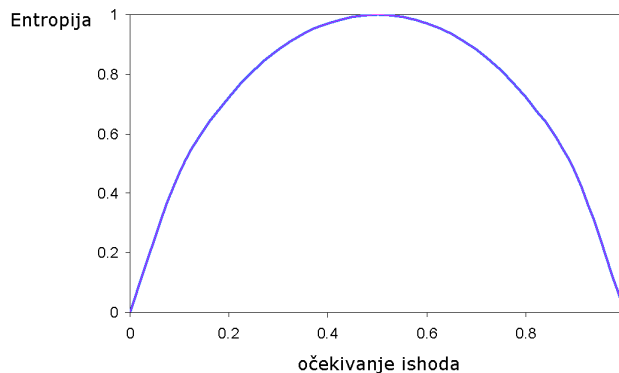
Osim što je zanimljivo saznati koliko je očekivanje pojedinog ciljnog koncepta, veoma je korisno saznati koja informacija u bazi znanja vodi ka najvažnijim odlukama najefikasnije. Drugim riječima, potrebno je izračunati *informativnu dobit* svakog pojedinog dijagnostičkog čvora za svaki pojedini ciljni koncept. Postoji mnogo različitih mjera koje bi bile prikladne u ovoj situaciji; *očekivano smanjenje entropije* je mjera koja dobro pristaje prirodi problema.

Kako bi se izračunala entropija ciljnog čvora koriste se očekivanja svih ishoda ciljnog čvora:

$$Entropy(X) = E(X) = \sum_o -p_o \log_2 p_o,$$

gdje je X ciljni čvor a o ishod ciljnog čvora.

Iznos entropije je najveći (i iznosi 1) kada je informacija o pojedinom čvoru potpuno nejasna ($P(NO) = P(YES) = 0.5$). Kako se iznos očekivanja približava krajevima (0 ili 1) tako se entropija približava nuli. Slika 4.21 prikazuje ovisnost entropije o očekivanju ishoda ciljnog čvora.



Slika 4.21. Ovisnost iznosa entropije o očekivanju ishoda ciljnog čvora.

Primjerice, vrijednost entropije ciljnog pojma mjerenja krvnog tlaka računa se iz očekivanja svih vrijednosti pojma (čvora):

Čvor DIASTOLIC HF

$$P(\text{NO}) = 0.905 \quad P(\text{YES}) = 0.095$$

$$Entropy(\text{DIASTOLIC HF}) = -0.905 * \log_2 0.905 - 0.095 * \log_2 0.095 = 0.4529$$

Poželjno je iznos entropije što više približiti nuli, jer to znači da je odgovor na ciljno pitanje izvjesniji. Obavljanjem neke pretrage iznos entropije će se promijeniti ovisno o ishodu obavljene pretrage. Mjera koja uzima sve moguće ishode entropije sa pripadnim težinama naziva se *očekivana entropija*.

Pogleda li se pretraga SYSTOLIC BLOOD PRESSURE trenutno je jasno da su ishodi pretrage izvjesni sa sljedećim očekivanjima (a priori vrijednosti):

Čvor SYSTOLIC BLOOD PRESSURE

$$P(\text{HIGH}) = 0.296 \quad P(\text{NORMAL}) = 0.372 \quad P(\text{LOW}) = 0.332$$

Obavljanjem pretrage mjerenja sistoličkog tlaka moguća su tri ishoda. Svaki od ishoda rezultira različitom entropijom ciljnog izraza kako je prikazano u Tablici 4.6. Tablica analizira sva tri moguća ishoda mjerenja. Prvi ishod mjerenja jest da pacijent ima povišen krvni tlak (što se događa u 29.6% slučajeva), pri čemu se očekivanje da ima dijastoličko srčano zatajenje mijenja (spušta se na 1.4%), što mijenja i entropiju ciljnog izraza (na 0.0874). Sličan scenarij se ponavlja za sve tri mogućnosti.

Tablica 4.6. Ovisnost entropije ciljnog izraza o ishodu obavljene jedne pretrage.

Ishod mjerenja SBP		HIGH (0.296)	NORMAL (0.372)	LOW (0.332)
Nova očekivanja u čvoru DIASTOLIC HF	NO	0.989	0.978	0.748
	YES	0.011	0.022	0.252
Nova entropija		0.0874	0.1525	0.8144

Sumarna mjera koja obuhvaća sve moguće ishode pojedinog testa u obzir naziva se *očekivanje entropije* i računa se na sljedeći način:

$$ExpectedEntropy(X, D) = \sum_d p_d E(X | D = d),$$

gdje je X ciljni čvor, D promatrani dijagnostički čvor, d je ishod čvora D , p_d očekivanje ishoda d , $E(X | D=d)$ je entropija ciljnog čvora X kada se ishod d dogodio.

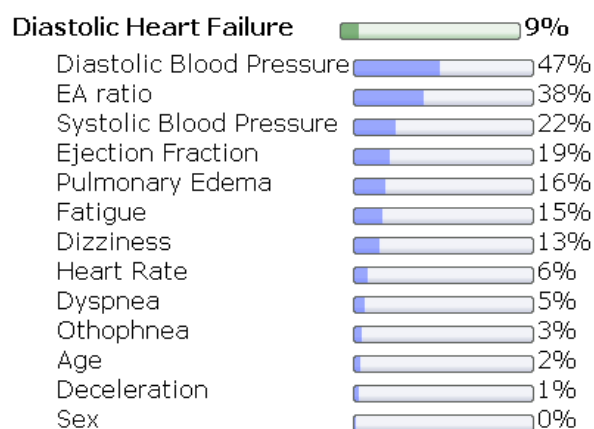
Na ovaj način može se izračunati očekivanje entropije nakon obavljanja pretrage mjerenja sistoličkog tlaka pacijenta:

$$\begin{aligned}
 \text{ExpectedEntropy}(\text{DIASTOLIC HF, SYSTOLIC BLOOD PRESSURE}) &= \\
 &= 0.296 * 0.0874 + 0.372 * 0.1592 + 0.332 * 0.8144 \\
 &= 0.3530
 \end{aligned}$$

Dakle, mjerenjem sistoličkog tlaka pacijenta se očekivanje entropije za dijastoličko zatajenje srca smanjuje sa 0.4592 na 0.3530. Razlika između ove dvije vrijednosti naziva se *informacijska dobit* (engl. *information gain*). U postotcima gledano, entropija se smanjuje za 22% početnog iznosa. Ovaj postotak naziva se *informativnost* pretrage s obzirom na ciljni pojam za nekog konkretnog pacijenta.

Može se dokazati kako se niti jednom pretragom u Bayesovoj mreži ne može povećati iznos entropije logičkog izraza [24]. Stoga je informativnost svake moguće pretrage pozitivna.

Sada se može isti postupak računanja informativnosti ponoviti i za ostale pretrage u mreži. Kada se te pretrage poslažu po dobivenim iznosima informativnosti može se nacrtati graf kakav je prikazan na Slici 4.22.



Slika 4.22. Informativnosti elementarnih pretraga pri dijagnozi dijstoličkog srčanog zatajenja.

4.5.3 Informacijska dobit grupiranih pretraga

Poteškoća kod grupiranih pretraga jest što broj različitih mogućih ishoda raste sa brojem pretraga u grupi. Primjerice, mjeri li se istovremeno sistolički i dijastolički tlak od kojih svaki poprima jednu od tri moguće vrijednosti (HIGH, NORMAL, LOW), moguće je ukupno devet različitih ishoda grupirane pretrage, kao što je prikazano u Tablici 4.7.

Ukupno gledajući, broj mogućih ishoda jednak je umnošku broja ishoda svih elementarnih pretraga u grupi. Primjerice, sastoji li se grupa od deset elementarnih pretraga od kojih svaka ima dva moguća ishoda, ukupan broj mogućih ishoda grupe je 2^{10}

= 1024. Ako elementarne pretrage imaju svaka po tri moguća ishoda, ukupan broj mogućih ishoda grupe je $3^{10} = 59,049$. Broj ishoda u grupi raste sa veličinom grupe i brojevima ishoda elementarnih pretraga u grupi.

Tablica 4.7. Mogući ishodi grupirane pretrage mjerenja krvnog tlaka.

Ishod	1	2	3	4	5	6	7	8	9
DBP	LOW	LOW	LOW	NORMAL	NORMAL	NORMAL	HIGH	HIGH	HIGH
SBP	LOW	NORMAL	HIGH	LOW	NORMAL	HIGH	LOW	NORMAL	HIGH

Informativnost grupirane pretrage

Formalno gledano, da bi se izračunala informativnost grupe G (s obzirom na neki zadani ciljni koncept X) potrebno je za svaki mogući ishod g grupe G izračunati:

- vrijednost očekivanja događanja ishoda (p_g) i
- entropiju ciljnog izraza ako se ishod g dogodi $E_g = E(X | G = g)$.

Tada je očekivana entropija ciljnog izraza nakon obavljanja grupirane pretrage jednaka:

$$ExpectedEntropy(X, G) = \sum_g p_g E_g,$$

gdje je X promatrani ciljni koncept a G promatrana grupna pretraga.

Kako različiti ishodi mogu imati različite a priori vjerojatnosti događaja, a i zbog koreliranosti elementarnih pretraga u grupi, neki se ishodi događaju jako često dok se neki u pravilu uopće ni ne događaju. Primjerice, kako su sistolički i dijastolički tlak izrazito korelirane veličine, nerealno je očekivati (ili se izrazito rijetko događa) da će neki pacijent imati istovremeno visok dijastolički tlak i nizak sistolički tlak (ishod 7). Taj ishod zbog toga ima jako nisku razinu očekivanja (p_7 je jako mali) te stoga ne utječe značajno na konačni iznos entropije ciljnog izraza.

Nadalje, zanimljiva situacija se može dogoditi ako neko elementarno mjerenje u grupi izravno utječe konačni ishod ciljnog izraza (kao npr. dijastolički tlak na dijastoličko zatajenje srca). Primjerice, ako pacijent nema nizak dijastolički tlak (svi ishodi od 4 do 9 u Tablici 4.7 i Tablici 4.8) očekivanje ciljnog izraza jednaka je nuli, te je zbog toga i entropija ciljnog izraza jednaka nuli ($E_g = 0$). Stoga ishodi 4-9 uopće ne doprinose ukupnom iznosu entropije ciljnog izraza.

Dakle, iako teoretski ima mnogo ishoda koji se mogu dogoditi, u praksi je broj ishoda koji značajno doprinosi entropiji ciljnog izraza znatno manji.

Tablica 4.8 prikazuje postupak računanja očekivane entropije (0.231) ciljnog izraza (dijastoličko srčano zatajenje) nakon izmjerene grupnog mjerenja krvnog tlaka. Kako je entropija ciljnog izraza prije obavljanja mjerenja krvnog tlaka jednaka 0.453, mjerenjem krvnog tlaka se entropija ciljnog izraza smanjuje za ukupno 49%. Ovaj postotak predstavlja iznos *informativnosti grupirane pretrage* mjerenja krvnog tlaka s obzirom na izraz za dijastoličko zatajenje srca.

Tablica 4.8. Računanje očekivane entropije ciljnog izraza s obzirom na grupiranu pretragu.

Ishod grupirane pretrage (g)	Očekivanje događanja ishoda (p_g)	Očekivanje istinitosti ciljnog izraza	Entropija ciljnog izraza (E_g)	Doprinos očekivanoj entropiji ($p_g * E_g$)
1	0.193	0.429	0.985	0.191
2	0.038	0.214	0.750	0.028
3	0.018	0.171	0.661	0.012
4	0.106	0	0	0
5	0.262	0	0	0
6	0.061	0	0	0
7	0.033	0	0	0
8	0.074	0	0	0
9	0.215	0	0	0

$$\Sigma = 1.0$$

$$\Sigma = 0.231$$

Pogleda li se ponovno Slika 4.22 vidi se je informativnost zasebno obavljenih elementarnih pretraga mjerenja sistoličkog i dijastoličkog krvnog tlaka sljedeća:

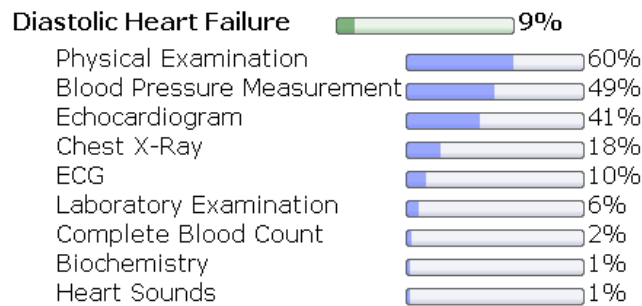
Dijastolički krvni tlak: 47%

Sistolički krvni tlak: 22%

Iz navedenog se može zaključiti da je najveći dio informativnosti sistoličkog krvnog tlaka prisutan upravo zahvaljujući vezi sa dijastoličkim krvnim tlakom. Nadalje, važno je primjetiti da informativnost grupiranog mjerenja nikada *ne može* biti manja od informativnosti bilo koje od elementarnih pretraga u grupi.

Na isti način mogu se izračunati informativnosti svih ostalih pretraga u ontologiji. Slika 4.19 prikazuje informativnosti definiranih grupiranih pretraga za ciljni pojam dijastoličko

srčano zatajenje. Dalje je moguće i raščlanjivati pojedine grupirane pretrage po strukturi kako je definirano u ontologiji.



Slika 4.23. Informativnosti grupiranih pretraga pri dijagnozi dijastoličkog srčanog zatajenja.

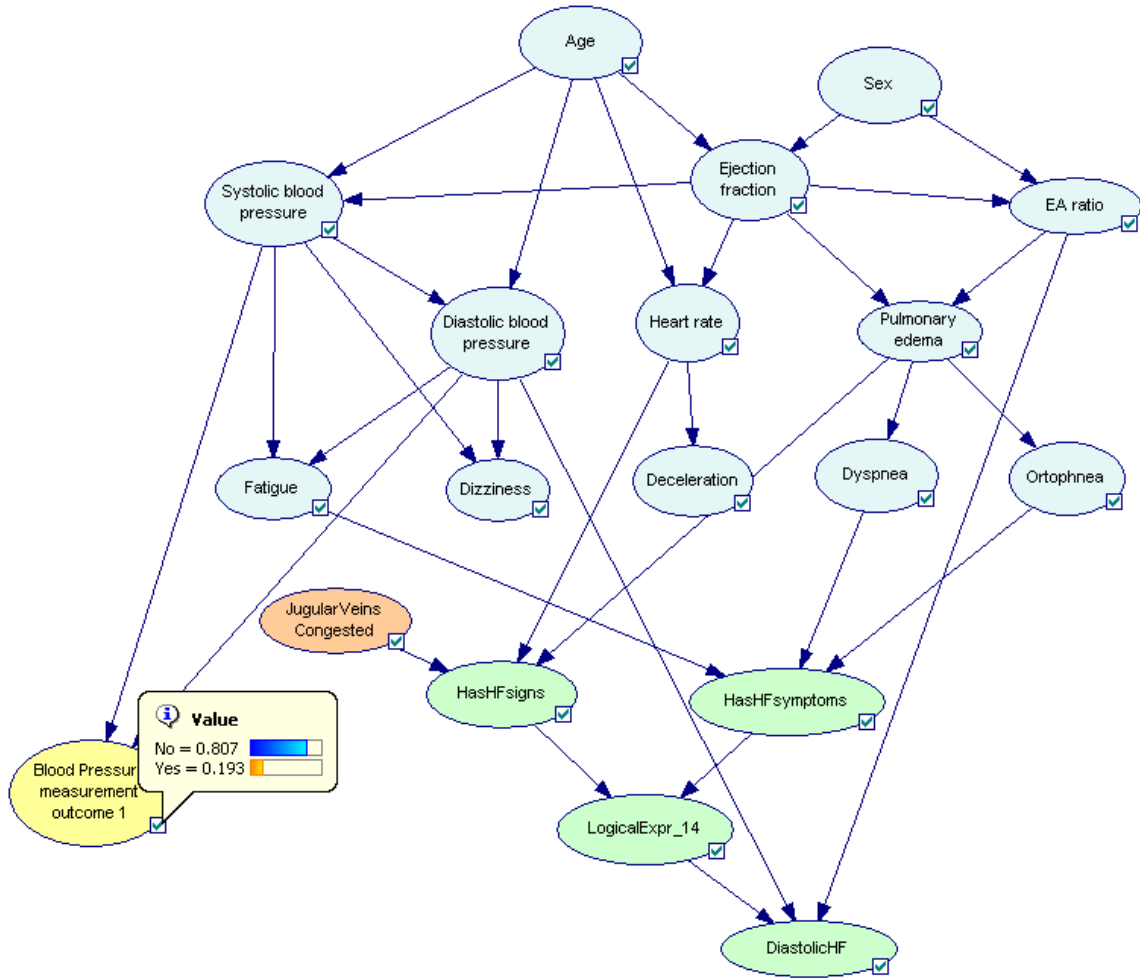
4.5.4 Otkrivanje informacijske dobiti egzaktnim postupcima

Postupku računanja informacijske dobiti opisanom u prethodnom odjeljku još nedostaje metodologija izračuna vjerojatnosti ishoda p_g i računanja a posteriori vjerojatnosti ciljnog koncepta kada se promatrani ishod g dogodi (a ujedno time i a posteriori entropiju E_g).

Valja naglasiti da iz Bayesove mreže *nije* moguće direktno iščitati ove dvije vrijednosti. Primjerice, iz Bayesove je mreže moguće očitati pojedinačne vjerojatnosti niskog sistoličkog krvnog tlaka kao i niskog dijastoličkog krvnog tlaka ali nije moguće direktno očitati koja je vjerojatnost istovremenog događanja ovih ishoda. Također, nije moguće direktno pročitati ni koje je a posteriorno očekivanje ciljnog čvora kada se neki ishod dogodi. Da bi se očitala a posteriorna vjerojatnost potrebno je u mrežu postaviti dokaze kao da se neki promatrani ishod dogodio, pa tek tada osvježavanjem očekivanja u Bayesovoj mreži očitati traženu a posteriornu vjerojatnost ciljnog čvora.

Kako bi se pronašla a priori vjerojatnost promatranog ishoda p_g potrebno je u Bayesovoj mreži konstruirati pomoćni (privremeni) čvor sa dva ishoda (YES i NO) kojemu bi roditelji bili svi čvorovi elementi promatrane grupe. Nadalje, pomoćnom čvoru bi tablica uvjetnih vjerojatnosti bila definirana na način da daje ishod YES samo u stupcu koji odgovara trenutno promatranom ishodu g . Primjerice, pri računanju očekivanja p_I grupirane pretrage mjerenja krvnog tlaka u Bayesovoj mreži je potrebno dodati pomoćni čvor koji će imati roditelje SYSTOLIC BLOOD PRESSURE i DIASTOLIC BLOOD PRESSURE kao na Slici 4.24 a tablicu uvjetnih vjerojatnosti definiranu na način kako je to prikazano u Tablici 4.9.

Kada se obnove vjerojatnosti u mreži tada će vjerojatnost ishoda YES na pomoćnom čvoru biti jednaka upravo iznosu p_g kao što je prikazano na Slici 4.24.



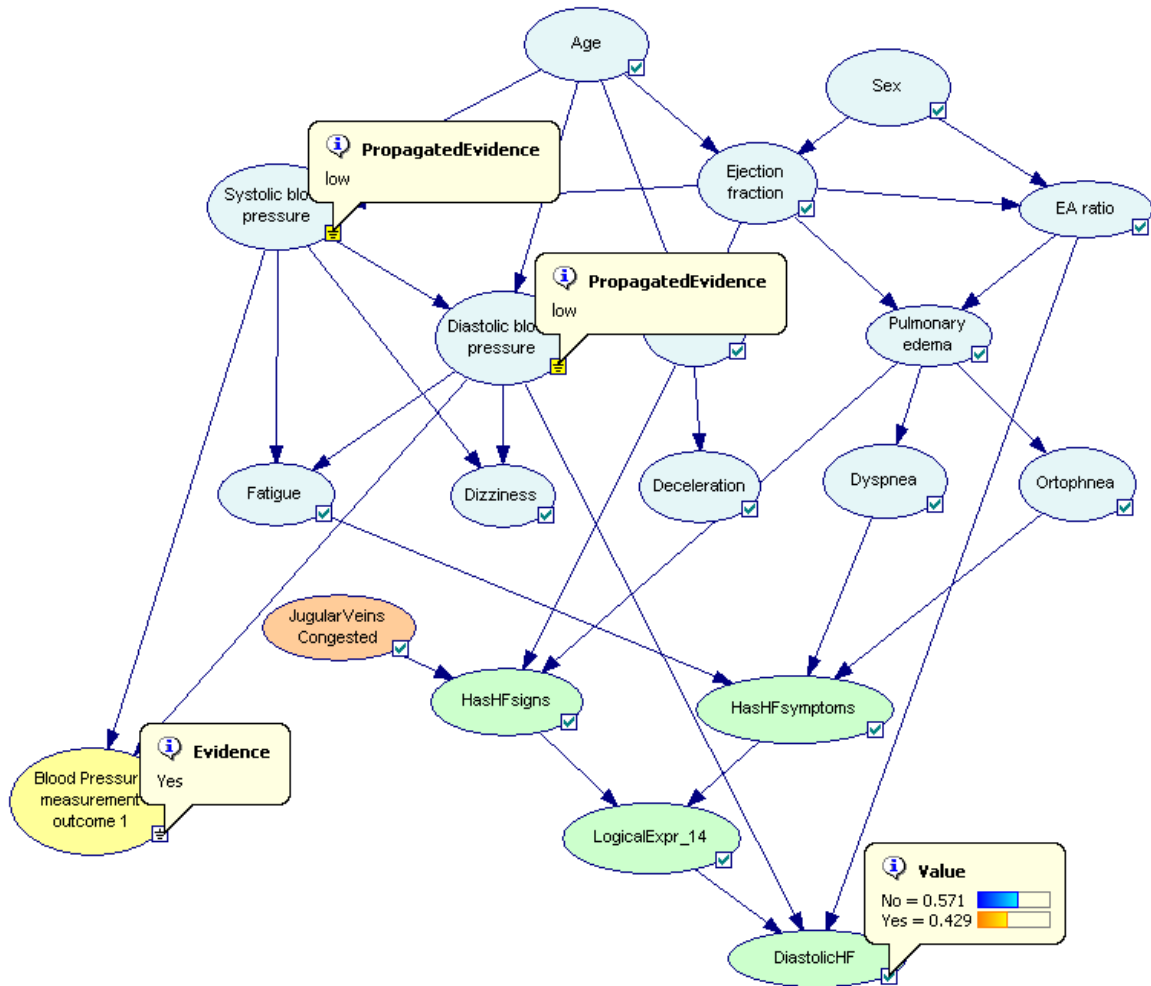
Slika 4.24. Koristeći privremeni čvor otkriva se vjerojatnost jednog ishoda grupirane pretrage.

