

DEFAULT-LOOGIKA JA SELLE RAKENDAMISE  
VÕIMALUSI DIALOOGSÜSTEEMIDES

Tanel Tammet

1. Sissejuhatus

Viimase kümne aasta jooksul on tehisintellektialaste uuringute raames oluliselt suurenenud loogikauuringute osakaal. On saanud üha selgemaks, et loogika puuduste kõrval on olemas sellised eelised, et alternatiivsed andmebaaside struktuurid (näiteks freimid, semantilised võrgud, lihtsad produktsioonisüsteemid) ei anna meile üldjuhul nii üldisi ja võimsaid väljendamis- ja järeldamisvahendeid kui loogika kasutamine. Loogika eelisteks on infoühikute (so loogikavalemitena) iseseisvus, üldisus ja arusaadavus. Erinevalt protseduraalsest teadmiste esitamise viisist on loogikavalemitena esitatud andmebaas lihtsalt modifitseeritav, mis lihtsustab tunduvalt näiteks selle kasutamist iseõppivates süsteemides, ning on põhimõtteliselt sõltumatu konkreetsest järelduste tuletamise mehhanismist.

Nagu ükskõik milline formalism, ei ole ka loogika ilmselt piisav igasuguste teadmiste efektiivseks esitamiseks. Traditsioonilise loogika (esimest järku predikaatarvutuse) mitmesugused puudused ongi põhjuseks, miks peale loogika kasutatakse tehisintellektisüsteemides veel kõikvõimalikke muid teadmiste formaliseerimise meetodeid. Soov nendest puudustest üle saada traditsioonilise loogika rikastamise teel ning teiselt poolt soov loogikas formaliseerida olemasolevaid, aga mitte otseselt loogikal baseeruvaid süsteeme on viinud intensiivsele uurimistööle väga mitmesuguste mitte-traditsiooniliste loogikate alal.

Üks rakenduste mõttes eriti perspektiivne klass loogikaid on mittemonotoonsed loogikad. Monotoonsus tähendab, et teadmiste lisamisel andmebaasi kunagi ei kahane faktide hulk, mida on võimalik sellest andmebaasist järeldada. Traditsioonilised loogikad on alati monotoonsed. Mittemonotoonsus tähendab, et mingi uue teadmise lisamine võib blokeerida mõned varem võimalikud järeldused. Järeldatavate faktide hulk võib seega kahaneda.

Harilik inimõtlemine on üldiselt mittemonotoonne. Kui meile näiteks räägitakse mingist linnust, siis järeldame

automaatselt, et tal on tiivad ja ta oskab lennata, ning enamasti see ongi tõsi. Kui me järgnevalt saame aga kuulda, et tegemist on jaanalinnuga või pingviiniga, siis me enam ei järelda, et ta lennata oskab.

Mittemonotoonse mõtlemise formaliseerimiseks on pakutud väga mitmesuguseid loogikaid. Perspektiivsemad neist näivad J. McCarthy poolt pakutud circumscription-tehnika(5), R. Reiteri default-loogika (8), D. McDermotti ja J. Doyle'i mitte-monotoonne modaalne loogika (6) ja mõned meetodid, mis kasutavad metateadmisi. Käesolevas käsitleme neist ainult default-loogikat.

## 2. Default-loogika.<sup>1</sup>

R. Reiter defineerib default-teooria kui esimest järku predikaatarvutuse, mida on täiendatud default-reeglitega kujul:

$$\frac{A(x_1, \dots, x_n) : B(x_1, \dots, x_n)}{C(x_1, \dots, x_n)},$$

mis tähendab, et kui on täidetud tingimus A ning kui B ei ole meie teadmistega vastuolus (st ei saa tuletada  $\neg B$ -d), siis võime järeldada C (A, B ja C on mingid valemid). Näiteks

$$\frac{\text{Lind}(x) : \text{Lennuvõimeline}(x)}{\text{Lennuvõimeline}(x)}$$

tähendab, et kui me ei tea (st ei saa kusagilt tuletada), et konkreetne lind ei ole lennuvõimeline, siis järeldame, et ta on lennuvõimeline. Oletame, et on teada:

Varblane (Sass),

Jaanalind (Jaanika),

Varblane (x)  $\rightarrow$  Lind (x),

Jaanalind (x)  $\rightarrow$  Lind (x),

Jaanalind (x)  $\rightarrow \neg$ Lennuvõimeline (x).