Tablica 4.9. Definiranje uvjetnih vjerojatnosti u pomoćnom čvoru pri računanju očekivanja ishoda 1 grupirane pretrage mjerenja krvnog tlaka.

DBP	LOW			NORMAL			HIGH		
	LOW	NORMAL	HIGH	LOW	NORMAL	HIGH	LOW	NORMAL	HIGH
NO	0	1	1	1	1	1	1	1	1
YES	1	0	0	0	0	0	0	0	0

Kako bi se pronašao iznos E_g potrebno je izračunati a posteriori vjerojatnost promatranog ciljnog koncepta kada se ishod g dogodi. Kako bi se ta vrijednost izračunala moguće je

postaviti dokaz na ranije konstruirani pomoćni čvor na ishod YES. Naime, kada je vrijednost pomoćnog čvora YES tada se u mreži zbog specifične konstrukcije i definicije pomoćnog čvora nužno propagiraju dokazi na sve čvorove roditelje. Ova je situacija prikazana Slikom 4.25. Nakon obnavljanja vjerojatnosti u mreži moguće je očitati koja je vjerojatnost promatranog ciljnog čvora. Iz očitane vjerojatnosti dalje se jednostavno izračuna i a posteriori entropija ciljnog čvora E_g .



Slika 4.25. Računanje a posteriori očekivanja promatranog ciljnog čvora.

Vidimo da je u jednom koraku potrebno dva puta obaviti obnavljanje vjerojatnosti u Bayesovoj mreži. Opisani postupak računanja informativnosti potrebno je ponoviti:

- za svaki mogući ishod
- svake postojeće grupirane pretrage i
- za svaki postojeći ciljni izraz.

Ako je velik broj ciljnih izraza, grupiranih pretraga i ishoda u grupama postupak postaje izrazito složen. Primjerice, ako neka grupirana pretraga ima 20 elementarnih pretraga (dakle oko milijun ishoda) potrebno je izračunati očekivanja (vjerojatnosti) svakog pojedinog ishoda te dodatno i a posteriori očekivanja promatranog ciljnog čvora (također za svih milijuh slučajeva). Kada bi obnavljanje vjerojatnosti za jedan ovakav slučaj trajalo desetinku sekunde (što je u okvirima trajanja rasuđivanja u Bayesovoj mreži) tada bi ukupan proces trajao:

$$\text{Vrijeme izvršavanja} = 2^{20} * 0.1s \approx 1,000,000 * 0.1s = 100,000 \text{ sekundi} \approx 27 \text{ sati.}$$

Ovoliko bi okvirno trajalo računanje informativnosti za samo jednu grupiranu pretragu i za samo jedan ciljni koncept. Kada je definirano više pretraga i više ciljnih koncepata (što je također sasvim uobičajena situacija) vrijeme rasuđivanja raste linearno sa oba ova parametra.

Iz izloženog je evidentno da valja pokušati pronaći neki drugačiji pristup kako bi se željene vrijednosti mogle izračunati u stvarnom vremenu.

4.5.5 Otkrivanje informacijske dobiti uzorkovanjem

Koristeći Bayesovu mrežu moguće je stvoriti skup uzoraka (umjetni skup pacijenata) u kojemu se vrijednosti u uzorcima pojavljuju na način da podupiru vrijednosti zadane u Bayesovoj mreži. Uzorkovanje (engl. *sampling*) se često koristi za klasično računanje a priori vjerojatnosti i vrijednosti očekivanja u Bayesovoj mreži. Algoritmi za uzorkovanje su nešto detaljnije obrađeni u uvodnom poglavlju (poglavlje 3.2).

Tablica 4.10 prikazuje mali isječak iz skupa uzorka koji je generiran koristeći Bayesovu mrežu sa slike 4.12.

Tablica 4.10. Uzorci generirani na temelju Bayesove mreže i postavljenih dokaza.

Uzorak	AGE	SEX	SBP	DBP	E-A RATIO	LVEF	...
1	OLD	MALE	NORMAL	NORMAL	NORMAL	LOW	...
2	OLD	MALE	HIGH	HIGH	HIGH	NORMAL	...
3	YOUNG	MALE	HIGH	NORMAL	HIGH	NORMAL	...
4	OLD	FEMALE	LOW	LOW	NORMAL	LOW	...
5	OLD	FEMALE	NORMAL	LOW	NORMAL	NORMAL	...
...

Ako je broj generiranih uzoraka dovoljno velik, moguće je sa zadovoljavajućom točnošću izračunati očekivanja u svim čvorovima u mreži ovisno o postavljenim dokazima:

Čvor AGE

$$p(old) = \frac{\text{broj uzoraka gdje je Age = old}}{\text{ukupan broj uzoraka}}$$

$$p(young) = \frac{\text{broj uzoraka gdje je Age = young}}{\text{ukupan broj uzoraka}}$$

Izračunate vrijednosti predstavljaju očekivanja ishoda čvora AGE. Ako su prije uzorkovanja u mreži bili postavljeni neki dokazi, tada oni utječu na proces uzorkovanja pa izračunata vrijednost uistinu predstavlja očekivanje tog ishoda.

Kod računanja informativnosti grupirane pretrage potrebno je izračunati očekivanja svih mogućih ishoda grupirane pretrage p_g . Koristeći (isti) generirani skup uzoraka to se može izračunati na sljedeći način:

$$p_g = \frac{\text{broj uzoraka sa ishodom } g}{\text{ukupan broj uzoraka}}$$

Na taj način izračunavaju se očekivanja p_g u drugom stupcu Tablice 4.8:

ukupan broj uzoraka:	100,000	
broj uzoraka sa ishodom 1:	19,333	$p_1 = 0.193$
broj uzoraka sa ishodom 2:	3,791	$p_2 = 0.038$
broj uzoraka sa ishodom 3:	1,779	$p_3 = 0.018$
...		

Vidi se da se generirani skup uzoraka podijelio (neravnomjerno) u devet disjunktih podskupova. U svakom pojedinom podskupu moguće je prebrojiti uzorke *za koje je istinit ciljni logički izraz*:

ishod 1	uzoraka: 19,333	istinitih: 8,292	očekivanje: 0.429
ishod 2	uzoraka: 3,791	istinitih: 813	očekivanje: 0.214
ishod 3	uzoraka: 1,779	istinitih: 305	očekivanje: 0.171
...			

Na taj način se izračunaju očekivanja istinitosti ciljnog izraza za svaki pojedini ishod kao u trećem stupcu Tablice 4.8. Iz dobivenih se vrijednosti dalje jednostavno izračunaju entropije za svaki pojedini ishod, te na koncu ukupna očekivana entropija ciljnog izraza i informativnost grupirane pretrage kao što je opisano u odjeljku 4.5.3.

Važno je primjetiti da se iz *samo jednog skupa uzoraka* istovremeno računaju vrijednosti informativnosti:

- za svaki mogući ishod,
- za svaku postojeću grupiranu pretragu i
- za svaki postojeći ciljni izraz

i to u *samo jednom* prolasku kroz skup uzoraka.

4.5.6 Podešavanje veličine skupa uzoraka

U praktičnoj primjeni veoma je važno podesiti broj uzoraka na dovoljno visoku vrijednost kako bi se izbjegle veće pogreške u proračunu. Istovremeno, odabirom nepotrebno velikog broja uzoraka značajno se povećava vrijeme izvođenja. Pokazuje se da je najveći utrošak vremena izvođenja pri rasuđivanju u samom postupku generiranja uzoraka, dok je vrijeme izračuna vrijednosti iz dobivenog skupa uzoraka zanemarivo (Slika 6.11).

Kada se informativnost grupe računa na temelju skupa uzoraka može se činiti problematičnim to što se ishodi koji su malo vjerojatni podupiru u jako malenom podskupu uzoraka. Kao primjer se može zamisliti neki ishod grupne pretrage za kojeg se generira samo ukupno 4 uzorka od ukupno 100,000 uzoraka. Zbog tako malog broja generiranih uzoraka postoji velika mogućnost pogreške pri računanju očekivanja ciljnog izraza (npr. ako su 3 pozitivna primjera očekivanje ishoda je $3/4 = 0.75$). Međutim, propagacija pogreške na konačan iznos očekivane entropije je zanemariva jer se doprinos očekivanoj entropiji računa tako da se entropija ciljnog izraza E_g pomnoži sa p_g ($p_g = 4/100,000 = 0.00004$). Stoga se može zaključiti sljedeće:

- kada je za neki ishod g jako *malen* pripadni podskup uzoraka tada će i pogreška u ukupnoj očekivanoj entropiji biti neznatna jer je p_g jako malen;
- kada je za neki ishod g jako *velik* pripadni skup uzoraka tada će pogreška u ukupnoj očekivanoj entropiji opet biti malena jer će izračun biti proveden na velikom broju uzoraka.

U slučajevima kada su svi ishodi grupirane pretrage podjednako vjerojatni, skup uzoraka se dijeli na podskupove podjednake veličine. Primjerice, ako jedna grupirana pretraga ima 10 elementarnih pretraga sa po dva ishoda, ukupno je moguće 1024 ishoda grupirane pretrage. Tada se skup od 100,000 uzoraka dijeli na podjednake podskupove od oko 100 uzoraka po podskupu. Ako korisnik nije zadovoljan sa ovom točnošću mora se povećati ukupan broj generiranih uzoraka.

Konkretna implementacijska značajka kao što su vremena izvođenja rasuđivanja te pogreška učinjena pri rasuđivanju detaljnije se razmatraju u poglavlju 6. No prije samog korištenja implementirane baze znanja potrebno je implementirati i sučelje prema stvarnim podacima promatranog pacijenta kako bi se omogućilo rasuđivanje sa konkretnim (i stvarnim) značajkama promatranog pacijenta.

5 Uključivanje činjeničnog znanja

U ovom poglavlju opisuje se formalizam koji je razvijen za potrebe integracije stvarnih značajki pacijenata iz medicinske baze podataka u proces rasuđivanja. Osnovne značajke formalizma i prva implementirana verzija nazvana KBDB-ETL (engl. *Knowledge Base from Data Base – Extraction, Transformation and Loading*) predstavljena je u [41]. U međuvremenu je sustav nadograđen nešto složenijim konstruktima i XML sintaksom te je preimenovan u *OwlMapper*. Implementirani sustav za integraciju činjeničnog znanja također je sastavni dio HEARTFAID platforme.

5.1 Činjenično znanje u medicinskom okruženju

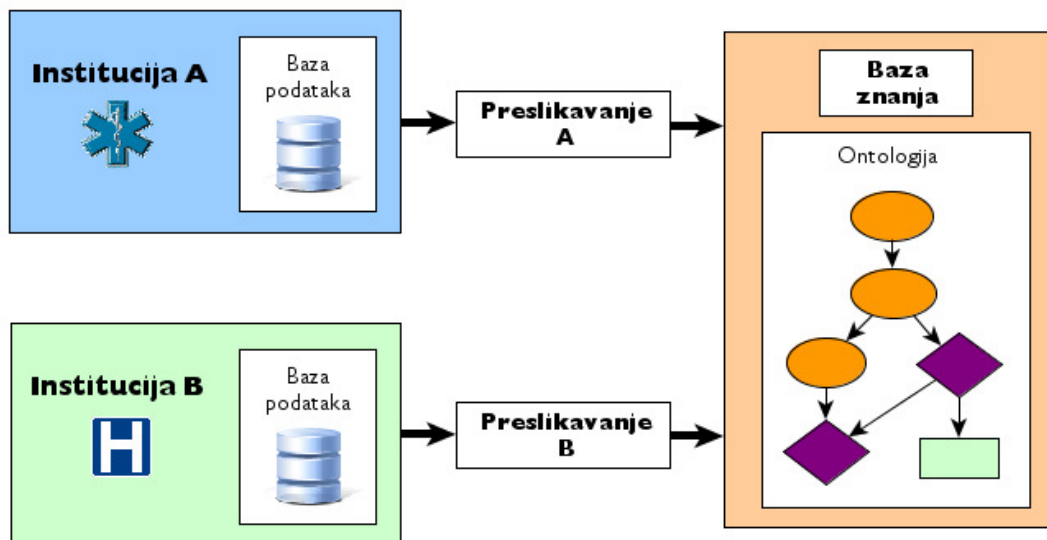
Zaključci donešeni u sustavu za potporu pri odlučivanju donose se na temelju individualnih značajki pojedinog pacijenta koje su predstavljene podacima u bazi podataka. Danas se u medicinskim institucijama podaci o pacijentima redovito pohranjuju u lokalnim bazama podataka. Stoga je pri izgradnji medicinskog sustava za potporu pri odlučivanju važno omogućiti pristup tim podacima u trenutku potrebe.

Termin *činjenično znanje* (engl. *factual knowledge*) predstavlja činjenice u sustavu za potporu pri odlučivanju na temelju kojih se donose zaključci. U medicinskim primjenama može se reći da je činjenično znanje u svom velikom dijelu predstavljeno podacima u medicinskim bazama podataka.

U općem slučaju su medicinske baze podataka od institucije do institucije po strukturi i organizaciji znatno različite jedna od druge. Drugim riječima, još nije usvojen standard po kojem bi se medicinski podaci u uniformnoj strukturi pohranjivali u medicinskim institucijama. Danas se ulažu veliki naponi u standardizaciju oblika medicinskih podataka: krajnji cilj je ostvarenje uniformnog *elektronskog registra pacijenata* (engl. *electronic patient record*) i/ili njegove još općenitije inačice *elektronskog zdravstvenog registra* (engl. *electronic health record*) [4][31]. S obzirom na postojeće različite baze podataka među institucijama, postupak ekstrakcije činjeničnog znanja danas se nužno implementira specifično od institucije do institucije.

Struktura ontologije u bazi znanja je čvrsta i unaprijed definirana. Ontologija definira medicinsko znanje na temelju kojega se donose zaključci pri korištenju sustava za potporu pri odlučivanju. Stoga postupak uključivanja činjeničnog znanja ne može mijenjati postojeću strukturu ontologije nego joj samo dodavati nove instance, ostvarivati neke relacije među već postojećim instancama te razmješati odgovarajuće podatkovne vrijednosti u za to predviđena mjesta u ontologiji.

Okruženje u kojem se odvija uključivanje činjeničnog znanja prikazano je na Slici 5.1. Dvije medicinske institucije koje žele koristiti sustav za potporu pri odlučivanju moraju implementirati specifična preslikavanja iz relacijske baze podataka prema ontologiji u bazi znanja.



Slika 5.1 Postupak uključivanja činjeničnog znanja specifičan je za svaku pojedinu medicinsku instituciju.

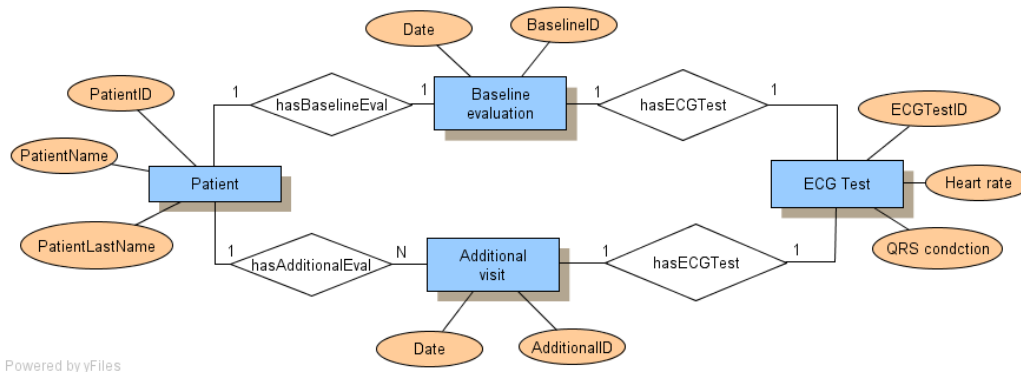
Sumarno gledano, mogu se izdvojiti sljedeći čimbenici koji karakteriziraju problem uključivanja činjeničnog znanja u sustavu za potporu pri odlučivanju:

- Struktra ontologije zadana je unaprijed (izgrađenom bazom znanja) i nije ju moguće procesom preslikavanja mijenjati.
- Struktura relacijske baze podataka unutar jedne medicinske institucije također je zadana unaprijed i nije je moguće procesom preslikavanja mijenjati.
- Modeli relacijske baze podataka mogu se znatno razlikovati od institucije do institucije.
- Struktura i ontologije i baze podataka može biti mijenjana (unaprijeđivana) kroz vrijeme.

5.1.1 Medicinski podaci i relacijske baze podataka

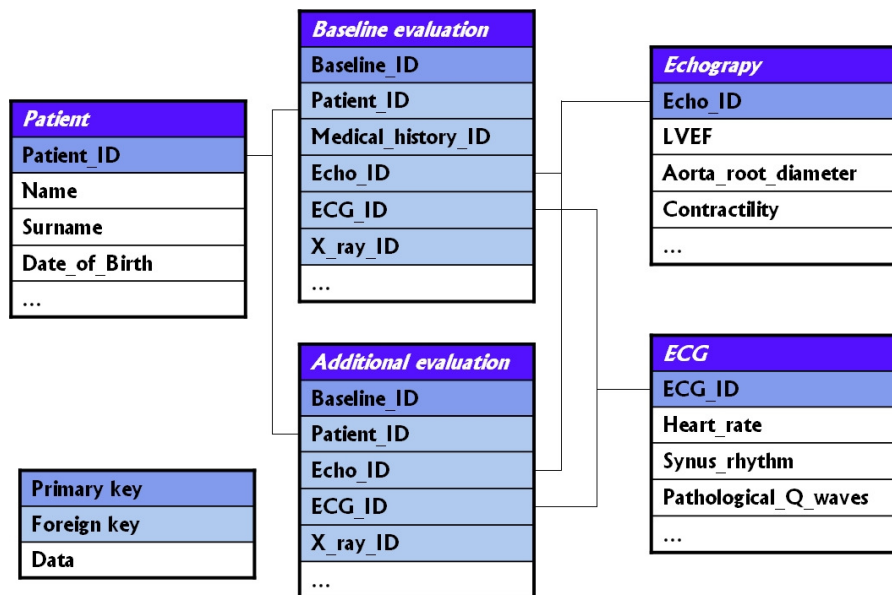
Poželjna karakteristika modela podataka osim osnovne funkcionalnosti pohranjivanja podataka i efikasne implementacije jest da je intuitivne strukture kako bi bio jednostavan za uporabu (postavljanje upita). Često se prije implementacije samog relacijskog modela

koristi *konceptualni* ER model (engl. *entity-relationship*) za prikaz entiteta potrebnih u modelu i relacija među njima. Primjer ER dijagrama prikazan je na Slici 5.2.



Slika 5.2 Primjer ER konceptualnog modela podataka.

Kvadrati na slici definiraju entitete (npr. *Patient*, *Baseline Evaluation*, *ECG Test*), rombovi definiraju veze među entitetima i kardinalitet veza (npr. *hasECGTest*) a elipse definiraju svojstva entiteta (npr. *PaientID*, *PatientName*). Osnovna prednost ovog modela jest što je nešto pregledniji od relacijskog modela a istovremeno se vrlo jednostavnim transformacijama lako pretvara u relacijski model podataka primjenjiv u praksi.

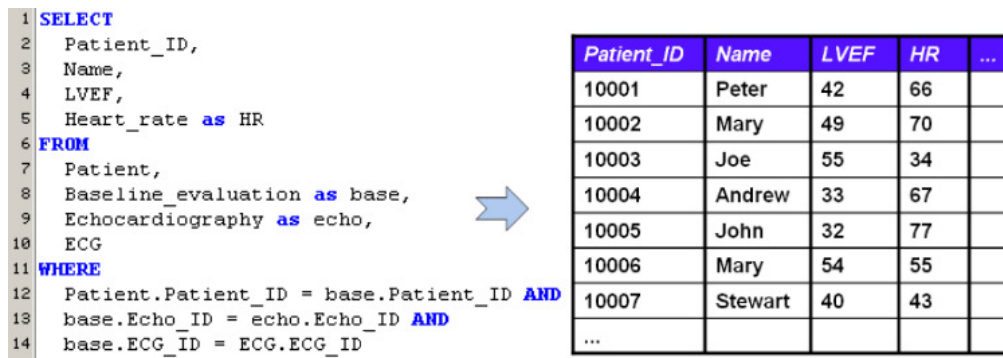


Slika 5.3 Primjer relacijskog modela podataka.

Relacijski model podataka prikladan je za implementaciju sustava zasnovanog na transakcijama. Primjer relacijskog modela dobivenog transformacijom prikazanog ER modela prikazan je na Slici 5.3. Struktura podataka u relacijskom modelu definirana je

tablicama (drugi naziv: *relacijama*) koje sadrže polja koja definiraju oblik pohranjenih podataka. Veze između tablica realizirane su postojanjem ključeva (primarni, strani) dok je kardinalitet veze izravna posljedica strukture tablica i razmjesta stranih ključeva.

Postavljanjem upita na relacijsku bazu podataka (SQL) dobiva se odgovor koji je opet u obliku tablice (relacije). Slika 5.4 prikazuje primjer SQL upita i primjer rezultata koji se njime dobiva.



Slika 5.4 Primjer SQL upita (lijevo) i odgovora dobivenog od baze podataka (desno).

5.1.2 Uključivanje činjeničnog znanja u ontologiju

Kako bi sustav za donošenje odluka mogao uzeti u obzir sve trenutne značajke pojedinog pacijenta zapisane u relacijskoj bazi podataka *nužno je omogućiti transformaciju podataka iz relacijske baze u ontološku strukturu*. Na taj se način činjenično znanje uključuje u proces donošenja odluke. Kako bi se to omogućilo potrebno je definirati formalizam koji podatke dobivene SQL upitima smješta u prikladna mjesta u ontologiji. Taj cilj ilustriran je slikom 5.5.

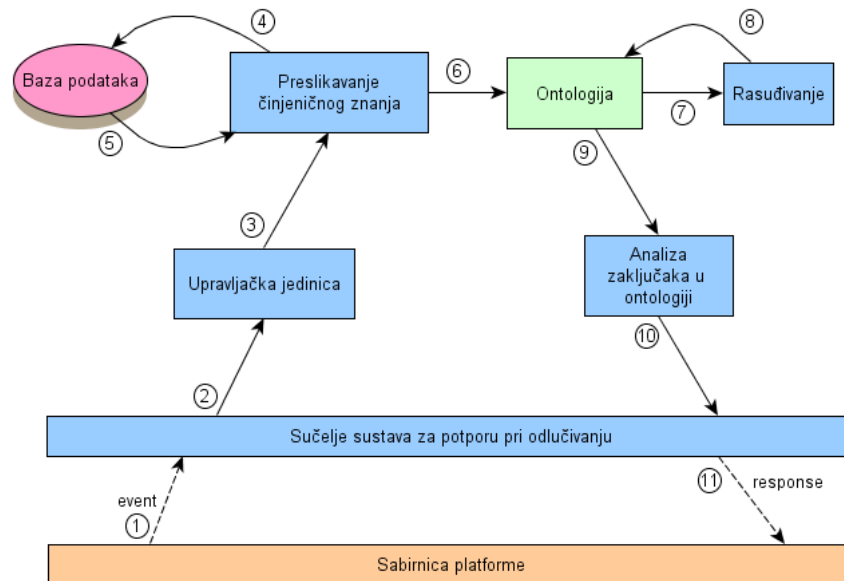


Slika 5.5. Prmjer formalizma koji preslikava podatke dohvaćene SQL upitom u strukturu definiranu ontologijom.

Lijeva strana slike prikazuje primjer formalizma koji je baziran na XML strukturi. Ključno je primjetiti da se SQL upitom definiranom u samom vrhu XML strukture izvlači dio podataka iz baze podataka prikazan tablicom desno. Zatim se dalje navedenim XML elementima točno definira gdje će se u ontologiji razmjestiti dohvaćeni podaci. Razmještanje dohvaćenih podataka u ontološkoj strukturi je i ilustrativno prikazano na desnoj strani slike.

5.1.3 Okruženje za dohvat činjeničnog znanja

Slika 5.6 prikazuje na koji se način proces uključivanja činjeničnog znanja uklapa u ciklus zaključivanja u sustavu za potporu pri donošenju odluka. Kada dođe do potrebe za pokretanje sustava za potporu pri odlučivanju iz platforme se šalje zahtjev (engl. *event*) jedinici preko definiranog sučelja (1-2). U realnom svijetu to se događa npr. kada liječnik na svojem računalu zatraži analizu nekog konkretnog pacijenta. U takvom scenariju sam zahtjev nosi u sebi neke relevantne informacije, primjerice identifikaciju promatranog pacijenta, vrstu usluge koja se zahtjeva i slično.



Slika 5.6. Slijed izvršavanja akcija u jednom ciklus zaključivanja u sustavu za potporu pri odlučivanju.

U tom trenutku pokreće se proces uključivanja činjeničnog znanja u ontologiju (3-6). Pri tome se samo preslikavanje obavlja ovisno o parametrima koji su pristigli u generiranom zahtjevu.

Nakon uključivanja činjeničnog znanja pokreće se proces rasuđivanja (7-8), te se dobiveni zaključci (koji su opet zapisani u ontologiji) analiziraju i šalju preko sučelja

nazad na platformu (9-11). U tom trenutku liječnik na svojem zaslonu dobiva odgovor na svoj upit: rezultate sustava za potporu pri odlučivanju.

Iz opisanog scenarija je vidljivo da je preslikavanje činjeničnog znanja proces koji se učestalo ponavlja, izvršava se praktički u svakom ciklusu pozivanja sustava za potporu pri odlučivanju. Zbog toga sama implemetacija preslikavanja mora biti brza i prikladna za izvršavanje u trenutku potrebe.

5.2 OwlMapper – formalizam za preslikavanje činjeničnog znanja

OwlMapper je formalizam za preslikavanje činjeničnog znanja iz relacijske baze podataka u OWL ontologiju. Realiziran je mahom na temelju zahtjeva definiranih okruženjem HEARTFAID platforme, ali je dovoljno općenit da može poslužiti i u drugim primjenama. *OwlMapper* obavlja preslikavanje činjeničnog znanja implementiranjem sljedećih osnovnih funkcionalnosti:

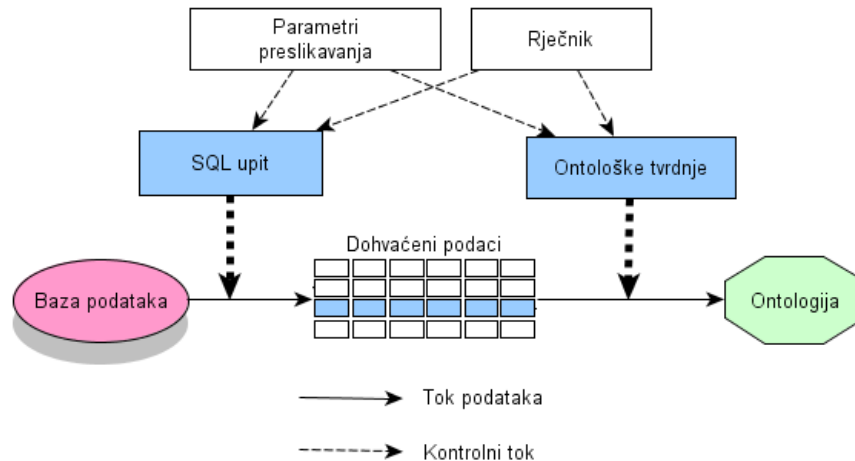
- Dohvat podataka – pomoću SQL upita dohvaćju se podaci iz baze podataka.
- Uključivanje parametara preslikavanja – preslikavanje se izvršava ovisno o parametrima koji mogu biti specifični za svaki pojedini ciklus.
- Rječnik pojmova – za očekivati je da će se neki pojmovi u bazi podataka i ontologiji označavati različitim terminima. Stoga je korisno u procesu preslikavanja omogućiti i rječničke zamjene.
- Razmještanje podataka u ontologiji – dohvaćene vrijednosti potrebno je smjestiti na odgovarajuće mjesto u ontologiji.

5.2.1 Slijed preslikavanja

Program napisan u *OwlMapper* jeziku sastoji se od niza naredbi od kojih svaka sadrži:

- Jedan SQL upit i
- Niz ontoloških tvrdnji.

Slika 5.7 prikazuje slijed kojim se izvršava jedna *OwlMapper* naredba. Prije izvršavanja SQL upit se prilagođava prema parametrima preslikavanja i prema definiranom rječniku. Pomoću tako dobivenog SQL upita dohvaćaju se podaci iz baze podataka u obliku tablice u kojem svaki redak predstavlja jedan *zapis*. Ontološke tvrdnje se također prilagođavaju ovisno o parametrima preslikavanja i o rječniku te se izvršavaju jedna po jedna za svaki zapis dohvaćene tablice zasebno. Prilikom izvršavanja ontoloških tvrdnji u onologiji se stvaraju nove instance, ostvaruju relacije između već postojećih instanci i smještaju podatkovne vrijednosti u predviđena mjesta.



Slika 5.7. Izvršavanje jedne naredbe preslikavanja. SQL upit i ontološke tvrdnje se prilagođavaju ovisno o parametrima preslikavanja i definiranom rječniku.

Kako *OwlMapper* koristi XML sintaksu, njegova se struktura može definirati jednim od postojećih oblika formatiranja XML datoteka. U konkretnoj implementaciji za specifikaciju sintakse korišten je DTD (engl. *Document Type Definition*), no jednako dobra bila bi i bilo koja druga shema (npr. *XML Schema*).

5.2.2 Reference na ontologiju

Definicija sintakse jedne naredbe mapiranja prikazana je sljedećim DTD izrazima:

01	<!ELEMENT mapping (sql, onto+)>
02	<!ELEMENT onto (glossary?, cls, instance, branch*)>
03	<!ELEMENT branch (property, onto+)>
04	<!ELEMENT sql (#PCDATA)>
05	<!ELEMENT cls (#PCDATA)>
06	<!ELEMENT instance (#PCDATA)>
07	<!ELEMENT property (#PCDATA)>

U prvoj liniji definirano je da se jedna naredba mapiranja (element *mapping*) sastoji od točno jednog SQL upita (element *sql*) te jedne ili više ontoloških tvrdnji (element *onto*). Svaka ontološka tvrdnja definira točno jednu ontološku instancu pomoću elemenata *instance* i *cls* koji predstavljaju ime referencirane instance i klasu u kojoj se ona nalazi respektivno. Pomoću elementa *onto* mogu se definirati nove ili referencirati već postojeće instance u ontologiji:

```

01 <onto>
02     <cls>Characteristic</cls>
03     <instance>LowLVEF</instance>
04 </onto>

```

Element *branch* omogućava stvaranje relacija između instanci u ontologiji. Primjerice, moguće je definirati neke karakteristike pacijenta:

```

01 <onto>
02     <cls>Patient</cls>                                PACIJENT
03     <instance>patient_308</instance>
04     <branch>
05         <property>hasCharacteristic</property>
06         <onto>
07             <cls>Characteristic</cls>                SVOJSTVO
08             <instance>LowLVEF</instance>
09         </onto>
10         <onto>
11             <cls>Characteristic</cls>                SVOJSTVO
12             <instance>NormalSBP</instance>
13         </onto>
14     </branch>
15 </onto>

```

Kako su elementi *onto* i *branch* definirani rekurzivno pomoću njih moguće je definirati proizvoljno duboku strukturu ontoloških instanci sa proizvoljno velikim stupnjem grananja na svakoj razini.

5.2.3 Razmještanje vrijednosti iz baze podataka

Element *sql* sadrži SQL upit koji čita podatke iz baze podataka. Pročitane podatke potrebno je smjestiti na odgovarajuća mjesta u definiranoj ontološkoj strukturi. To se omogućava referenciranjem na ime stupca dobivenog SQL upitom pomoću znaka '!'. Sljedeći primjer ilustrira dohvat podataka iz baze i razmještanje dobivenih vrijednosti u ontološku strukturu:

01	<sql>
02	SELECT
03	Patient_ID, Name, LVEF, Heart_rate as HR
04	FROM ...
05	</sql>
06	<onto>
07	<cls>Patient</cls>
08	<instance>patient_![Patient_ID]</instance>
09	<branch>
10	<property>hasLVEF</property>
11	<onto>
12	<cls>Float</cls>
13	<instance>![LVEF]</instance>
14	</onto>
15	</branch>
16	</onto>

U elementu *sql* napisan je SQL upit koji izvršavanjem dohvaća tablicu s podacima (primjer prikazan na slici 5.4). Ontološka tvrdnja opisana *onto* elementom izvršava se za *svaki pojedini zapis* u dohvaćenoj tablici, i to na način da se pomoću znaka ‘!’ i uglatih zagrada referencira na imena stupaca u SQL upitu (označeni crvenom bojom).

5.2.4 Rječnik

Rječničke zamjene se definiraju elementom *glossary*. Na svakoj razini grananja moguće je zasebno definirati jednu instancu rječnika. Sintaksa rječnika definirana je na sljedeći način:

01	<!ELEMENT glossary (item*)>
02	<!ELEMENT item (#PCDATA)>
03	<!ATTLIST item source CDATA "">
04	<!ATTLIST item target CDATA "">

Jedan rječnik se sastoji od niza rječničkih zamjena nazvanih *item*. Svaka rječnička zamjena ima naznačenu riječ koja se zamjenjuje (*source*) i riječ kojom se zamjenjuje (*target*). Reference na rječnik se uvode pomoću znaka ‘@’ i uglatih zagrada. Sljedeći primjer ilustrira uporabu rječničkih zamjena:

```

01 <onto>
02     <glossary>
03         <item source="PAS" target="SBP">
04         <item source="PAD" target="DBP">
05     </glossary>
06     <cls>Characteristic</cls>
07     <instance>Normal@[PAD]</instance>
08 </onto>

```

Iako je navedeni primjer uporabe jednostavan i u praktičnoj uporabi beskoristan, pokazuje se da se pomoću rječnikih zamjena mogu elegantno implementirati mnogo kompleksnija i u praksi veoma korisna preslikavanja.

5.2.5 Promjenjivi parametri preslikavanja

Preslikavanja je moguće obavljati ovisno o trenutno aktivnim parametrima sustava. Na taj način se omogućavaju specijalizirana preslikavanja ovisno o situaciji. Primjerice, preslikavanja je često potrebno obaviti samo za jednog, trenutno promatranog pacijenta. Činjenica da se trenutno promatra samo neki konkretan pacijent se prosljeđuje sustavu za preslikavanje preko vrijednosti parametra (npr. parametar ID=10005). U naredbi mapiranja ovaj se parametar postavlja na proizvoljno mjesto u programu; u ovom konkretnom slučaju unutar SQL upita. Referenciranje na neki konkretan parametar naznačava se oznakom '\$' i uglatim zagradama.

```

01 <sql>
02     SELECT
03         Patient_ID, Name, LVEF, Heart_rate as HR
04     FROM ...
05     WHERE ...
06         AND Patient.Patient_ID = ${ID}
07 </sql>
08 <onto>
09     <cls>Patient</cls>
10     <instance>patient_${ID}</instance>
11     <branch>
12         ...

```

SQL upit se prije izvođenja transformira zamjenom parametara i tek nakon toga slijedi izvođenje upita. Na ovaj način će se iz baze podataka dohvatiti samo podaci koji su specifični za promatranog pacijenta. Shodno tome, sva obavljena preslikavanja bit će specifična za trenutno prmatranog pacijenta.

5.2.6 Program: niz preslikavanja

Na koncu, potrebno je definirati da se *OwlMapper* program sastoji od niza naredbi preslikavanja. To se pomoću DTD sintakse definira na sljedeći način:

01	<!ELEMENT program (glossary?, mapping+) >
----	---

Osim samog niza naredbi moguće je (opcionalno) unutar programa definirati i rječnik koji vrijedi na globalnoj razini.

5.2.7 Konfiguracija sustava

Sustav za uključivanje činjeničnog znanja potrebno je usmjeriti na konkretnu instancu baze podataka na kojoj se radi. Konfiguracijska datoteka sustava ima sljedeći oblik:

01	149.156.12.29/heartfaid?port=3306&user=mprcela&password=*****
02	WebHeartfaid.xml
03	WebHeartfaidData.xml

Prva linija konfiguracijske datoteke sadrži sve podatke relevantne za spajanje na bazu podataka: IP adresu baze podataka, ime sheme koja se referencira, port na kojem se osluškujaju spajanja na bazu, te korisničko ime i zaporka korisnika. U sljedećim linijama konfiguracijske datoteke slijede imena XML datoteka u kojima se nalaze *owlMapper* programi koji specificiraju konkretne implementacije preslikavanja činjeničnog znanja.

5.3 Primjer uključivanja činjeničnog znanja

U ovom odjeljku je kao ogledni primjer prikazana struktura podataka u bazi podataka HEARTFAID platforme. Na temelju iste prikazano je i nekoliko konkretnih primjera preslikavanja implementiranih u HEARTFAID platformi.

5.3.1 Korisničko sučelje HEARTFAID platforme

Podaci koji se koriste u HEARTFAID platformi su od strane medicinskih eksperata (medicinskih partnera na projektu) definirani u obliku formulara za praćenje pacijenata sa srčanim zatajenjem (*CRF*, engl. *Case Report Form*). *CRF* formulari su uobičajeni u medicinskoj praksi za praćenje stanja pacijenta, najčešće na kliničkim studijama ali i u mnogim drugim primjenama.

Na temelju *CRF*-a izgrađeno je i računalno sučelje za medicinske eksperte preko kojega se unose podaci o pojedinim pacijentima. Pri tome je podatke moguće unositi ručno, automatskim očitavanjem direktno sa nekih mjernih uređaja (medicinske opreme) ili računalnom obradom signala i slike.

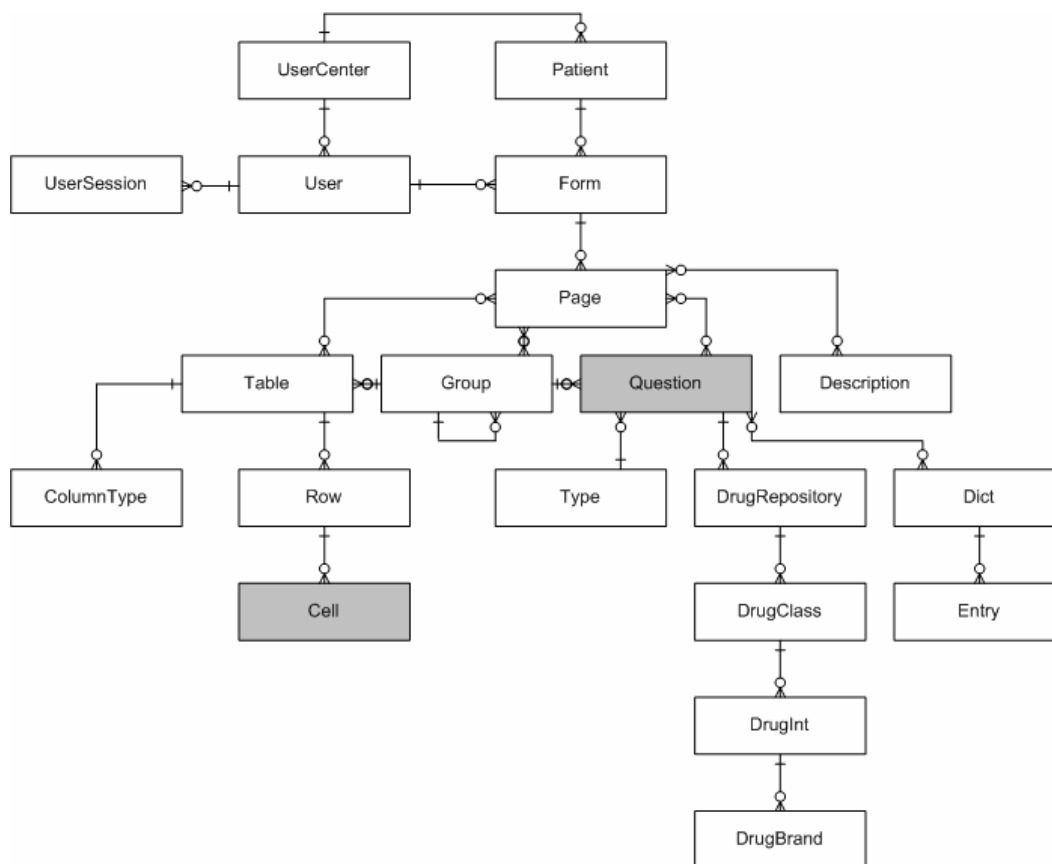


Slika 5.8. Korisničko sučelje HEARTFAID platforme za ručni unos podataka.