Sellest teooriast (st andmebaasist) järeldub, et Sass on lennuvõimeline (arvestades eelpool toodud default-i),

<sup>1</sup>Ingliseelsele terminile default ei ole seni leitud sobivat eestikeelset vastet. On küll pakutud mitmeid variante, nagu kaudloogika, vaikimisloogika jms., kuid need kannavad endas ebasoovitavaid tähendusvarjundeid, mida inglise terminil default ei ole. Seetõttu on peetud paremaks jätta default õnnestunuma eestikeelse vaste leidmiseni tõlkimata.

kuid Jaanika ei ole, kuigi nad mõlemad on linnud. Arusaadav, et tegelikkuses võib olla 1001 põhjust, miks üks lind lennata ei saa. Kui me tahaksime "lendamise teooria" formaliseerida traditsioonilises predikaatarvutuses, peaksime kõik need võimalused lendamise reeglis üles lugema (mis on ilmselt võimatu, sest alati võib ette tulla olukordi, mida me ei suuda ette näha), ja tõestamise käigus üle kontrollima (mis oleks kohutav töö).

## 2.1. Default-loogika puudused.

R. Reiter näitab (8), et üldised default-teooriad käituvad äärmiselt halvasti. Default-teooria teoreemide (st võimalike järelduste) hulk ei ole rekursiivselt loetletav, st ei saa olla algoritmi, mis põhimõtteliselt suudaks alati tõestada tegelikult tõestatavad teoreemid (traditsioonilise predikaatarvutuse jaoks on sellised algoritmid olemas). Võib konstrueerida default-teooriaid, millel polegi teoreeme. Isegi teoreemide hulga ühesus ei ole iga teooria jaoks garanteeritud: mõnest teooriast on võimalik järeldada erinevaid, üksteisele vasturääkivaid teoreemide hulki.

## 2.2. Normaalsed ja poolnormaalsed teooriad.

Halvast üldjuhust on õnneks võimalik välja eraldada tunduvalt paremate omadustega teooriate klasse. Eriti kasulikud on nn normaalsed ja poolnormaalsed default-teooriad. Normaalseks nimetatakse teooriat, mille iga default on kujul:

$$\frac{A : B}{B}$$

poolnormaalseks teooriat, mille defaultidel on kuju

$$\frac{A : B \ \& \ C}{B}$$

Normaalsed defaultid võimaldavad formaliseerida väga suurt hulka praktikas esinevaid reegleid, kuid täiesti piisavad nad ei ole. D. Etherington (9) toob näiteks järgmise teooria:

$$(1) \ \frac{\text{Täiskasvanu}(x) : \text{Omab} \dots \text{tööd}(x)}{\text{Omab} \dots \text{tööd}(x)}$$

$$(2) \ \frac{\text{Väljalangenu}(x) : \text{Täiskasvanu}(x)}{\text{Täiskasvanu}(x)}$$

### (3) Väljalangenu(x) : 7 Omabtööd(x)

7 Omabtööd(x)

Teisiti öeldes, selle teooria järgi täiskasvanud inimene omab tüüpiliselt tööd; koolist väljalangenud on tüüpiliselt täiskasvanud; ja koolist väljalangenutel tüüpiliselt ei ole tööd. Arusaadavalt on siin tegemist kapitalistliku ühiskonnaga, kus koolist väljalangenud oma hilisemas elus on enamasti töötud. Olgu John üks selline väljalangenu. Tahaksime (3) abil näidata, et tal tööd ei ole. Selleks on tarvis, et me ei saaks näidata, et tal tööd on; paraku viitavad (2) ja (1) sellele, et Johnil kui täiskasvanul peaks siiski töö olema. Konflikt (1) ja (3) vahel viib selleni, et me ei saa teooriast järeldada, kas Johnil on töö või ei ole, kuigi intuitiivselt oleme valmis järeldama, et tal tööd ei ole. Viga on selles, et väljalangenud on üldjuhul täiskasvanute alamhulk ning reeglite konflikti puhul peaksime eelistama selles alamhulgas kehtivaid konkreetseid reegleid kogu täiskasvanute hulgas kehtivatele üldisematele reeglitele. Ethington soovib asendada normaalne default (1) poolnormaalsete defaultiga:

(1') Täiskasvanu(x) : Omabtööd(x) & 7 Väljalangenu(x),  
Omabtööd(x)

mis jätab koolist väljalangenud reegli rakendamispiirkonnast (so täiskasvanute hulgast) lihtsalt välja. Selline lähenemine on muidugi väga mõistetav, kuid tal on mitu puudust. Kõigepealt: ebatüüpiliste alamhulkade otsene väljaeraldamine ei sobi hästi default-loogika ideoloogiaga. Teiseks, uute default-reeglite sissetoomine teooriasse muutub keeruliseks, sest iga kord tuleb üles otsida võimalikud konfliktid, ja kui vaja, siis äsjatoodud näite eeskujul modifitseerida vastavaid üldisemaid defaulte.