Slika 5.8 prikazuje sučelje za unos podataka u internet pregledniku. Na slici je aktivna stranica za unos laboratorijski podataka pri prvom posjetu nekog pacijenta (engl. *Baseline evaluation*). U izborniku na desnoj strani sučelja moguće je odabrati medicinsku pretragu koju promatrani pacijent upravo obavlja.

5.3.2 Relacijski model: HEARTFAID eCRF

Na temelju strukture CRF-a izgrađen je relacijski model podataka koji se koristi za pohranu podataka na HEARTFAID platformi. Pojednostavljeni ER model implementirane baze podataka prikazan je na Slici 5.9. Slika prikazuje samo entitete definirane u modelu, a djelomično i veze među entitetima te njihove kardinalitete. Polja pojedinog entiteta su radi jednostavnosti prikaza odstranjena iz modela.



Slika 5.9. Pojednostavljeni ER model HEARTFAID baze podataka za pohranjivanje podataka sakupljenih eCRF sučeljem.

Neki od ključnih koncepata prikazani na slici su:

- *Patient* – pohranjuje osnovne podatke o pacijentu, mahom identifikacijske prirode,
- *Form* – svaki posjet liječniku se u bazi podataka registrira jednom formom.
- *Question* – medicinski podaci u formi se pohranjuju kao odgovori na pitanja pri čemu svako pitanje ima svoj identifikator. Pitanja su organizirana u grupe i podgrupe pitanja (*Group*), koje se nalaze na stranicama (*Page*) forme.

Kako bi se dohvatio neki određeni podatak (npr. ritam otkucaja na EKG-u) za nekog konkretnog pacijenta potrebno je znati ID promatranog pacijenta i ID traženog pitanja te povezati ova dva podatka preko tablica forme, polja i stranice.

5.3.3 Primjeri preslikavanja

U prvom primjeru preslikavanja se iz baze podataka čitaju podaci koji pripadaju nekom konkretnom pacijentu te se zatim koristeći definirane zamjene transformiraju u ontološke

instance i organiziraju u ontološku strukturu. Jedini ulazni parametar programu je PAT_ID koji nosi identifikaciju trenutno promatranog pacijenta.

```

01 <sql>
02     select
03         question.skey, question.svalue
04     from
05         patient, form, page, question, pages_questions as pq
06     where
07         patient.id = form.patient_id
08         and form.id = page.form_id
09         and page.id = pq.page_id
10         and pq.question_id = quest.id
11         and patient.id = ${PAT_ID}
12 </sql>
13 <onto>
14     <glossary>
15         <item source="hfcrf_9_3==1" target="QwavesAbnormal" />
16         <item source="hfcrf_9_3==0" target="QwavesNormal" />
17
18         <item source="hfcrf_8_8==1" target="FatiguePresent" />
19         <item source="hfcrf_8_8==0" target="FatigueAbsent" />
20
21     </glossary>
22     <cls>Patient</cls>
23     <instance>patient_${PAT_ID}</instance>
24     <branch>
25         <property>hasCharacteristic</property>
26         <onto>
27             <cls>Characteristic</cls>
28             <instance>@[! [skey]==! [svalue]]</instance>
29         </onto>
30     </branch>
31 </onto>

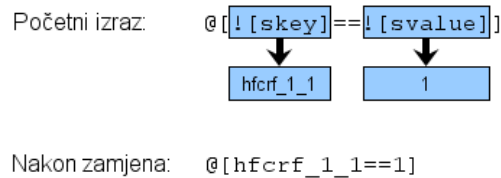
```

Tablica 5.1. Primjer rezultata koji se dohvaća SQL upitom.

skey (<i>identifikator pitanja</i>)	svalue (<i>odgovor na pitanje</i>)
hfcrf_1_1	1
hfcrf_1_2	46
hfcrf_8_8	0

Kako bi se dokućilo značenje ovih naredbi potrebno je prvo pogledati kako izgleda rezultat koji se dobiva SQL naredbom. Primjer dohvaćenog rezultata prikazan je u Tablici 5.1.

Ključna linija u prikazanom preslikavanju je linija 28. U njoj se kombiniraju rječničke zamjene i referenciranje na podatke dohvaćene SQL naredbom. Prilikom izvođenja programa promatra se zapis po zapis dohvaćene tablice. Kod prvog zapisa se u liniji 28 programa jednostavnim zamjenama izvršava transformacija opisana Slikom 5.10.



Slika 5.10. Uzastopnim transformacijama u izrazu dolazi se do konačnog rezultata.

U ovom trenutku pokušava se obaviti i rječnička zamjena prema dobivenom nizu u uglatim zagrada. S obzirom da takva rječnička zamjena nije definirana u rječniku programa dobiveni se redak zanemaruje i izvođenje se nastavlja sljedećim redkom tablice. Sličan scenarij događa se i za drugi redak tablice (redak se zanemaruje zbog nepostojeće rječničke zamjene).

Tek kod trećeg redka tablice događa se uspješna rječnička zamjena te se ulazni niz zamjenjuje nizom *FatigueAbsent*. Rezultantni XML kod je sljedeći:

01	<code><cls>Patient</cls></code>	PACIJENT
02	<code><instance>patient_10005</instance></code>	
03	<code><branch></code>	
04	<code> <property>hasCharacteristic</property></code>	
05	<code> <onto></code>	
06	<code> <cls>Characteristic</cls></code>	SVOJSTVO
07	<code> <instance>FatigueAbsent</instance></code>	
08	<code> </onto></code>	
09	<code></branch></code>	

Na ovaj način se konkretnoj instanci pacijenta u ontologiji dodijelila činjenica da ne postoji problem sa bezrazložnim umorom (*FatigueAbsent*). Ovaj se zaključak u ontologiju unio na temelju stvarnih podataka koji su aktualni u bazi podataka. Dodatnim proširivanjem rječnika se u ovom konkretnom primjeru može na isti način preslikati velik broj značajki pacijenta samo jednim upitom u bazu podataka.

U drugom primjeru uključivanja činjeničnog znanja demonstrira se preslikavanje podatkovnih vrijednosti iz baze podataka u ontologiju. Na slici 5.5 prikazan je primjer

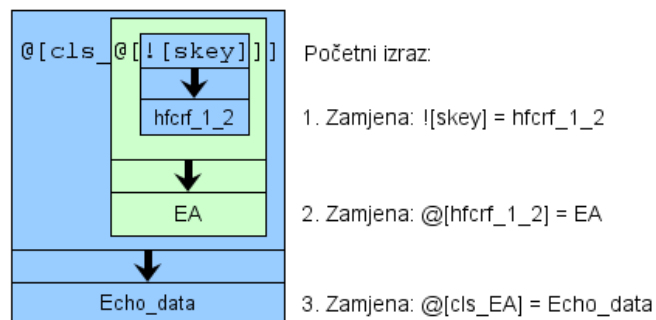
ontološke strukture po kojoj se trenutno promatranom pacijentu dodjeljuju podatkovne vrijednosti.

```

01 <glossary>
02   <item source="hfcfrf_1_2" target="EA" />
03   <item source="cls_EA" target="Echo_data" />
04   <item source="instance_EA" target="Echo_data_instance" />
05   <item source="property_EA" target="E_A_ratio" />
06 </glossary>
07 <cls>Patient</cls>
08 <instance>patient_${PAT_ID}</instance>
09 <branch>
10   <property>hasData</property>
11   <onto>
12     <cls>@[cls_@[! [skey]]]</cls>
13     <instance>@[instance_@[! [skey]]]</instance>
14     <branch>
15       <property>@[property_@[! [skey]]]</property>
16       <onto>
17         <cls>Float</cls>
18         <instance>![svalue]</instance>
19       </onto>
20     </branch>
21   </onto>
22 </branch>

```

Ovaj primjer je vrlo ilustrativan jer pokazuje kako se svaki podatak uredno smješta na svoje mjesto koristeći isti *owlMapper* kod preslikavanja. To se postiže korištenjem uzastopnih rječničkih zamjena. Slika 5.11 prikazuje na koji se način nizom izmjena izračunava ime klase (*Echo_data*) u koju se smještaju podaci očitani Ehokardiografijom (linija 12).



Slika 5.11. Slijed transformacija pri otkrivanju imena klase za smještanje očitanoeg podatka.

Slično se transformiraju i izrazi za ime instance (linija 13) i ime svojstva (linija 15) te se na taj način otkriva potpuna informacija o lokaciji u ontologiji na koju je potrebno zapisati očitanu vrijednost E/A omjera.

U sljedećem retku dohvaćene tablice podataka može se nalaziti neki drugi podatak, primjerice brzina otkucaja srčanog ritma očitana na EKG-u. Taj će se podatak također razmjestiti na prikladno mjesto pomoću istog koda, samo uz dodavanje nekoliko potrebnih rječničkih zamjena:

01	<code><item source="hfcrf_6_5" target="HR" /></code>
02	<code><item source="cls_HR" target="ECG_data" /></code>
03	<code><item source="instance_HR" target="ECG_data_instance" /></code>
04	<code><item source="property_HR" target="Heart_rate" /></code>

Shodno tome, kako bi se postiglo potpuno preslikavanje podataka iz baze u ontologiju potrebno je dalje definirati adekvatne rječničke zamjene za sve postojeće podatke u bazi podataka.

5.4 Diskretizacija podatkovnih vrijednosti

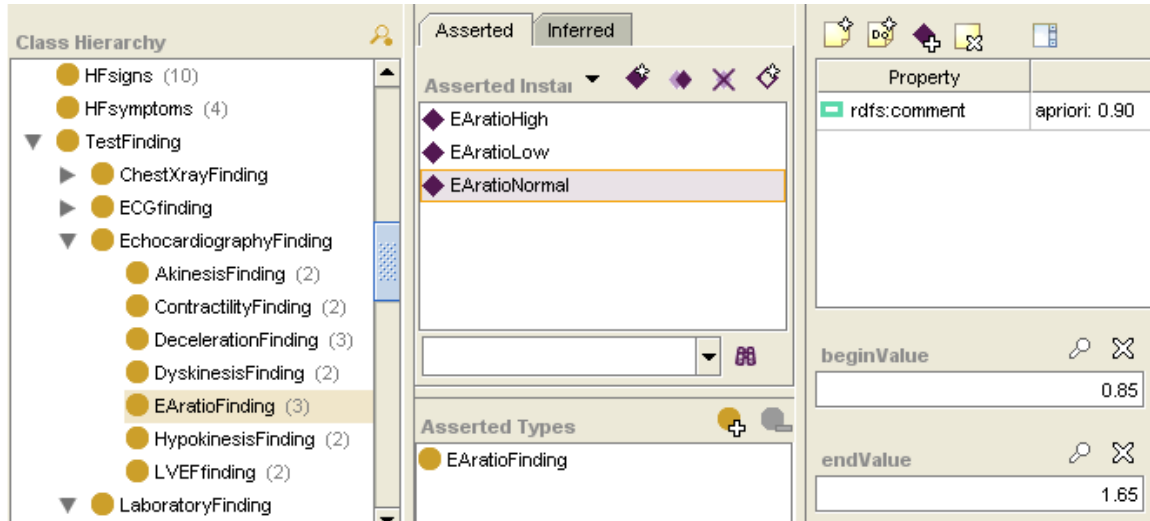
Podatkovne vrijednosti pročitane iz baze podataka nisu prikladne za korištenje u predloženoj metodologiji s obzirom da i deskriptivna logika i Bayesove mreže podrazumijevaju korištenje isključivo nominalnih vrijednosti varijabli. Stoga je nužno podatkovne vrijednosti diskretizirati s obzirom na neke unaprijed definirane referentne intervale. Time se obavlja dodatna obrada činjeničnog znanja prije samog postupka rasuđivanja u integriranom sustavu.

Intervali diskretizacije definirani su u prvom redu uobičajenom medicinskom praksom (npr. referentni interval za razinu crvenih krvnih stanica u analizi krvi za osobe ženskog spola je 3.8-5.2 T/l) a u nekim se slučajevima procjenjuju i po subjektivnom osjećaju medicinskog eksperta (primjerice definiraju se granice u godinama za mlađe pacijente, pacijente u srednjoj dobi i starije pacijente).

Ove je intervale moguće definirati u postupku integracije činjeničnog znanja prikladnim korištenjem SQL naredbi. Međutim, pokazuje se da postoji mnogo elegantnije rješenje pomoću kojega se diskretizacija podatkovnih vrijednosti prenese u proces rasuđivanja. Pri tome se referentne granice intervala pojedinih mjerenja definiraju unutar ontologije. Ovakvo rješenje je mnogo jednostavnije za rukovanje i za eventualne popravke a osim toga dojam je i da više pristaje duhu izgradnje baze znanja.

S obzirom da OWL jezik ne podržava manipulaciju podatkovnim vrijednostima, uvodi se proširenje baze znanja SWRL pravilima. SWRL (engl. *Semantic Web Rule Language*) je jezik koji se temelji na kombinaciji OWL-DL i OWL-Lite ontoloških jezika sa RuleML

jezikom (engl. *Rule Markup Language*). Jezik je nastao u nastojanju da se OWL jezik nadogradi sa semantikom produkcijskih pravila. SWRL pravila slijede uobičajenu sintaksu produkcijskih pravila (uvjet i posljedica) i uobičajenu semantiku produkcijskih pravila (kada je uvjet istinit prihvaća se da je i posljedica istinita).



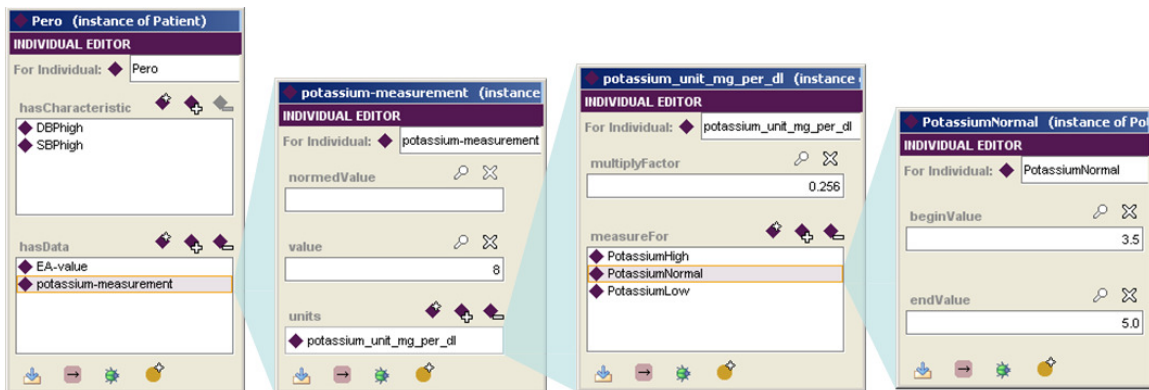
Slika 5.12. Interval za normalnu razinu E/A omjera definira se postavljanjem granica intervala (0.85-1.65).

Slika 5.12 prikazuje na koji se način za određenu karakteristiku pacijenta mogu definirati referentne granice intervala. Kada se za sve instance na isti način definiraju granice intervala, može se napisati jedinstveno pravilo koje će promatranu instancu dodijeliti promatranom pacijentu kao karakteristiku.

Struktura ontologije koja omogućuje diskretizaciju podatakovnih vrijednosti SWRL pravilima prikazana je na Slici 5.13. Instanci pacijenta (*Pero*) se u postupku uključivanja činjeničnog znanja prilikom preslikavanja jednog podatka (mjerjenja) iz baze podataka dodijele dvije činjenice:

- vrijednost mjenog podatka i
- jedinica mjenog podatka.

U primjeru na Slici 5.13 pacijentu *Pero* izmjerena je vrijednost kalija u krvi koja iznosi 8, u mjernim jedinicama mg/dl. Ta dva podatka dodjeljuju se pacijentu prilikom uključivanja činjeničnog znanja. Kako bi se pročitana jedinica pretvorila u referentnu jedinicu (mmol/l) iznos se pomnoži sa faktorom 0.256 i pohrani u polje *normedValue*. Zatim se provjerava pripada li dobivena vrijednost u označeni interval (3.5-5.0); ako pripada pacijentu se dodjeljuje karakteristika *PotassiumNormal*. Na sličan način definirani su i intervali za nisku i visoku razinu kalija u krvi.

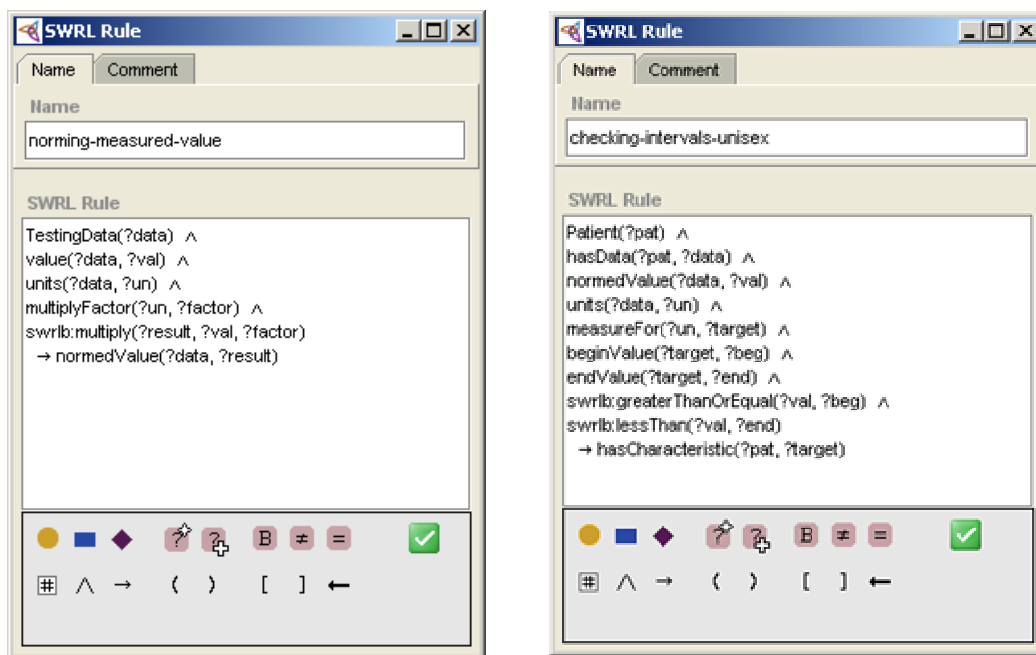


Slika 5.13. Struktura ontologije koja omogućava diskretizaciju podatkovnih vrijednosti pomoću SWRL pravila.

Za opisani proces diskretizacije potrebno je definirati dva SWRL pravila:

- pravilo za pretvorbu u referentne jedinice,
- pravilo za provjeru intervala.

Konkretna definicija ovih pravila prikazana je na slici 5.14. Prvo pravilo množi pročitanu vrijednost sa definiranim faktorom kako bi se dobila normirana vrijednost očitavanja. Drugo pravilo provjerava da li je normirana vrijednost unutar definiranog intervala te sukladno tome dodjeljuje prikladnu karakteristiku pacijentu.



Slika 5.14. Diskretizacija svih podatkovnih vrijednosti u sustavu obavlja se sa samo dva SWRL pravila.

Poteškoća nastaje kada su intervali diskretizacije definirani uz dodatne uvjete. Primjerice, normalna razina crvenih krvnih zrnaca u krvi posebno je definirana za muškarce i za žene. Taj problem se može riješiti tako da se u ontologiji napišu odvojeno vrijednosti granica intervala za žene i za muškarce. Kako bi diskretizacija i u tom slučaju ispravno funkcionirala potrebno je napisati dodatna SWRL pravila koja će znati očitati vrijednosti granica intervala na za to predviđenim mjestima.

Korištenje SWRL pravila u sustavu nosi i nekoliko nedostataka:

- Potreban je dodatni alat za rasuđivanje (alat za izvođenje produkcijskih pravila – CLIPS ili Jess),
- Izvođenje pravila dodatno troši vrijeme pri rasuđivanju,
- Napisana pravila su nepregledna (korištenje varijabli itd.).

U praktičnoj se primjeni pokazuje da navedeni nedostaci ne predstavljaju značajnije poteškoće. Alat CLIPS je u potpunosti slobodan za korištenje dok je Jess komercijalan proizvod ali je dostupan za korištenje u akademskoj zajednici. Vrijeme izvođenja se uistinu produljuje, ali je to produljenje neznatno u odnosu na vrijeme koje zahtijevaju ostale komponente upotrijebljene u sustavu (Slika 6.11). Osvrt na vremena izvođenja pojedinih komponenti u sustavu je dan u poglavlju 6.

Na koncu, ostaje još poteškoća što su SWRL pravila sama po sebi nepregledna i neintuitivna. Međutim, kao što smo vidjeli u prikazanom primjeru, u cijelom sustavu je broj pravila koji se koristi jako malen (2 do 10 pravila), a jednom napisana pravila nije potrebno učestalo mijenjati. Stoga se doista može reći da ni nepreglednost SWRL pravila ne predstavlja značajniju poteškoću.

Kako su se navedeni nedostaci pravila pokazali premostivima, a s obzirom na korisnost SWRL pravila, može se zaključiti da je uvođenje SWRL pravila u sustav za potrebe diskretizacije podatkovnih vrijednosti uistinu razumno i opravdano.

6 Vrednovanje funkcionalnih značajki

U ovom se poglavlju opisuje jedan scenarij korištenja sustava za potporu pri odlučivanju koji koristi prikaz znanja temeljen na integraciji ontologija i Bayesovih mreža. Zatim se analiziraju neke funkcionalne značajke sustava pri čemu se fokus stavlja na potrebna vremena izvođenja i pogrešku pri rasuđivanju u ovisnosti o nekim odabranim karakteristikama i postavkama sustava.

Ovdje valja naglasiti da je u službenoj verziji HEARTFAID platforme uključena ontologija (opisno znanje), produkcijska pravila (akcije koje se izvršavaju), sustav za ispis poruka i objašnjenja donešenih zaključaka na korisničko sučelje, te sustav za ekstrakciju činjeničnog znanja (*owlMapper*). Kako je metodologija integracije ontologija i Bayesovim mrežama nastala nešto kasnije (pred kraj projekta) nije bilo mogućnosti integrirati je u službenu verziju platforme. Unatoč tome, razvijena je sva potrebna programska podrška za implementaciju ove metodologije u HEARTFAID platformu. U ovom se poglavlju stoga opisuje samo neslužbeni prototip baze znanja i korisničkog sučelja kako bi se stekao dojam na koji se način može iskoristiti predložena metodologija.

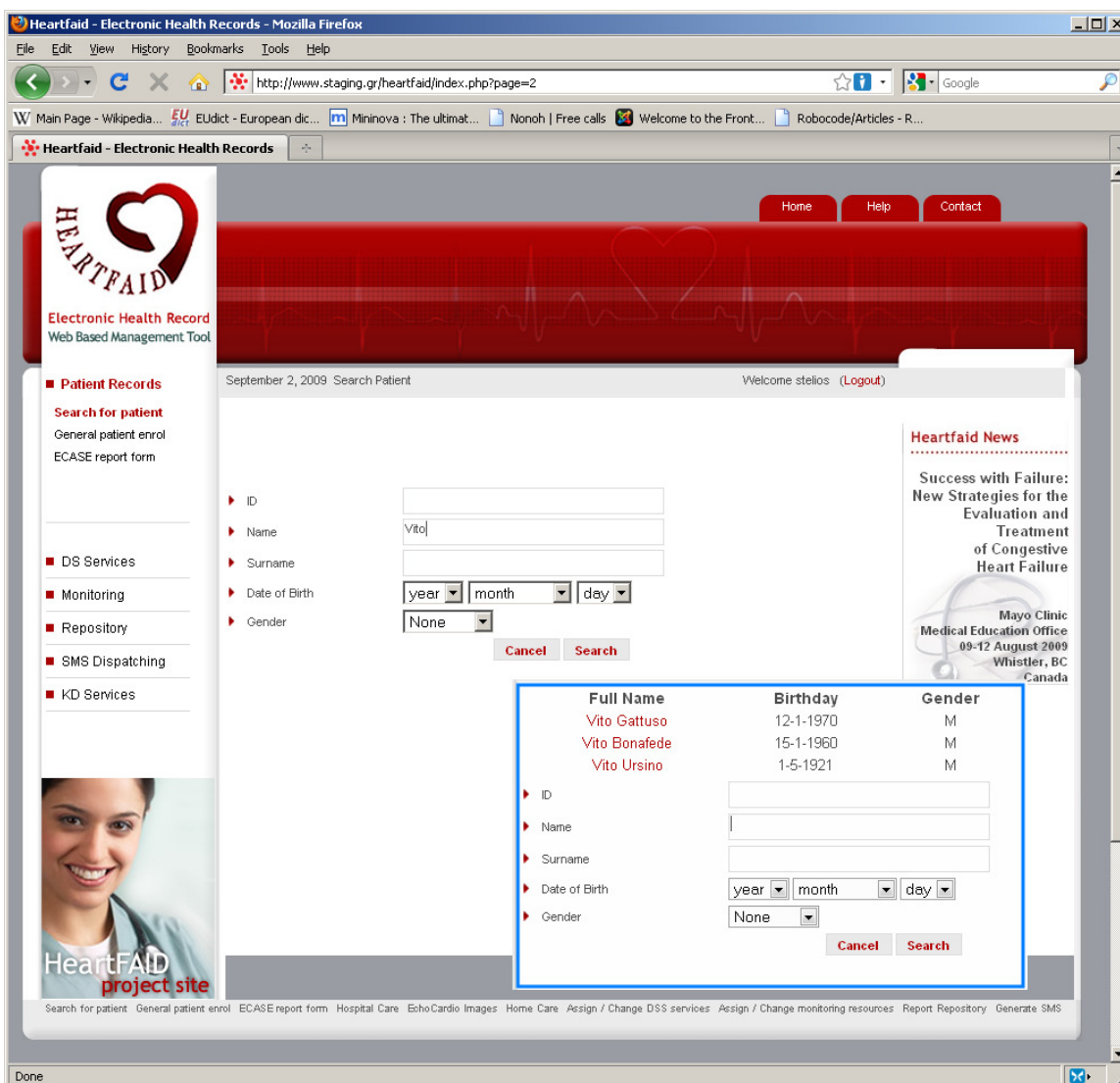


Slika 6.1. Početna stranica HEARTFAID platforme omogućuje prijavu liječnika u sustav.

6.1 Korisničko sučelje: primjer korištenja sustava za podršku pri odlučivanju

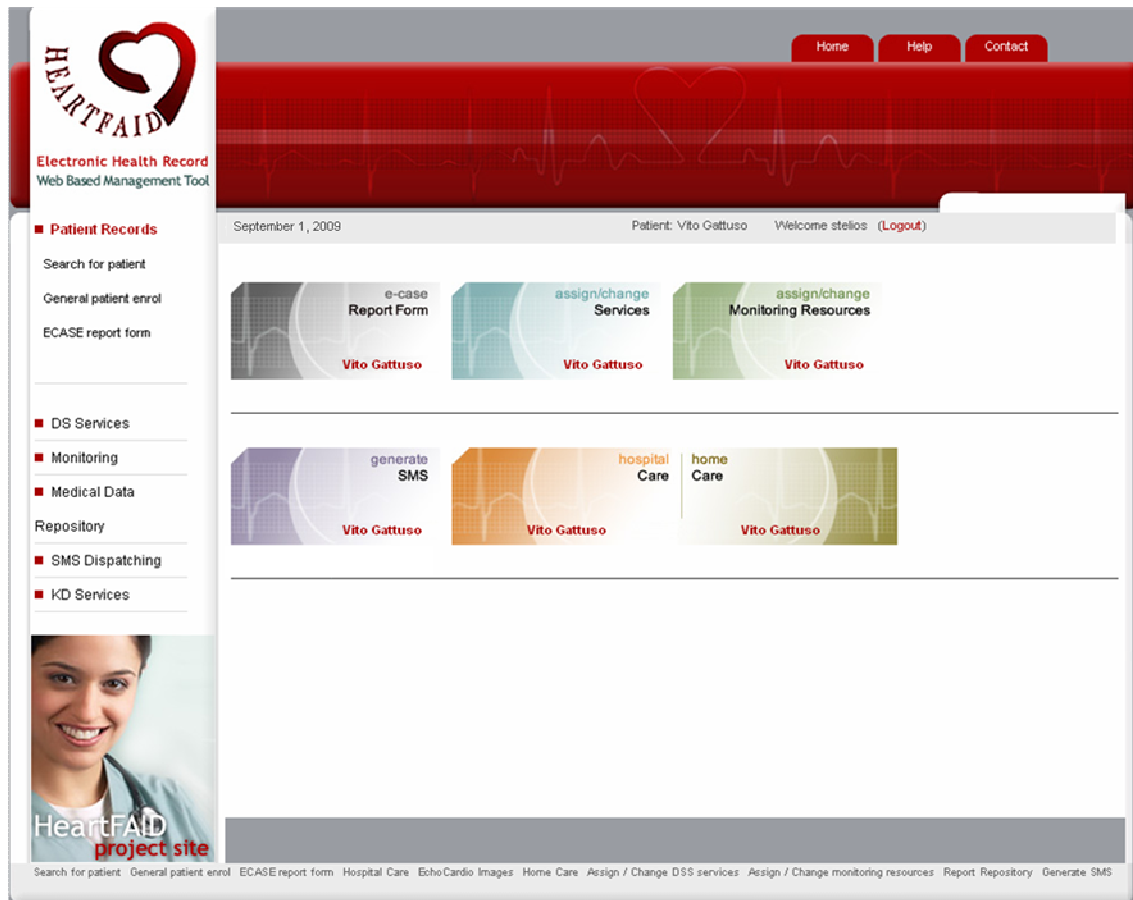
U ovom odjeljku se opisuje jedan primjer korištenja sustava za potporu pri odlučivanju koji se temelji na integraciji ontologija i Bayesovih mreža. Pri tome je naglasak stavljen na samo korisničko sučelje kako bi se dočaralo na koji način liječnici mogu koristiti sustav.

HEARTFAID platformi pristupa se korištenjem internet preglednika. Početna stranica sustava služi za autentikaciju liječnika i njegovu prijavu u sustav. Slika 6.1 prikazuje početnu stranicu platforme.



Slika 6.2. Pretraga pacijenata po osobnim podacima.

Slika 6.2 prikazuje sučelje za pretragu baze pacijenata na HEARTFAID platformi. Sa lijeve strane se nalazi glavni izbornik gdje se nude osnovne usluge platforme. U primjeru na slici otvoren je i podizbornik "Patient records" u kojem se omogućuje unos podataka za potpuno novog pacijenta te pretraga već evidentiranih pacijenata. Kako je odabrana opcija "Search for patient" u sredini stranice nude se razni kriteriji za pretragu pacijenata: ID, ime i prezime, datum rođenja i spol. Kada se pronađe i odabere željeni pacijent liječniku se nudi izbornik prikazan slikom 6.3.



Slika 6.3. Kada je odabran konkretan pacijent platforma nudi liječniku niz usluga od kojih su najvažnije prikazane kao kratice na sredini ekrana.

Slika 6.3 prikazuje izbor koji se nudi liječniku po odabiru konkretne instance pacijenta. Potpuni pregled mogućnosti platforme je dostupan preko glavnog izbornika dok se u sredini ekrana nude kratice za neke od najčešćih i/ili najvažnijih akcija. Već u ovom trenutku moguće je pokrenuti sustav za potporu pri odlučivanju odabirom opcije "Hospital care" na sredini ekrana. Pretpostavimo u ovom trenutku da liječnik prvo želi pregledati povijesne podatke ili unijeti nova zapažanja o pacijentu. Odabirom prve kratice (gore lijevo) dobiva se pregled ranije unesenih podataka koji su zabilježeni u bazi

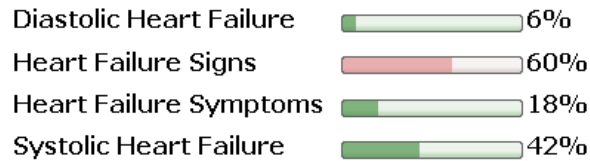
podataka. Liječnik tada ima mogućnost pregledavanja povijesnih podataka (svih prethodno obavljenih posjeta) te unosa podataka za novi posjet.

Slika 6.4 prikazuje unos podataka za novi posjet. Na desnoj strani slike je ponuđen niz aktivnosti koji se mogu obaviti tokom ovog jednog posjeta. Liječnik ima mogućnost odabira aktivnosti i unosa podataka vezanih za odabranu aktivnost. U primjeru na slici odabrana je aktivnost “Other history” te liječnik ima mogućnost unošenja povijesnih podataka pacijenta, npr. prisutnost dijabetesa, problema sa štitnjačom, zatajenja bubrega, itd. Slična situacija je prikazana ranije na slici 5.8 gdje se prikazuje unos podataka pri laboratorijskim pretragama.

Patient: JK461351370		Form: HEARTFAID CRFS - Baseline		Page 5	
2. Other medical history				<ul style="list-style-type: none"> → [1] Demographic → [2] Inclusion → [3] Exclusion → [4] Card. history [5] Other history → [6] Lifestyle&Fam. → [7] Therapy → [8] Anamnesis → [9] Physical → [10] Laboratory → [11] X-Ray → [12] 12-lead ECG → [13] ECG → [14] 24-Holter → [15] Cardiopulm. → [16] Life Quality → [17] Beat-to-beat 	
Endocrine disorders		<input type="radio"/> No <input type="radio"/> Yes			
diabetes		<input type="radio"/> No <input type="radio"/> Yes			
type:		<input type="radio"/> 1 <input type="radio"/> 2			
thyroid disorders		<input type="radio"/> No <input type="radio"/> Yes			
hypothyroidism		<input type="radio"/> No <input type="radio"/> Yes			
hyperthyroidism		<input type="radio"/> No <input type="radio"/> Yes			
Renal failure		<input type="radio"/> No <input type="radio"/> Yes			
Respiratory disorder		<input type="radio"/> No <input type="radio"/> Yes			
chronic obstructive heart lung disease		<input type="radio"/> No <input type="radio"/> Yes			
bronchial asthma		<input type="radio"/> No <input type="radio"/> Yes			
sleep disordered breathing		<input type="radio"/> No <input type="radio"/> Yes			
history of pulmonary thromboembolismg		<input type="radio"/> No <input type="radio"/> Yes			
pulmonary fibrosis		<input type="radio"/> No <input type="radio"/> Yes			
Anemia		<input type="radio"/> No <input type="radio"/> Yes			
Others (potentially related to heart failure diagnosis):		<input type="radio"/> No <input type="radio"/> Yes			
Connective tissue diseases		<input type="radio"/> No <input type="radio"/> Yes			
Sexually transmitted diseases (including HIV)		<input type="radio"/> No <input type="radio"/> Yes			

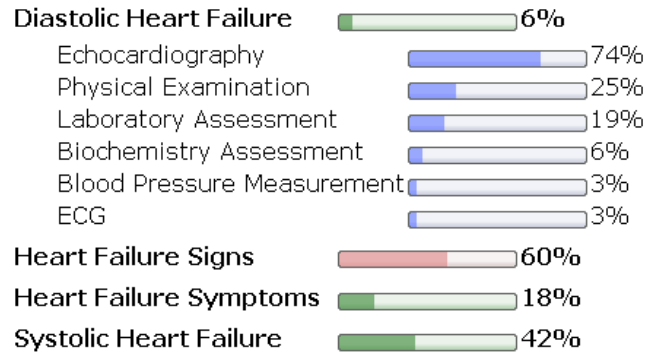
Slika 6.4. Unos povijesnih podataka u sustav prilikom pregleda pacijenta.

Kada su unešeni i pohranjeni izmjereni (trenutno dostupni) podaci moguće je pokrenuti sustav za potporu pri odlučivanju. Ta je opcija dostupna u glavnom izborniku kao “DS services” te odabirom pod-opcije “Hospital care”. Kada završi proces rasuđivanja korisniku se prikažu rezultati rasuđivanja kao niz koncepata i pripadnih vjerojatnosti.



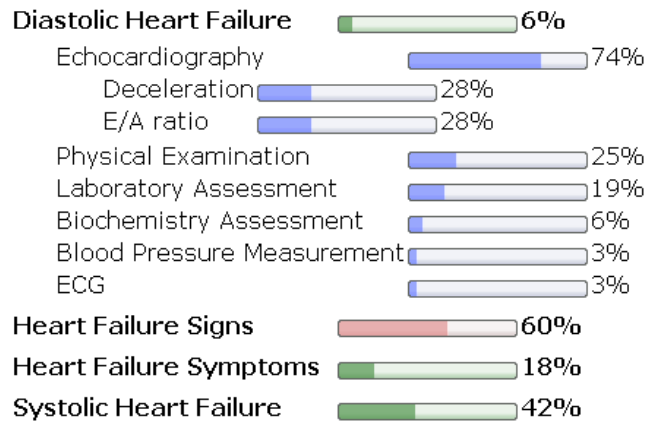
Slika 6.5. Rezultat rasuđivanja pri dijagnozi srčanog zatajenja.

Slika 6.5 prikazuje primjer rezultata koje dobiva liječnik. Crvenom bojom označeni su koncepti kojima je vjerojatnost preko 50% (zelenom ispod 50%). Rezultati pokazuju da promatrani pacijent ima dijastoličko srčano zatajenje s očekivanjem od 6%. Ako liječnik želi dalje razmotriti mogućnost dijastoličkog srčanog zatajenja on može pogledati koji su u tom pogledu najinformativniji testovi koji se mogu obaviti. Lijevim klikom miša na pojam “Diastolic Heart Failure” ispod pojma se otvara se niz testova i pripadnih informativnosti kao što je prikazano na slici 6.6.



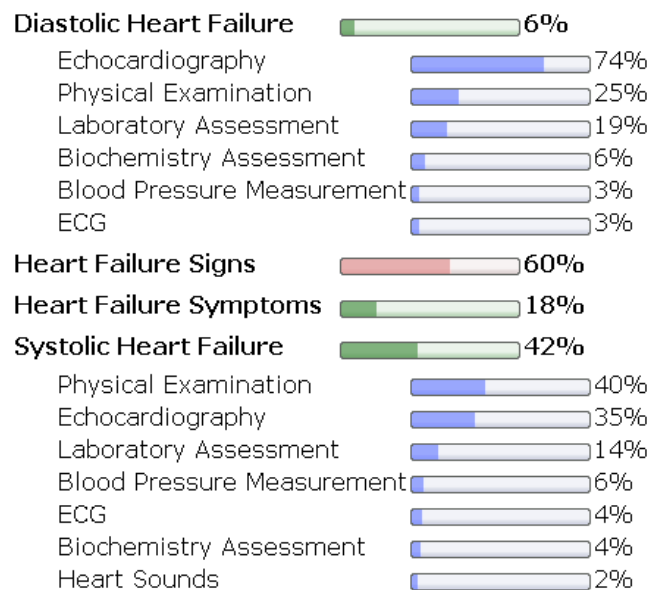
Slika 6.6. Informativnost testova koji se mogu obaviti s obzirom na odabrani ciljni pojam dijastoličkog srčanog zatajenja.

U ovom trenutku liječnika može zanimati i koje su pod-pretrage ključne u pojedinim pretragama (s obzirom na promatranog pacijenta). Lijevim klikom miša na sličicu koja ilustrira informativnost pojma otvara se dodatni podniz pojmova sa pripadajućim informativnostima kako je prikazano na Slici 6.7.



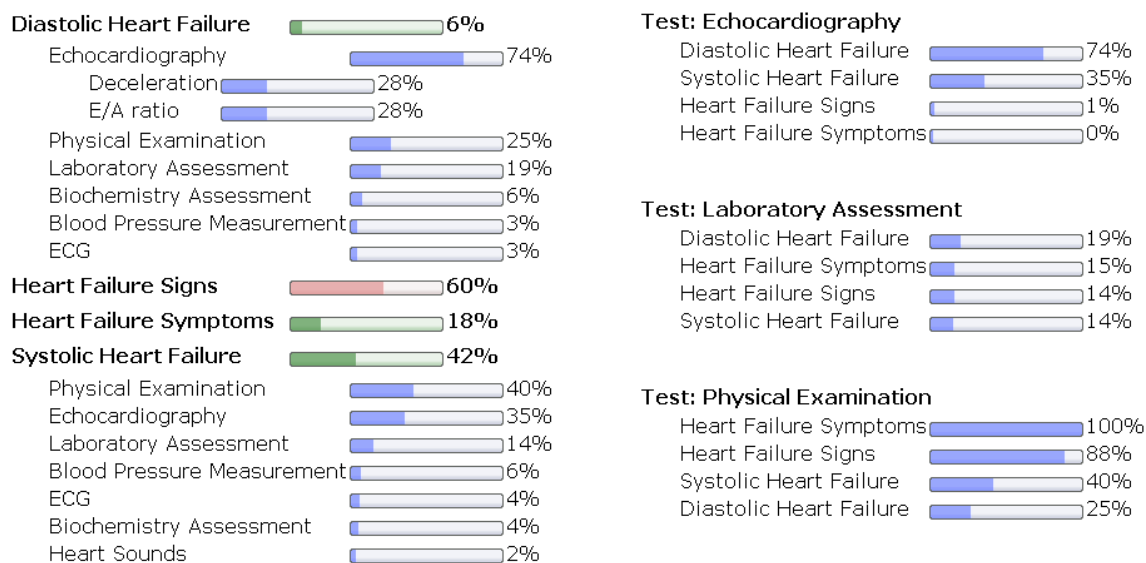
Slika 6.7. Informativnost pojmova se može pregledavati po strukturi koja je definirana ontologijom.

Moguće je i paralelno gledati informativnosti za više ciljnih pojmova odjednom. Ta je opcija prikazana na Slici 6.8. Slika prikazuje paralelan pregled informativnosti testova za dva odabrana pojma: sistoličko i dijastoličko srčano zatajenje.



Slika 6.8. Paralelan pregled informativnosti za više odabranih ciljnih pojmova.

Sada liječnik može odabrati i neki manji podskup testova koji mu se čine najinformativniji i paralelno ih usporediti. Lijevim klikom na tekst pojedine pretrage se sa desne strane otvara njezin sumarni pregled kako je to prikazano na desnoj strani Slike 6.9.