### 2.3. Defaultide järjestamine.

Teine võimalus konfliktide lahendamiseks on defaultide osaline järjestamine: näitame ära mingi hierarhia nii, et ainult mingi konkreetse alamhulga kohta kehtivat default-reeglit ei saaks blokeerida selle hulga ülemhulga kohta kehtiva üldisema default-reegliga. Meie näites tuleks deklareerida (3) < (1), keelates niimoodi defaulti (3) blokeerimiseks kasutada defaulti (1). Viimane meetod tähendab sisuliselt ~~edasama~~ mis poolnormaalsete defaultide kasutamine, kuid on

mõnest aspektist parem. Kõigepealt ei ole vaja uute defaultide lisamisel otseselt modifitseerida vanu defaulte. Piisab sellest, kui me määrame uue defaulti koht defaultide hierarhias. Siinkohal rõhutame, et kirjeldatud järjestus on osaliline, st ei esine iga kahe defaulti vahel (sisuliselt: kui kaks defaulti ei ole omavahel ülemhulga-alamhulga seoses, siis võivad nad üksteist blokeerida).

Vaatame lähemalt probleemi, mis tekib uute defaultide lisamisel teooriasse: kuidas leida need defaultid, millega uuel defaultil võib tekkida konflikt ning mis on rakendatavad üldisemal juhul kui uus default? Kuidas mele näites leida, et konflikt (3) ja (1) vahel tuleb lahendada (3) kasuks? Ilmselt tuleks püüda näidata, et üldiselt kehtib Väljalangu( $x$ ) $\rightarrow$ Täiskasvanu( $x$ ), st et väljalangenud on täiskasvanute alamhulk. Sisuliselt tähendab see, et kui meil on kaks defaulti, (1) ja (2), ning meil õnnestub üldjuhul näidata, et  $A(1)\rightarrow A(2)$ , kus  $A(1)$  ja  $A(2)$  on defaultide ülemised vasakpoolsed valemid, siis me ei tohi (2) kasutada (1) blokeerimiseks.

Defaultide järjestamise äsjatoodud printsiip nõuab muidugi veel hoolikat kontrollimist ja täpsustamist; siin tekib hulgaliselt komplikatsioone ja keerukusi ning ilmselt on just järjestamine üks default-teooriate võtmeküsimusi.

Nii või teisiti, sellestasamast järjestamise probleemist ei pääse mööda ka Etheringtoni lähenemisviis, sest konfliktide lahendamine poolnormaalsete defaultide abil eeldab ikkagi, et me teame, milliseid konflikte ja kuidas lahendada, st teame järjestust. Sellisel juhul aga võime konfliktide lahendamiseks kasutada otseselt järjestust ning poolnormaalsuse järele ei ole vajadust.

Kui tahame default-loogikal põhinevat andmebaasi tege-likult ehitada, siis on igal juhul vaja, et loodav süsteem suudaks defaultide järjestuse automaatselt kindlaks teha. Siin on kaks võimalikku lähenemisviisi:

- 1) järjestus on antud ilmsel kujul ning seda modifitseeritakse iga kord, kui andmebaasi lisatakse uus valem või likvideeritakse vana;
- 2) järjestust ei ole ilmsel kujul näidatud; töestaja kasutab järjestamisprintsiipe selleks, et töestamise käigus iga konkreetse defaulti jaoks "välja rehkendada", kas seda tohib antud momendil kasutada või mitte.

Käesolevas artiklis piirdume ainult nõ. puhaste default-

tide vaatlemisega, kuid praktikas võib tihti osutada kasulikuks defaultide mingite metareeglitega seotud tõenäosuste lisamine. Selline tõenäosustega täiendatud default-teooria kujutab endast "muutuva täpsusega loogikat" (4), kus on võimalik kasutada niisuguseid mittetäielikke tõestusstrateegiaid, mille puhul defaulti blokeerimisele suunatud otsingute maht on piiratud näiteks otsingu sügavusega või mingite teiste metateadmistega (3).

### 3. Dialoogi mõnede põhimõtete formaliseerimisest.

Järgnevas püüame valgustada dialoogi pragmaatiliste aspektide (1), (10) formaliseerimise küsimusi. Mingisuguse tervikliku dialoogipragmaatika teooria väljatöötamist ei ole eesmärgiks võetud. Püütakse näidata, kuidas saaks loogikat (eriti default-loogikat) dialoogiprintsiipide formaliseerimisel rakendada. Seejuures vaadeldakse ainult kooperatiivseid dialooge. Esitatavad printsiibid on küll lihtsustatud, kuid mitte alusetud.

#### 3.1. Teadmised suhtluspartneri kohta.

Dialoogi pidamiseks on oluline, et dialoogist osavõtjatel, A-1 ja B-1, oleks teatud ettekujutus partneri teadmistest, sealhulgas ka teadmistest tema enda teadmiste kohta jne jne potentsiaalselt lõpmatuseni (A teab, et B teab, et A teab, et B ei tea, et ...). Sellise andmebaasi ehitamise probleemid ei ole triviaalsed (2), (7) ja kuna me käesolevas nendega otseselt ei tegele, kasutame edaspidi lihtsalt sellist kirjaviisi: valem  $U_{sub}(B, "info")$  A andmebaasis tähendab, et A teada B usub "info"-t (info on mingisugune valem).

$U_{sub}(B, "U_{sub}(A, "info")")$  tähendab, et A teada B usub, et A usub info-t.  $U_{sub}(B, "info1") + U_{sub}(B, "info2")$  tähendab, et A teada B usub kas info1-te või info2-te (plussmärk on välistava "või" tähenduses).

$(\exists i) U_{sub}(B, "i") \& (i \rightarrow V)$  on teist järku predikaatavutuse valem, mis ütleb, et B usub midagi (i on mingi valem), millest meie saaksime järeldada V. Kui me ise V-d ei tea, siis on selle valemi sisuliseks tõlgenduseks, et B-1 on meie jaoks kasulikku informatsiooni. Predikaati Usub tõlgendame

nii: B on võimeline (oma isikliku tuletusalgoritmiga) oma-  
enda andmebaasist tuletama i.

Vaatame nüüd järgmist olukorda: kohtuvad kaks üliõpi-  
last, A ja B, kes üksteisest enne midagi kuulnud ei ole,  
ning hakkavad vestlema. Juba enne vestluse algust on kummal-  
gi mingi ettekujutus teise teadmistest olemas. See ettekujut-  
us ei pruugi muidugi olla tõene, veel vähem on see täielik.  
Mõningates A teadmistes võib B kindel olla:

$\text{Inimene}(x) \ \& \ \text{Paistab harilik}(x) \rightarrow \text{Usub}(x, \text{"Inimene(B)"}).$

Mõnedes aga mitte:

$\text{Üliõpilane}(x) : \text{Usub}(x, \text{"Asukoht(õppehoone1, van.tänav)"})$

$\text{Usub}(x, \text{"Asukoht(õppehoone1, van.tänav)"})$

Oletused kaaslane teadmiste kohta esitame defaultidena, kind-  
lad teadmised kaaslane teadmiste kohta aga harilike valemi-  
tega. Selliseid oletusi võib hilisem dialoog kinnitada või  
ümber lükata. Näiteks, kui B saab teada, et A tõesti teab,  
kus õppehoone nr.1 asub, võib ta lisada oma andmebaasi juba  
kindla fakti:

$\text{Usub}(A, \text{"Asukoht(õppehoone1, van.tänav)"})$ .

Kui teame, et kaaslane usub midagi (mingit i-d), ei  
pruugi me olla kindlad, et see i on tõsi, sest kaaslane võib  
eksida. Asjaolu, et me usume, et i on tõsi, kirjutame edas-  
pidi nii: Tõsi("i"). Lisaks teadmistele teise teadmiste koh-  
ta on meil veel teadmisi tema tuletusalgoritmi kohta. Näiteks  
olukorras, kus vestluskaaslane on purjus või nõrgamõistuslik,  
ei saa temalt oodata, et ta näeks läbi keerulisi seoseid, st  
tema tuletusalgoritm on osaliselt piiratud. Muidugi, ka meie  
endi algoritm (nagu enamik algoritme) on piiratud. Käesolevas  
jätame küsimuse sellistest teadmistest üldse käsitlemata ja  
eeldame, et suhtlejate tuletusalgoritmid on sarnased, st kaas-  
lase algoritmi lähendina kasutame omaenda algoritmi.