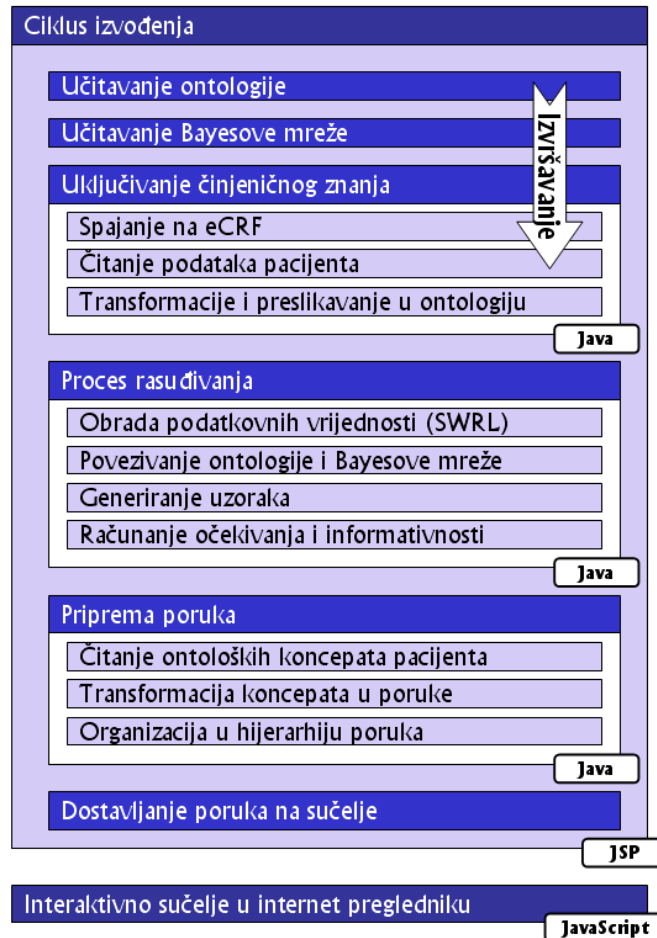


Slika 6.9. Sa desne strane prozora moguće je dobiti sumarni pregled informativnosti odabranih dijagnostičkih pretraga.

S obzirom na sumarne preglede liječnik može zaključiti koji mu se dijagnostički test čini najprikladniji za obaviti. Nakon odluke o obavljanju ehokardiografije i/ili fizičkog pregleda pacijenta te nakon prikupljenih novih saznanja (podataka) o pacijentu (ručnim unošenjem podataka na fizičkom pregledu ili automatskom ekstrakcijom značajki na ehokardiografiji) liječnik može ponovno pokrenuti sustav za potporu pri odlučivanju i vidjeti koja su nova očekivanja ciljnih koncepata te koje su pripadne informativnosti pojedinih pretraga.

6.2 Okruženje sustava

Predviđeni slijed izvršavanja akcija unutar jednog ciklusa u sustavu za potporu pri odlučivanju prikazan je Slikom 6.10. Akcije su organizirane u funkcionalne cjeline i podcjeline. U jednom ciklusu izvođenja učitavaju se ontologija i Bayesova mreža, učitavaju se podaci iz baze podataka specifični za promatranog pacijenta, pokreće se proces rasuđivanja, pripremaju se poruke na osnovu dobivenih zaključaka te se dostavljaju korisniku na interaktivno sučelje u internet pregledniku.



Slika 6.10. Slijed izvođenja akcija u sustavu koji koristi prikaz znanja temeljn na integraciji ontologija i Bayesovih mreža.

U samom procesu rasuđivanja izdvojene su četiri cjeline: obrada podatkovnih vrijednosti, povezivanje ontologije i Bayesove mreže, generiranje skupa uzoraka, te na koncu izračun vrijednosti koji obuhvaća računanje očekivanja ciljnih koncepata i računanje informativnosti svake definirane pretrage za svaki ciljni koncept.

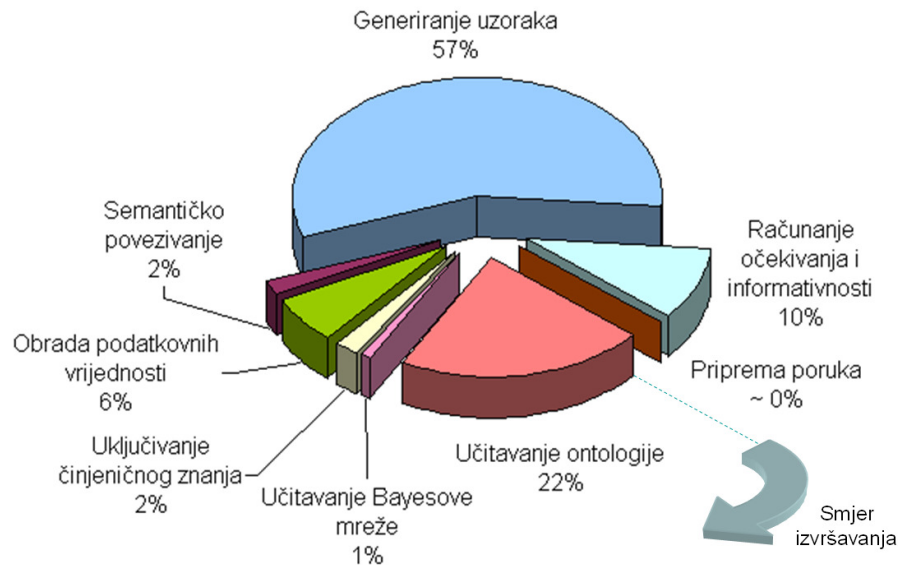
Konkretna implementacija svih navedenih aktivnosti realizirana je u programskom jeziku Java pri čemu se korisničke internet stranice generiraju na serverskoj strani pomoću JSP jezika (engl. *Java Server Pages*). Rezultirajuća internet stranica generirana je u *JavaScript* jeziku kako bi se ostvarilo interaktivno sučelje sa korisnikom sustava prilikom pregledavanja rezultata.

6.3 Vremena izvođenja

Očekivana vremena izvođenja aktivnosti prikazanih na Slici 6.10 međusobno se značajno razlikuju. Već i samo učitavanje ontologije može trajati značajno dugo (npr, ontologija

koja broji oko 100 klasa, 300 instanci i 20 svojstava učitava se u intervalu od 5 do 10 sekundi), dok s druge strane priprema poruka za ispis na ekran traje zanemarivo kratko (red veličine milisekunde).

Kako bi se dobio opći osjećaj o očekivanom trajanju pojedine akcije na Slici 6.11 prikazan je graf koji daje približan omjer trajanja akcija prilikom jednog ciklusa izvođenja sustava za potporu pri odlučivanju. Dakako, izgled ovog grafa značajno ovisi o odabranim parametrima sustava.



Slika 6.11. Očekivana vremena izvođenja.

Iz slike se vidi da najznačajnije dijelove vremena pri izvođenju nose generiranje uzoraka, učitavanje ontologije, računanje očekivanja i informativnosti te obrada podatkovnih vrijednosti, dok ostale akcije zahtjevaju zanemarivo malen udio u vremenu izvođenja.

Sva mjenjenja opisana u nastavku ovog odjeljka obavljena su koristeći jedno serversko računalo sa AMD 64-bitnim procesorom Opteron Dual-Core, uz 4 GB RAM-a.

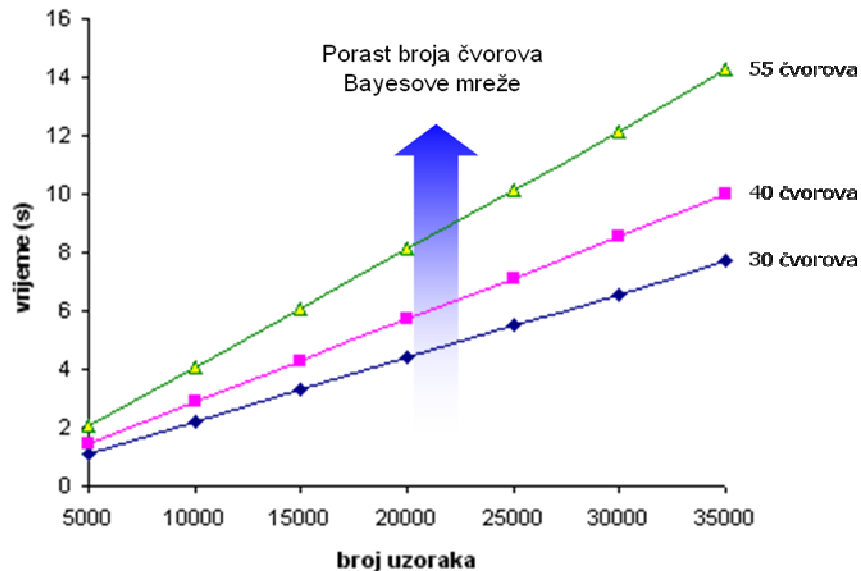
6.3.1 Vrijeme generiranja uzoraka

Vrijeme generiranja uzoraka linearno je ovisno o broju generiranih uzoraka. Vrijeme generiranja *jednog* uzorka linearno ovisi o samoj veličini uzorka koja je pak direktno definirana brojem čvorova u Bayesovoj mreži. Linearnost ovih zavisnosti eksplicitno je vidljiva je iz same definicije algoritma uzorkovanja Bayesove mreže.

U realnoj primjeni postupak generiranja uzoraka unatoč linearnosti traje značajno dugo. Kako bi se dobio osjećaj koliko realno traje generiranje uzoraka prikladno je provesti praktična mjerenja. Mjerenja na Slici 6.12 provedena su uzorkovanjem pomoću *maximum*

likelihood sampling algoritma na mrežama koje su sadržavale ukupno 30, 40 i 55 čvorova. Eksperimentalni podaci prema očekivanjima pokazuju linearne zavisnosti među promatranim parametrima.

Kao što je prikazano na Slici 6.11, vrijeme generiranja uzoraka uzima najviše vremena u ciklusu izvođenja sustava za potporu pri odlučivanju. S obzirom da ono izravno utječe i na točnost pri rasuđivanju biti će od iznimne važnosti broj uzoraka pravilno prilagoditi potrebama sustava.



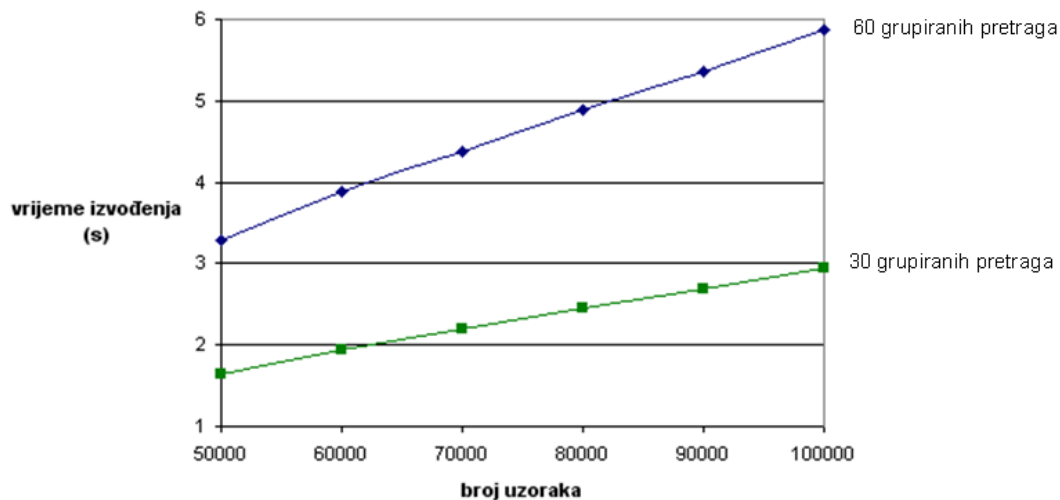
Slika 6.12. Vrijeme generiranja uzoraka linearno ovisi i o broju generiranih uzoraka i o broju čvorova u Bayesovoj mreži.

6.3.2 Vrijeme računanja očekivanja i informativnosti

Računanje očekivanja i informativnosti se iz generiranog skupa uzoraka vrši u samo jednom prolasku kroz skup uzoraka. Pri tome se za svaki pojedini uzorak registrira zadovoljava li on ili ne neke definirane uvjete. Primjerice, za neku definiranu grupiranu pretragu provjerava se u koji ishod grupirane pretrage promatrani uzorak pripada. Stoga se već na temelju samog algoritma rasuđivanja može zaključiti da vrijeme izvođenja pri računanju informativnosti i očekivanja raste:

- linearno sa porastom broja generiranih uzoraka,
- linearno sa porastom broja definiranih ciljnih pojmova i grupiranih testova u ontologiji.

Iako su izmjerena vremena izračuna (Slika 6.13) u realnim primjenama osjetna (sekunde), u usporedbi sa pripadnim vremenima generiranja uzoraka ona su za red veličine manje.



Slika 6.13. Trajanje računanja očekivanja i informativnosti ovisi o broju generiranih uzoraka i o broju grupiranih pretraga.

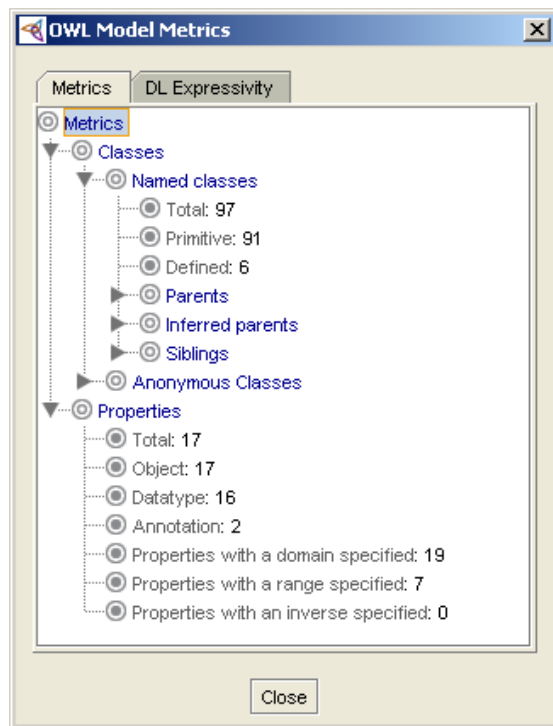
6.3.3 Vrijeme učitavanja ontologije

Vrijeme učitavanja ontologije je važno za promotriti jer već i pri sasvim malenim ontologijama učitavanje ontologije može trajati primjetno dugo. Ontologije nerijetko znaju brojiti i na tisuće a ponekad i na milijune koncepata, pa osim vremena učitavanja ontologije (koje je izrazito problematično) postoje problemi i sa prostorom za pohranjivanje ontologije (memorija).

Konkretno vrijeme učitavanja ontologije je veoma složeno za predvidjeti jer ovisi o brojnim parametrima kao što su:

- broj definiranih koncepata i svojstava,
- broj definiranih ontoloških instanci i relacija (gustoća povezanosti ontologije),
- broj definiranih logičkih izraza,
- korištenje eksternih ontoloških izvora (npr. učitavanje SWRL pod-ontologija), itd.

Da vrijeme učitavanja ontologije nije zanemarivo može se i pokazati vrlo jednostavnim pokusom. Ontologija čije su neke osnovne značajke prikazane na Slici 6.14 učitava se okvirno u pet do sedam sekundi. Pri tome je za učitavanje ontologije korišten alat Protégé-OWL.



Slika 6.14. Neke značajke ontologije koja se u Protégé-OWL alatu učitava okvirno od pet do sedam sekundi.

U praktičnoj uporabi, kada je ontologija u sustavu za potporu pri odlučivanju kompletno izmodelirana, moguće je izmjeriti okvirna vremena učitavanja ontologije te dobivena vremena uvrstiti u procjenu trajanja ukupnog ciklusa izvođenja.

6.4 Pogreška pri rasuđivanju

Kada se rasuđivanje u sustavu provodi pomoću uzorkovanja valja očekivati nekakvu pogrešku pri izračunu traženih vrijednosti. Stoga je broj uzoraka pri rasuđivanju potrebno prilagoditi na način da se maksimalno reducira vrijeme generiranja uzoraka a da se pri tome ostane u području prihvatljive razine učinjene pogreške.

Pogreška koja se učini pri računanju informativnosti grupirane pretrage uzorkovanjem ovisi o:

- broju generiranih uzoraka,
- broju ishoda promatrane grupirane pretrage,
- pojedinačnim očekivanjima ishoda u grupi,
- vrijednostima očekivanjima ciljnog koncepta kada se neki ishod grupirane pretrage već dogodio.

Za očekivati je da se najveća pogreška pri rasuđivanju događa kod najveće postojeće grupirane pretrage (kod pretrage sa najvećim brojem ishoda). Nadalje, najveća pogreška se događa kada su svi definirani ishodi jednako vjerojatni jer se tada skup generiranih uzoraka dijeli ravnomjerno po ishodima grupirane pretrage.

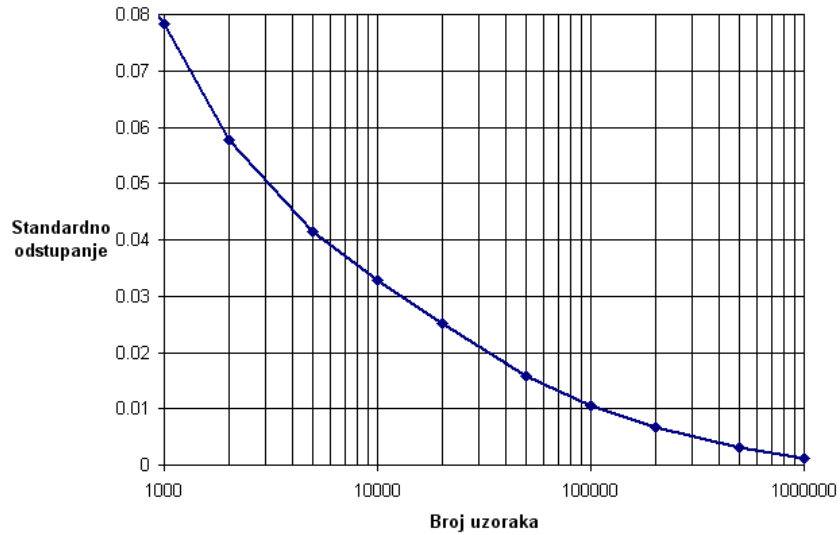
6.4.1 Pogreška i broj uzoraka

Kako bi se izmjerila pogreška pri rasuđivanju potrebno je usporediti izmjerenu vrijednost (dobivenu uzorkovanjem) sa točnom vrijednosti promatranog parametra (izračunata na temelju teorijskog modela). Kada se točne vrijednosti očekivanja i informativnosti mogu izračunati iz teoretskog model moguće je prilikom uzorkovanja izmjeriti točan iznos učinjene pogreške. Međutim, teoretski model računanja informativnosti se ne može upotrijebiti za veće grupirane pretrage (više od 20 elementarnih mjerenja) zbog predugog trajanja izvođenja (poglavlje 4.5.4). Stoga se u ovoj disertaciji uzima da se točna vrijednost informacijske dobiti može dobiti korištenjem ekstremno velikog broja uzoraka (5 milijuna). Na ovaj se način uvodi dodatna pogreška u mjerenje ali je ona zanemarivog iznosa.

Kako je na treći i četvrti parametar naveden u točki 6.4 teško utjecati u nekom realnom modelu baze znanja, prilikom mjerenja pogreške koristi se:

- niz slučajno odabranih ciljnih koncepata u ontologiji te
- niz slučajno generiranih grupiranih pretraga u ontologiji,

te se ponavljanjem pokusa (200 iteracija) mjeri učinjena pogreška prilikom rasuđivanja. Ovaj se postupak ponavlja za različite veličine skupa uzoraka kako bi se mogla vidjeti ovisnost pogreške računanja o veličini skupa uzoraka. Ontologija na kojoj je provedeno mjerenje brojila je ukupno 97 klasa, 278 instanci te 18 svojstava dok je korištena Bayesova mreža sadržavala ukupno 19 čvorova. Najveća grupirana pretraga definirana u ontologiji brojila je 1,327,104 ishoda (*PhysicalExamination*). Najmanja grupirana pretraga brojila je ukupno 4 moguća ishoda (*HeartSounds*).