### 3.2. Lihtsamate vastuste interpreteerimine.

Vaatame situatsiooni, kus B küsis A-lt mingi ja-ei kü-  
simuse, näiteks: "Kas Jaan on raamatukogus?", ja sai mingi  
vastuse V (näiteks: V1: "Jah", V2: "Ei", V3: "Ta on oma koms-  
pekti juba ära andnud"). Küsija eesmärgiks on nüüd vastusest

võimalikult palju informatsiooni välja pigistada.

1) Eeldame, et B-l on mingi konkreetne eesmärk, mille saavutamiseks vajalikku infot ta püüab dialoogis A käest saada. Ja eeldame, et B on kindel, et A on B sellest soovist teadlik (näiteks on B seda A-le dialoogi käigus öelnud või mõista andnud). See tähendab:  $U_{sub}(A, "Soovibteada(B, "L")")$ , kus L tähistab B-d huvitavat lisainfot. Olgu L fakt "Jaaniil on konspekt".

2) B oletab, et A teab teda huvitavat asjaolu (näiteks, kas L või  $\neg L$ ):

$(U_{sub}(A, "L") \& Tõsi("L")) \vee (U_{sub}(A, "\neg L") \& Tõsi("\neg L"))$ . Kui B on kindel A teadmises, siis B andmebaasis on see valem äsjatoodud kujul; kui B ei ole kindel, siis default-kujul.

3) B usub, et A teab, et B peab tõenäoliseks L-i (miks muidu ta Jaani asukohta küsib):  $U_{sub}(A, "U_{sub}(B, \frac{":L"}{L})")$ .

Kui nüüd A teatab otseselt L-i või  $\neg L$ -i, siis B võtab selle lihtsalt teadmiseks: Tõsi("L") või Tõsi(" $\neg L$ ") ning täiendab oma ettekujutust A teadmiste kohta:  $U_{sub}(A, "L")$ ,  $U_{sub}(A, "U_{sub}(B, "L")")$ . Viimasest valemist võib B reegli  $U_{sub}(x, "y") \rightarrow \neg Soovibteada(x, "y")$  abil näiteks tuletada, et  $U_{sub}(A, "\neg Soovibteada(B, "L")")$ . Peale teadmiste A hetke-teadmiste kohta täieneb B andmebaas ka teadmistega A eelne-nud teadmiste kohta, kuid jätame selle aspekti siin vaatluse alt välja.

Kui A ei ütle L-i kohta otseselt midagi (vastab lihtsalt "jah" või "ei"), peab B seda interpreteerima, nagu oleks A öelnud L, sest ta on kindel, et A teab, et B usub, et A teab, kas L või  $\neg L$ .

Kui A ei tea, kas L või  $\neg L$ , aga arvab, et B oletab A-d seda teadvat, peab ta teatama B-le, et ta ei tea.

Kõik niisugused kaalutlused on loogika vahenditega lihtsalt formaliseeritavad. Kirjutame näiteks välja olukorra, kus A ei tea, kas L või  $\neg L$ :

$\neg U_{sun}("x") \& \neg U_{sun}("\neg x") \& Tahabteada(y, "x") \& U_{sub}(y, "U_{sub}(Mina, "x") \vee U_{sub}(Mina, "\neg x")") \rightarrow Teatan(y, "\neg U_{sub}(Mina, "x") \& \neg U_{sub}(Mina, "\neg x")")$ .

### 3.3. Vastuse informatiivsuse suurendamine.

Kui B soovib A-le anda mingit informatsiooni i, peab ta oma teate A-le formuleerima nii, et A saaks sellest või-