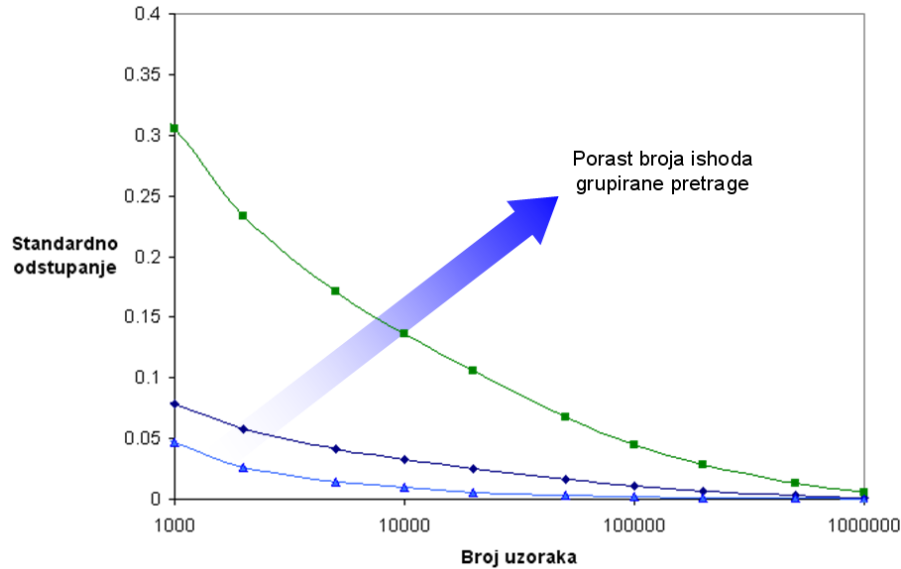


Slika 6.15. Ovisnost prosječne učinjene pogreške o korištenom broju uzoraka.

Slika 6.15 prikazuje ovisnost učinjene pogreške prilikom uzorkovanja o korištenom broju uzoraka prilikom računanja. Na slici se vidi da pri korištenju samo tisuću uzoraka prilikom rasuđivanja standardno odstupanje od točne vrijednosti iznosi do 8% (dakako, to znači da u nekim specifičnim slučajevima iznos pogreške može narasti i na mnogo veći iznos). Stoga se može zaključiti da za odabrane značajke sustava nipošto nije dovoljno koristiti tisuću uzoraka pri rasuđivanju.

Povećanjem broja uzoraka na 100,000 izmjerena pogreška se značajno smanjuje, kada iznosi približno 1%. Daljnjim povećanjem broja uzoraka pogreška se može dodatno smanjiti.

Valja primjetiti da su se u provedenom mjeranju istovremeno koristile pretrage različitih veličina; od manjih (primjerice, pretrage sa ukupno četiri ishoda) do većih (grupe sa preko milijun ishoda). Kada se mjerenje izvrši odvojeno za veće i za manje grupirane pretrage očekuje se veća izmjerena pogreška pri većim grupama te manja izmjerena pogreška pri manjim grupama. Ta je situacija prikazana na slici 6.16. Središnji graf na slici samo je preslika grafa sa slike 6.15; iznad i ispod njega smješteni su grafovi koji prikazuju izmjerenu pogrešku za veće i manje veličine grupa respektivno. Prilikom mjerenja veće su grupe brojile od dvjesto tisuća do milijun i pol ishoda, dok su manje grupe brojile od četiri do deset ishoda.



Slika 6.16. Broj ishoda promatrane grupirane pretrage znatno utječe na učinjenu pogrešku pri rasuđivanju.

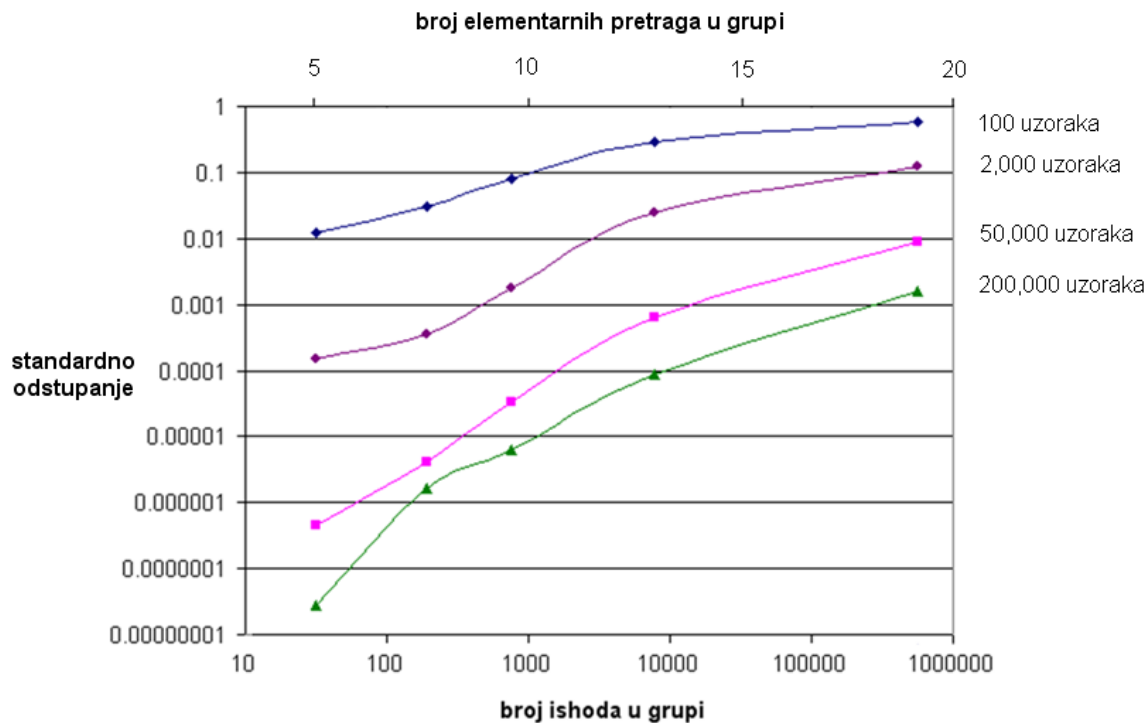
Iako neke pretrage korištene pri mjerenju broje i preko milijun ishoda, generirajući milijun uzoraka (što daje prosječno *samo oko jedan uzorak po ishodu*) učinjena je zanemarivo malena progreska pri rasuđivanju. Dakle, i u praksi se pokazuje da *je često dovoljno čak i manje od jednog uzorka po ishodu grupirane pretrage* da bi se postigla zadovoljavajuća točnost pri rasuđivanju (razlozi ovog efekta opisani su u odjeljku 4.5.4).

Dakako, teoretski se mogu dogoditi i specifične situacije kada ta teza ne vrijedi. Stoga je kod prikaza znanja zasnovanog na integraciji ontologija i Bayesovih mreža poželjno prije samog korištenja sustava provesti potrebna mjerenja kako bi se uspostavila željena razina točnosti.

6.4.2 Pogreška i veličina grupirane pretrage

Najveću poteškoću pri rasuđivanju predstavlja činjenica da broj ishoda grupirane pretrage raste eksponencijalno sa brojem elementarnih pretraga u grupi. Dodavanjem svake sljedeće elementarne pretrage u grupu broj ishoda grupe naraste za (barem) dvostruko što značajno smanjuje točnost pri rasuđivanju.

Slika 6.17 prikazuje kretanje izmjerene pogreške pri rasuđivanju ovisno o veličini grupirane pretrage. Mjerenja su izvršena pri rasuđivanju sa 100 uzoraka, 2,000 uzoraka, 50,000 uzoraka i 200,000 uzoraka. Pri tome se koristila ista ontologija i ista Bayesova mreža kao u prethodnom odjeljku.



Slika 6.17. Ovisnost izmjerene pogreške prilikom rasuđivanja o veličini grupirane pretrage.

Obje veličine na grafu prikazane su u logaritamskoj skali radi preglednosti. S obzirom da je broj ishoda u grupi izravno definiran brojem elementarnih pretraga u grupi, na apscisi je iscrtana (na vrhu grafa) i alternativna skala koja prikazuje broj elementarnih pretraga u grupi (u linearnoj skali). Pri tome je uvedena pretpostavka da svaka elementarna pretraga ima samo dva moguća ishoda.

Iz prikazanog grafa vidljiva je eksponencijalna veza veličine grupirane pretrage i učinjene pogreške pri rasuđivanju uzorkovanjem.

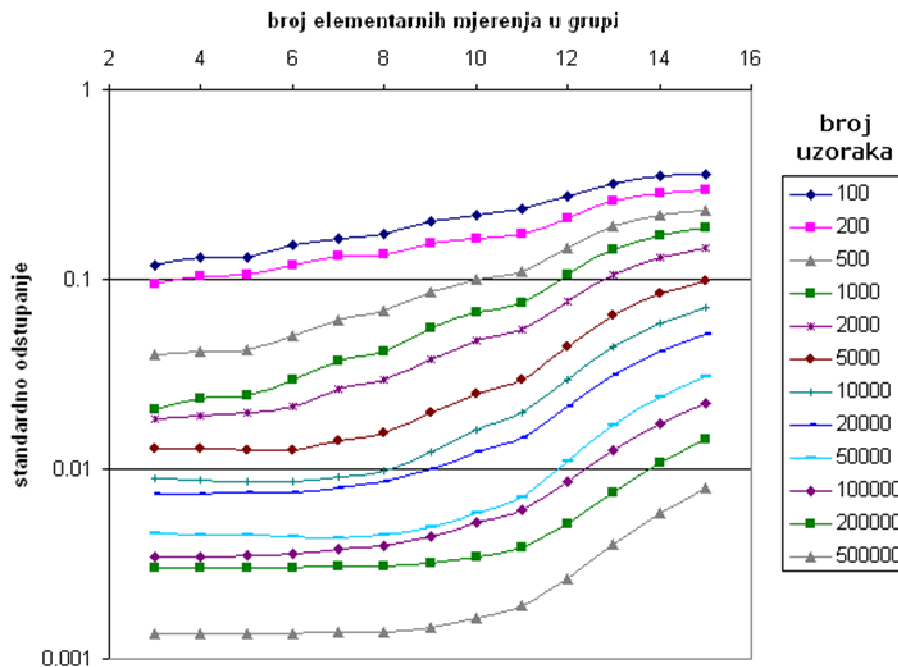
6.4.3 Pogreška pri rasuđivanju uz izbačene ekstremne vrijednosti

Važno je uočiti da se uzorkovanjem mjeri vrijednost *entropije* a tek posredno preko nje i vrijednost *informacijske dobiti*. Pri mjerenju pogreške se za pretrage koje imaju potpunu informacijsku dobit (kada je očekivana entropija jednaka nuli tada je informacijska dobit jednaka 100%) ne učini nikakva pogreška pri računanju informacijske dobiti – neovisno o broju uzoraka postupak uzorkovanja će dati točnu vrijednost informacijske dobiti: 100%. Ovakvi slučajevi značajno ometaju mjerenje na način da osjetno spuštaju izmjereno standardno odstupanje pogreške.

Dobra je vijest da su, iz medicinske perspektive gledano, mnogo „zanimljivije“ upravo pretrage koje imaju visoku informacijsku dobit (nisku entropiju). Naime, liječnik

uspoređuje pretrage sa visokom informacijskom dobiti te odabire jednu ili više njih. Pretrage sa niskom informacijskom dobiti liječniku nisu zanimljive za promatranje. Stoga se može zaključiti da će se veća pogreška učiniti kod medicinski „nezanimljivih“ pretraga a manja kod medicinski „zanimljivih“.

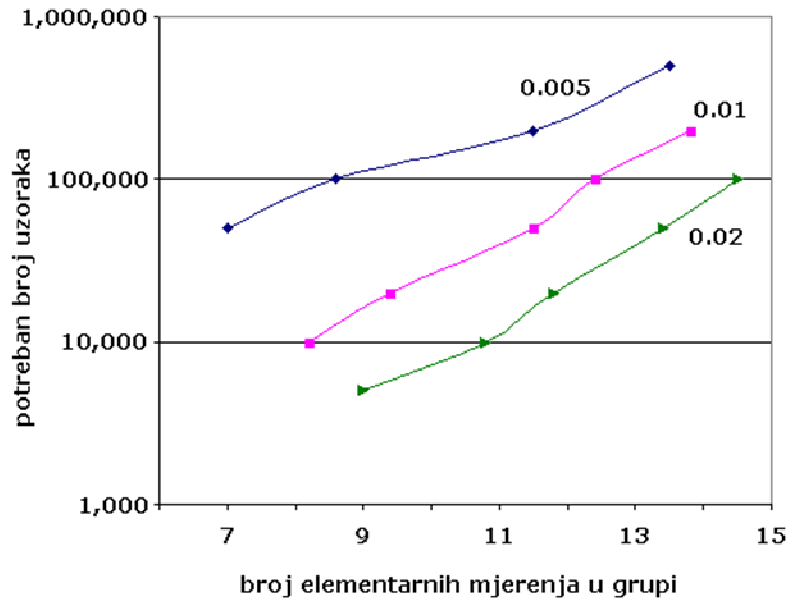
Međutim, kako bi se stekao pravi dojam o učinjenoj pogrešci pri rasuđivanju, korisno je provesti i mjerenje pogreške samo za slučajeve grupiranih pretraga koje nemaju ekstremne vrijednost informacijske dobiti. Stoga se za potrebe mjerenja odaberu grupirane pretrage koje imaju vrijednost informacijske dobiti približno jednaku 50% te se promatra učinjena pogreška pri mjerenju. Rezultati mjerenja prikazani su na Slici 6.18.



Slika 6.18. Pogreška učinjena pri mjerenju za grupirane pretrage kojima je informacijska dobit približno jednaka 50%.

Ako korisnik odabere granicu dozvoljene pogreške od 1% prikazani graf sugerira da je minimalan broj uzoraka koji se mora upotrijebiti 10,000, te da je ovaj broj uzoraka dovoljan samo u slučajevima kada se u ontologiji ne definiraju grupirane pretrage koje imaju više od 8 elementarnih mjerenja. Analogno tome, ako su u ontologiji definirane grupirane pretrage od 15 elementarnih mjerenja nužno je upotrijebiti barem 500,000 uzoraka pri rasuđivanju.

Gledan iz druge perspektive graf sa slike 6.18. može se doživjeti i na drugačiji način. Naime, pomoću istog grafa moguće je odgonetnuti koji je minimalan broj uzoraka koji se mora upotrijebiti pri rasuđivanju kako bi se postigla zadovoljavajuća razina pogreške. Ova je ovisnost jasnije prikazana grafom na slici 6.19.



Slika 6.19. Potreban broj uzoraka s obzirom na veličine definiranih grupiranih pretraga u ontologiji i na odabranu razinu pogreške.

Primjerice, graf sugerira da je uz maksimalnu veličinu grupe od 14 elementarnih mjerenja i dozvoljenu pogrešku od 2% potrebno upotrijebiti minimalno 100,000 uzoraka.

6.5 Poboljšanje performansi

Iz rezultata dobivenih mjerenjem može se zaključiti da se pri izgradnji baze znanja nikako ne smije pretjerivati sa veličinom grupirane pretrage. Dakako, u stvarnom su svijetu grupirane pretrage unaprijed definirane samom domenom koja se promatra pa se stoga na njihovu veličinu ne može utjecati. Ipak, u nekim se slučajevima može unaprijed predmnijevati koja elementarna pretraga u grupi nosi manje važnu informaciju te je prilikom rasuđivanja izbaciti iz grupe – na taj se način vrlo efikasno smanjuje broj ishoda grupe što povećava točnost mjerenja ili (alternativno) smanjuje vrijeme izvođenja.

Efikasnost postupka računanja se može dodatno povećati uvođenjem paralelizacije u sustav. U postupku rasuđivanja ne postoje nikakve prepreke koje bi onemogućile da se generiranje uzoraka provede distribuirano na više računala istovremeno. Time bi se vrlo jednostavnim zahvatima značajno ubrzao postupak rasuđivanja.

7 Zaključak

U disertaciji je opisan formalizam za prikazivanje znanja koji se zasniva na integraciji ontologija i Bayesovih mreža. Uporaba formalizma je ilustrirana kroz primjer sustava za potporu pri odlučivanju za dijagnosticiranje srčanog zatajenja.

U integriranom sustavu ontologije doprinose na način da:

- omogućavaju sistematičnu organizaciju znanja (opis domene),
- pružaju mogućnost opisa znanja pomoću deskriptivne logike,
- daju objašnjenja (interpretacije) donešenih zaključaka.

Bayesove mreže doprinose integriranom sustavu na način da:

- pronalaze vjerojatnosti definiranih ciljnih koncepata,
- pronalaze informacijsku dobit pojedinih grupiranih koncepata definiranih u ontologiji.

Integrirani sustav je ostvaren na sljedeći način:

- definiran je postupak integriranja ciljnih koncepata iz ontologije u Bayesovu mrežu,
- definiran je postupak računanja informacijske dobiti grupiranih koncepata definiranih u ontologiji.

Postupak stvaranja ontologije prikazao je kako je formalizam deskriptivne logike uistinu prikladan za opis medicinskog znanja, u prvom redu za opis strukture i odnosa medicinskih koncepata (primjerice, za opis strukture medicinskih pretraga) a jedanko tako i za opis ekspertnog znanja u obliku pravila.

Kroz nekoliko primjera prikazano je i kako u mnogim slučajevima ontološka struktura nije dostatna za jedan cjelovit ekspertni sustav – primjerice, izostanak bilo kakvog zaključka pri nedostatku potpune informacije o pacijentu. Uvođenjem Bayesove mreže u sustav omogućen je prelazak iz egzaktnog oblika zaključivanja u zaključivanje sa vjerojatnostima što u mnogim slučajevima daje dodatnu dimenziju sustava za potporu pri odlučivanju. Time je sustav zadržao svoje osnovne karakteristike temeljene na logici ali je i dodatno proširen sa vjerojatnostima za slučajeve kada slika pacijenta u sustavu nije savršeno jasna.

Zanimljivo je primjetiti da ni korištenje deskriptivne logike u sustavu nije sasvim neophodno. Naime, moguće je promatrati ontologiju samo kao alat za definiciju

strukturiranih dijagnostičkih pretraga (bez korištenja logičkih izraza) te ispitivati informacijsku dobit pretraga s obzirom na neke već postojeće ciljne čvorove u Bayesovoj mreži. U takvom okruženju je stoga dovoljno naučiti Bayesovu mrežu iz podataka i definirati strukturu dijagnostičkih testova u ontologiji. Ovo svojstvo sustava je korisno kada ne postoji ekspertno znanje koje se može formalizirati u obliku pravila.

Pri izgradnji Bayesove mreže razmatrana su dva moguća pristupa: ručna izgradnja i izgradnja strojnim učenjem na temelju skupa podataka. Ručna izgradnja Bayesove mreže i inače slovi kao iznimo težak problem, u prvom redu zahvaljujući tablicama uvjetnih vjerojatnosti u kojima je potrebno definirati velik broj numeričkih vrijednosti. S druge strane, procesom strojnog učenja se uglavnom postižu strukture mreže koje su neintuitivne i čija je medicinska interpretacija teška. Stoga se čini najzgodnijim upravo kompromisno rješenje: ručna izgradnja strukture mreže i učenje tablica uvjetnih vjerojatnosti iz skupa podataka. Takav pristup je primjenjen i u okviru ove doktorske disertacije.

Opisana metodologija računanja informacijske dobiti pokazala je kako je zahvaljujući uporabi strukturiranih (grupiranih) dijagnostičkih pretraga složenost rasuđivanja drastično porasla u odnosu na klasične oblike rasuđivanja kod Bayesovih mreža. Ovaj problem smo pokušali riješiti uporabom uzorkovanja pri rasuđivanju. Naravno, na taj se je način samo postigao svojevrstan kompromis jer se uštedjelo na vremenu rasuđivanja uz gubitke u točnosti rasuđivanja.

Mjerenja pogreške učinjene pri rasuđivanju i mjerenja vremena potrebnog za rasuđivanje koja su provedena u okviru ove disertacije prikazuju da je predložena metodologija uz određene pretpostavke upotrebljiva u praktičnim primjenama. Ograničenja u uporabi metodologije odnose se uglavnom na veličinu korištenih grupiranih pretraga – kada su definirane veličine grupa potrebno je pronaći zadovoljavajući odnos vremenskog trajanja rasuđivanja i pogreške koja se učini pri rasuđivanju. Navedeni odnos uvelike ovisi i o konkretnim karakteristikama ontologije i konkretnim karakteristikama Bayesove mreže te se svakako preporuča pri izgradnji baze znanja svaki put ponovno provesti opisana mjerenja.

Kako bi sustav za potporu pri odlučivanju zaživio nužno je ostvariti sučelje baze znanja prema okolini. U okvirima medicinskog sustava za potporu pri odlučivanju to se odnosi u najvećoj mjeri na medicinske baze podataka. Stoga je u okviru disertacije predložen formalizam koji ima mogućnost čitanja podataka iz medicinskih baza podataka i transformacije u oblik koji je prikladan za rasuđivanje – ontološki oblik. Opisani formalizam preslikavanja „činjeničnog znanja“ se u korištenom primjeru sustava za potporu pri odlučivanju pokazao kao vrlo prikladan, iako se još uvijek mogu zamisliti složene transformacije koje nisu obuhvaćene predloženom sintaksom. Međutim, dojam je da je mnoge složene transformacije podataka potrebno obaviti zasebno u fazi *pripreme podataka* kako bi se baza znanja i sam ekspertni sustav rasteretio specifičnih i složenih

proračuna a više se koncentrirano na samo ekspertno znanje u svom osnovnom konceptualnom obliku.

Predložena metodologija još uvijek mora pokazati svoju praktičnu korisnost u stvarnim medicinskim okruženjima sa stvarnim pacijentima i stvarnim liječnicima. Za ovo će biti potrebno nešto dulje vrijeme te se može smatrati tek dugoročnim ciljem. Nadalje, prikladnost metodologije u nekim drugim domenama (osim medicine) se također još mora pokazati. Ovi se problemi čine kao zanimljiva tematika za budući rad.

Konačno, opisana metodologija predstavlja i svojevrstan pokušaj integracije deklarativnog znanja (opisanog ontologijom) i induktivnog znanja (dobivenog u procesu strojnog učenja u Bayesovoj mreži). Na taj način je u bazu znanja uključeno i eksplicitno ekspertno znanje sadržano u medicinskim smjernicama (ili dobiveno razgovorima s ekspertom) kao i implicitno znanje koje se inherentno nalazi u sakupljenim medicinskim podacima. Ovako koncipirana baza znanja stoga rezultira i sveobuhvatijim sustavom za potporu pri odlučivanju.

Literatura

- [1] Achour, S.L., Dojat, M., Rieux, C., Bierling, P., Lepage, E., (2001). *A UMLS-based Knowledge Acquisition Tool for Rule-based Clinical Decision Support System Development*, J Am Med Inform Assoc. (2001), 8:351-360
- [2] An, Y., Borgida, A., Mylopoulos, J., (2004). *Refining the Semantic Mappings from Relational Tables to Ontologies*. In Proceedings of 2nd International workshop on Semantic Web and Databases (SWDB'04) in conjunction with Very Large Databases (VLDB'04), Toronto, Canada, August 29-30, 2004, p. 84-90.
- [3] Baader, F., Nutt, W. (2002). *Basic Description Logics (Description Logic Handbook)*. Edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002, p. 47-100.
- [4] Baron, R.J., Fabens, E.L., Schiffman, M., Wolf, E., (2005). *Electronic Health Records: Just around the Corner? Or over the Cliff?* Annals of Internal Medicine, vol 143, issue 3, p. 222-226.
- [5] Barrasa, J., Corcho, O., Gomez-Perez, A., (2003). *Fund Finder Wrapper: A case study of database-to-ontology mapping*. Semantic Integration Workshop SI-2003, Sanibel Island, Florida, October 20, 2003, p. 9-15.
- [6] Barrasa, J., Gomez-Perez, A., (2006). *Upgrading Relational Legacy Data to the Semantic Web*. Proceedings of the 15th International Conference on World Wide Web WWW '06, Edinburgh, Scotland, May 23 - 26, 2006, p. 1069-1070.
- [7] Baumeister, J., Kleemann, T., Seipel, D., (2007). *Towards the Verification of Ontologies with Rules*. Proc. of the 20th International Florida Artificial Intelligence Research Society Conference FLAIRS-2007, Key West, Florida, May 7-9, 2007, p. 524-529.
- [8] Bradley, J.H., Paul, R., Seeman, E., (2006). *Analyzing the structure of expert knowledge*. Information and Management (2006), 43:77-91
- [9] Ceusters, W., Smith, B., (2003). *Ontology and Medical Terminology: why Descriptions Logics are not enough*. Proceedings of the conference Towards an Electronic Patient Record (TEPR 2003), San Antonio, May 10-14, 2003 (elektronska publikacija).
- [10] Charniak, E., (1991). *Bayesian networks without tears*. Artificial Intelligence Magazine, 12(4): 50-63.

- [11] Cleland, J., Dargie, H. et al. (2005). *Guidelines for the diagnosis and treatment of chronic heart failure: executive summary (update 2005): The Task Force for the Diagnosis and Treatment of Chronic Heart Failure of the European Society of Cardiology*. European Heart Journal 26: 1115 - 1140.
- [12] deClercq, A., Blom, J.A., Korsten, H.M., Hasman, A. (2004). *Approaches for creating computer-interpretable guidelines that facilitate decision support*. Artificial Intelligence in Medicine, 31(1):1-27.
- [13] Devitt, A., Danev, B., Matusikova, K. (2006). *Constructing Bayesian Networks Automatically using Ontologies*. Formal Ontology Meet Industry workshop 2006, Trento, Italy, December 14-15, 2006.
- [14] Ding, Z., Peng, Y. (2004). *A Probabilistic Extension to Ontology Language OWL*. 37th Hawaii Int. Conf. on System Sciences HICSS-37, Hilton Waikoloa Village, Big Island, Hawaii, 7-10 January, 2002.
- [15] Eiter, T., Ianni, G., Schindlauer, R., Tompits, H., Wang, K. (2006). *Forgetting in Managing Rules and Ontologies*. International Conference on Web Intelligence WI2006, Hong Kong, China, December 18-22, 2006, p. 411-419.
- [16] Enrico, C., (2003). *Clinical Decision Support Systems* (knjiga, 2nd revised edition). Hodder Arnold publishing.
- [17] Frenster, J.H., (1989). *Expert Systems and Open Systems in Medical Artificial Intelligence*. Medical Systems and Informatics, 7:118–120.
- [18] Gamberger, D., Prcela, M., Bošnjak, M., (2008). *Attribute Ranking for Intelligent Data Analysis in Medical Applications*. Proceedings of the ITI 2008 30th International Conference on Information Technology Interfaces 2008, Cavtat, Croatia, June 23-26, 2008, p. 323-328.
- [19] Gamberger, D., Prcela, M., Jović, A., Šmuc, T., et. al. (2008). *Medical knowledge representation within Heartfaid platform*. Health Informatics Conference (BIOSTEC-HEALTHINF 2008), January 28-31, 2008, p. 307-314.
- [20] Goasdoue, F., Lattes, V., Rousset, M., (2000). *The Use of CARIN Language and Algorithms for Information Integration: The PICSEL Project*. International Journal of Cooperative Information Systems, 9(4):383-401.
- [21] Grimm, S., Motik, B. (2006). *Closed World Reasoning in the Semantic Web through Epistemic Operators*. Workshop on OWL: Experiences and Directions (OWLED 2005), Galway, Ireland, November 11–12 2005.
- [22] Pryor, T.A., Hripcsak, G., (1993). *The arden syntax for medical logic modules*. International Journal of Clinical Monitoring and Computing, 10:215-224.

- [23] Huhns, M.N., Valtorta, M.G. (2007). *Ontological Support for Bayesian Evidence Management*. Ontology for the Intelligence Community Workshop 2007, November 28-29, 2007, Columbia MD, USA, p. 47-51.
- [24] Jagt, R.M., (2002). *Support for multiple cause diagnosis with Bayesian Networks*. Thesis for degree of Master of Science in Applied Mathematics. Information Sciences department, University of Pittsburgh, USA.
- [25] Jenders, R.A., Huang, H., Hripcsak, G., Clayton, P.D., (1998). *Evolution of a Knowledge Base for a Clinical Decision Support System Encoded in the Arden Syntax*. AMIA Symposium, Lake Buena Vista, Florida, USA, November 7-11, 1998, p. 558-562.
- [26] Jeon, B.J., Ko, I.Y., (2007). *Ontology-Based Semi-automatic Construction of Bayesian Network Models for Diagnosing Diseases in E-health Applications*. Frontiers in the Convergence of Bioscience and Information Technologies Fbit 2007, Ramada Plaza Jeju, Jeju Island, Korea, October 11-13, 2007, pp.595-602
- [27] Jović, A., Prcela, M., Gamberger, D., (2007). *Ontologies in Medical Knowledge Representation*. Proceedings of the ITI 2007 29th International Conference on Information Technology Interfaces 2007, Cavtat, Croatia, 25-28 June, 2007, p. 535-540.
- [28] Jović, A., Prcela, M., Krstačić, G., (2007). *Medical Plans as a Middle Step in Building Heart Failure Expert System*. Xi Meditarranean Conference on Medical and Biological Engineering and Computing MEDICON 2007, Ljubljana, Slovenia, June 26-30, 2007, p. 549-553.
- [29] Kawaguchi, A., Motoda, H., Mizoguchi, R., (1991). *Interview-Based Knowledge Acquisition Using Dynamic Analysis*. IEEE Expert: Intelligent Systems and Their Applications Vol. 6, 5:47-61.
- [30] Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A., (2004). *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. International Semantic Web Conference 2004, Hiroshima, Japan, November 7-11, p. 229-243.
- [31] Linder, J.A., Ma, J., Bates, D.W., Middleton, B., Stafford, R.S., (2007). *Electronic Health Record Use and the Quality of Ambulatory Care in the United States*. Archives of Internal Medicine 167(13):1400-1405.
- [32] Lukasiewicz, T., (2008). *Expressive Probabilistic Description Logics*. Artificial Intelligence Journal, 172 (2008) 852-883.
- [33] McGarry, K., Garfield, S., Wermter, S., (2007). *Auto-Extraction, Representation and Integration of a diabetes Ontology using Bayesian Network*. Computer-Based

- Medical Systems Symposium 2007, Maribor, Slovenia, June 20-22, 2007, p. 612-617.
- [34] Mena, E., Kashyap, V., Sheth, A., Illarramendi, A., (2000). *OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies*. International Journal on Distributed And Parallel Databases (DAPD) (2):223-272.
- [35] Metaxiotis, K.S., Samouilidis, J.E., (2000). *Expert Systems in Medicine: Academic Illusion or Real Power?* Information Management and Computer Security Journal, 8:75–79.
- [36] Mizoguchi, R., Matsuda, K., Nomura, Y., (1990). *ISAK: Interview system for acquiring design knowledge - A new architecture of interview systems using examples.*, Proceedings of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop (JKAW) (1990), Ohmusha, Japan, October 25-31, 1990, p. 277-286.
- [37] Neapolitan, R.E. (2005). *Learning Bayesian Networks* (book). Publisher: Prentice Hall. Cleland, J., Dargie, H. et al. (2005).
- [38] Noy, N.F., Rubin, D.L., Musen, M.A., (2004). *Making Biomedical Ontologies and Ontology Repositories Work*. IEEE Intelligent Systems Journal, 19(6), 78-81.
- [39] Pan, R., Ding, Z., Yu, Y., Peng, Y., (2005). *A Bayesian Network Approach to Ontology Mapping*. Fourth International Semantic Web Conference 2005, Galway, Ireland, November 6-10, 2005, p. 563-577.
- [40] Peleg, M., Gutnik, L.A., Snow, V., Patel, V.L., (2006). *Interpreting procedures from descriptive guidelines*. Journal of Biomedical Informatics, 39(2), 184-195.
- [41] Prcela, M., Gamberger, D., Bogunović, N., (2007). *Developing Factual Knowledge from Medical Data by Composing Ontology Structures*. Proceedings of MIPRO 2007, Opatija, Croatia, May 21-25, 2007, Vol. III. p. 145-150.
- [42] Prcela, M., Gamberger, D., Jović, A., (2008). *Semantic Web Ontology Utilization for Heart Failure Expert System Design*. Proceedings of MIE2008, The XXIst International Congress of the European Federation for Medical Informatics, Goteborg, Sweden, May 25-28, 2008, p. 851-856.
- [43] Prcela, M., Gamberger, D., Šmuc, T., Bogunović, N., (2009). *Information Gain of Structured Medical Diagnostic Tests*. International Conference on Health Informatics HEALTHINF 2010 (Rad prihvaćen za tisak, Valencia, Spain, January 19-24, 2010).
- [44] Rubin, D.L., Dameron, O., Bashir, Y., Grossman, D., Dev, P., Musen, M.A., (2006). *Using ontologies linked with geometric models to reason about*

- penetrating injuries*. Artificial Intelligence in Medicine Journal (AIIM), 37:167 - 176.
- [45] Schmidt-Schauss, M., Smolka, G., (1991). *Attributive concept descriptions with complements*. Artificial Intelligence Journal, 48(1):1-26.
- [46] Shahr, Y., Miksch, S., Johnson, P., (1998). *The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines*. Artificial Intelligence in Medicine Journal (AIIM), 14(1-2), 29–51.
- [47] Wyatt, J., Spiegelhalter, D., (1990). *Evaluating medical expert systems: what to test and how?* Informatics for Health and Social Care Journal, 15:205-217.
- [48] Stojanović, N., Stojanović, Lj., Volz, R., (2002). *A reverse engineering approach for migrating data-intensive web sites to the Semantic Web*. Intelligent Information Processing Workshop 2002 (IIP-2002, Part of the IFIP World Computer Congress WCC2002), Montreal, Canada, August, 25-30, 2002, p. 141-154.
- [49] Sutton, D.R., Fox, J., (2003). *The Syntax and Semantics of the PROforma guideline modelling language*. Journal of the American Medical Informatics Association: JAMIA, 2003 10(5):433–43.
- [50] Svab, O., Svatek, V., (2006). *Combining Ontology Mapping Methods Using Bayesian Networks* (poster). International Workshop on Ontology Matching, Athens, Georgia, USA, November 5, 2006, volume 225.
- [51] Tierney, W.M., Overhage, J.M., Takesue, B.Y., Harris, L.E., Murray, M.D., Vargo, D.L., McDonald, C.J., (1995). *Computerizing guidelines to improve care and patient outcomes: the example of heart failure*. Journal of the American Medical Informatics Association, 2(5):316-322.
- [52] Town, C., (2004). *Ontology-driven Bayesian Networks for Dynamic Scene Understanding*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop 2004, Washington D.C., USA, June 27 – July 02, 2004.
- [53] Wang, W., Zeng, G., Zhang, D., Huang, Y., et al. (2008). *An intelligent Ontology and Bayesian Network based Semantic Mashup for Tourism*. IEEE Congress on Services, part I, Hawaii, USA, July 6-11, 2008, p. 128-135.
- [54] Zheng, H.T., Kang, B.Y., Kim, H.G., (2005). *An Ontology-based Bayesian Network Approach for Representing Uncertainty in Clinical Practice Guidelines*. Uncertainty Reasoning for the Semantic Web workshop, Galway, Ireland, November 7, 2005. p. 161-173.

- [55] HEARTFAID projekt: <http://heartfaid.org/>²
- [56] Genie&Smile – alati za Bayesove mreže: <http://genie.sis.pitt.edu/>
- [57] Protégé – alat za uređivanje ontologija: <http://protege.stanford.edu/>
- [58] Tehnike optimizacije tableau algoritma: <http://dl.kr.org/dig/optimisations.html>
- [59] Pregled alata za uređivanje Bayesovih mreža:
<http://people.cs.ubc.ca/~murphyk/Bayes/bnsoft.html>
- [60] Hugin – alat za uređivanje Bayesovih mreža: <http://www.hugin.com/>

² Sve interesantne stranice su na dan 25. studenog 2009 dostupne na navedenim adresama.

Oznake

U tekstu su korištena sljedeća pravila označavanja:

	Označavanje	Primjeri uporabe
Koncepti u deskriptivnoj logici	Tiskana slova uz veliko početno slovo	OSOBA, KOLEGIJ, HFSIGN
Relacije u deskriptivnoj logici	Tiskana slova uz malo početno slovo	PREDAJE, SLUŠA, HASCHARACTERISTIC
Individue u deskriptivnoj logici	Kurziv, malo početno slovo uz iznimku kratica	<i>pero, heartRate, EAratio, LVEFhigh, dyspnea</i>
Imena čvorova u Bayesovoj mreži	Tiskana slova uz veliko početno slovo	HEART RATE, LVEF, DYSPNEA
Imena ishoda u Bayesovoj mreži	Tiskana slova uz malo početno slovo	YES, NO, HIGH, LOW, NORMAL

Predstavljanje znanja zasnovano na integraciji ontologija i Bayesovih mreža

sažetak

Ontologije temeljene na deskriptivnoj logici imaju semantiku koja je dovoljno izražajna za formalizaciju opisnog znanja (opis domene, terminologija) a također u jednom dijelu i proceduralnog znanja (akcije koje se izvršavaju). Baza znanja formalizirana na taj način osnova je modernih medicinskih sustava za potporu pri odlučivanju, koji se u trenutku korištenja oslanjaju na eksterne izvore podataka o pacijentima – najčešće iz nekih elektronskih registara pacijenata (medicinskih baza podataka). U slučajevima kada dio informacije potreban za donošenje zaključaka nedostaje sustav nije u mogućnosti donositi konkretne odluke. U takvim slučajevima uvođenjem Bayesovih mreža u sustav mogu se donositi zaključci temeljeni na nepotpunoj informaciji: koji su najvjerojatniji zaključci i koje su najinformativnije akcije koje vode ka konkretnim zaključcima. U doktorskoj disertaciji se kroz primjer dijagnoze srčanog zatajenja predlaže metodologija integracije dva već provjerena oblika predstavljanja znanja: ontologija i Bayesovih mreža. Pri tome se analiziraju oblici uporabe ovako integriranog sustava te se definiraju postupci rasuđivanja u integriranom sustavu. Nadalje, opisuje se formalizam koji omogućuje uključivanje realnih medicinskih podataka promatranog pacijenta ovako oblikovanoj bazi znanja. Na koncu se daje uvid u korisničko sučelje sustava te se analiziraju neke funkcionalne značajke sustava kao što su vrijeme izvođenja i pogreška učinjena pri rasuđivanju.

ključne riječi

predstavljanje znanja, ontologije, Bayesove mreže, deskriptivna logika, integracija, informacijska dobit, sustavi za potporu pri odlučivanju, ekspertni sustavi

Knowledge representation based on integration of ontologies and Bayesian networks

abstract

Ontologies based on description logic have expressive enough semantics to cover the descriptive part of knowledge base (domain description, terminology), and also a large part of procedural knowledge in the knowledge base (actions to be taken). Such encoded knowledge is the basis of modern medical decision support systems, which at the time of use also have rely on external patient data – typically from some electronic health record. In cases when some amount of information is missing the system often has to refrain to draw any concrete conclusion. In such cases by introducing Bayesian networks into the system conclusions can be made based on uncertain knowledge: what are the most probable conclusions and what are the most informative actions that possibly lead to concrete conclusions. Through the example of diagnosis of heart failure disease this dissertation demonstrates methodology for integration of two well known forms of knowledge representation: ontologies and Bayesian networks. Features of such integrated system are analyzed and reasoning procedures are defined. Furthermore, a formalism that includes data of observed patients into reasoning process is described. Finally, user interface of the example decision support system is described and some functional properties of the system are analyzed such as duration of reasoning and error made in reasoning.

keywords

knowledge representation, ontologies, Bayesian networks, description logics, integration, information gain, decision support system, expert system

Životopis

Rođen sam u Slavonskom Brodu 1. travnja 1981. godine gdje sam završio osnovnu i srednju školu. Godine 1999. upisao sam Fakultet elektrotehnike i računarstva u Zagrebu gdje sam i diplomirao 2004. godine na smjeru računarstvo.

Po završetku studija i odsluženja vojnog roka zapošljava se tvrtci „Sky MobileMedia“ u Zagrebu na razvoju programske podrške za mobilne uređaje.

Ubrzo, već početkom 2006. godine, zapošljava se na Institutu „Ruđer Bošković“ kao stručni suradnik na Zavodu za elektroniku u Laboratoriju za informacijske sustave na europskom FP6 projektu HEARTFAID. U sklopu ovog projekta bavim se razvojem računalne baze znanja koja je temelj medicinskom sustavu za potporu pri odlučivanju pri dijagnozi i terapiji srčanih bolesnika. Istovremeno upisujem i poslijediplomski doktorski studij računarstva na Fakultetu elektrotehnike i računarstva.

Od jeseni 2008. godine zaposlen sam i kao asistent na Prirodoslovno-matematičkom fakultetu u sklopu predmeta „Umjetna inteligencija“ na Diplomskom sveučilišnom studiju Računarstva i matematike.

Od 2009. godine sudjelujem i na BICRO projektu pod nazivom „Razvoj modula za elektroničku razmjenu dokumenata i informacija u uredskom poslovanju kao nadogradnja poslovnom rješenju CENTRIX“. Na projektu se bavim organizacijom sintakse i semantike meta-podataka pri razmjeni poruka i informacija u uredskom poslovanju.

Također u 2009. godini postajem i znanstveni novak u sklopu projekta „Algoritmi strojnog učenja i njihova primjena“ pod vodstvom dr. Dragana Gambergera na Institutu „Ruđer Bošković“.

Biography

I was born on April 1, 1981, in town of Slavonski Brod, Croatia, where I attended primary school and high school. In period 1999-2004 I attended Faculty of Electrical Engineering and Computing in Zagreb, Croatia where I graduated in the field of Computing.

After graduation and military service I was employed in "Sky MobileMedia" company as software developer for mobile devices.

Very soon, in the beginning of 2006, I started to work on "Rudjer Bošković" Institute in Division of Electronics, Laboratory for Information Systems, as an associate on european FP6 project HEARTFAID. Within this project I was involved in development of knowledge base which is a basis of medical decision support system for diagnosis and therapy of patients with heart failure disease. At the same time I enrolled at the postgraduate study of computing at the Faculty of Electrical Engineering and Computing in Zagreb.

Since autumn 2008 I am also employed as teaching assistant on Faculty of Science in Zagreb, on "Artificial intelligence" course within Graduate study of Computing and Mathematics.

Since January 2009 I am working on BICRO project named "Development of modules for electronic interchange of documents and information as an upgrade to business solution CENTRIX". The main orientation of the project is organization of syntax and semantics of meta-data for interchange of documents and information in electronic records management.

In May 2009 I became a junior researcher on project "Machine Learning Algorithms and their Application" led by dr. Dragan Gamberger also on "Ruđer Bošković" Institute.