malikult palju informatsiooni (i + veel midagi). Siis peab olema B jaoks garanteeritud:  $U_{sub}(A, "M \rightarrow i")$ , et B tarvis oleks tema enda teatel mõte. Kuidas aga leida võimalikult informatiivset M-i? Selleks uurime, kuidas me enda jaoks i tuletasime ning proovime iga sammu selles tuletuskäigus tuletada A oletatavate teadmiste abil. Kui me ei leidnud ühtegi sammu, mida ei oleks saanud A teadmistega tuletada, peavad meie teadmised A teadmiste kohta olema puudulikud (eeldame, et A ei tea i-d ja tema tuletusalgoritm ei ole halvem meie omast). Sellisel juhul peame loomulikult täpsustama oma ettekujutust A teadmistest, st alustama dialoogi, mille eesmärgiks meie poolt on välja selgitada, millised default-teadmised A teadmiste kohta ei ole õiged. Kõige lihtsam on küsida järjest A iga oletatava teadmise kohta, mida me kasutaksime A oletatavas tõestuses i kohta, kas A seda tegelikult teab. Kui A neist mõnda ei tea, ütleme selle talle ja uurime järele, kas A sai aru, et i. Kui ei, kordame uuesti sama protsessi.

Vaatame nüüd juhtu, kus meil A oletatavate teadmiste abil ei õnnestunud tuletada kõiki omaenda tõestuskäigu samme. Siis on vaja A-le anda informatsiooni nende sammude kohta ning teha seda jälle võimalikult informatiivselt. Meie lihtsustatud pildist maksimaalselt informatiivse teate väljatöötamise kohta on näha, et see protsess on rekursiivne.

Äsjakirjeldatud protsessi käigus tuleb erilist tähelepanu pöörata sellistele tõestuse sammudele, kus blokeeritakse mõni default, sest on väga tõenäoline, et vestluskaaslase vead on tingitud mõnest tema eelarvamusest (st defaultist), mis antud olukorras ei kehti.

### 3.4. Üks lihtsustatud vastamisteooria.

Järgnevas tähendab Oluline(x, "y"), et x tahab saada vastust küsimusele y, kuid ta ei pruugi teada, kas vestluspartner suudab talle vastata või ei. Vastus("z", "y") tähendab, et z on vastus küsimusele y selles mõttes, et kehtib kas 1, 2 või 3:

1)  $z \equiv y$  ("jah")

2)  $z \equiv \neg y$  ("ei")

3) z on y-i mingi konkretiseering. Näiteks:  $y = \text{"Asukoht(Jaan, 1)"}$ ,  $z = \text{"Asukoht(Jaan, raamatukogu)"}$ .

$U_{sub}(x, "z")$  tähendab nagu varemgi, et x suudab oma andmebaa-

sist tuletada z-i. Veendunud(x, "z") tähendab, et x suudab z-i tuletada ilma default-reegleid kasutamata. Teatab(x, y, "z") tähendab, et x peab y-le teatama z-i.

1) Esialgse vastuse, n.ö. vastuse lähendi väljatöötamine:  $(\exists v) (\exists v1) \text{Usub}(x, \text{"Oluline}(y, \text{"k"})") \& \text{Usub}(x, \text{"v"}) \& \text{Veendunud}(x, \text{"v"}) \& \text{Vastus}(\text{"v"}, \text{"k"}) \& \text{Veendunud}(x, \text{"v1"}) \& \text{Usub}(x, \text{"Usub}(y, \text{"v1} \rightarrow \text{v"})") \& \text{Usub}(x, \text{"!k1"}) \& \text{Usub}(x, \text{"!Usub}(y, \text{"!k1} \& (\text{v1} \rightarrow \text{k1})")") \rightarrow \text{Teatab}(x, y, \text{"v1"})$ .

Näide artiklist (1): A ja B on üliõpilased. A küsib B-lt: "Kus Jaan on?". B ei tea kindlalt, aga oletab, et raamatukogus, sest ta nägi Jaani autot raamatukogu ees. Käesolev valem käsib B-l vastata midagi sellist: "Ma nägin ta autot TRÜ raamatukogu ees parkimisplatsil teatripoolses ääres". Valem arvestab ka asjaolu, et ei tohi teatada midagi, mida teine võib valesti tõlgendada.

2) Ülearuse informatsiooni kõrvaldamine.

$(\exists v1) \text{Teatab}(x, y, \text{"v"}) \& \text{Usub}(x, \text{"Usub}(y, \text{"(v1} \rightarrow \text{Oluline}(y, \text{"v"})") \& \text{!}(v1 \rightarrow v)")) \& \text{Usub}(x, \text{"!k1"}) \& \text{Usub}(x, \text{"!Usub}(y, \text{"!k1} \& (\text{v1} \rightarrow \text{k1})")") \rightarrow \text{Teatab}(x, y, \text{"v1"}) \& \text{Eiteata}(x, y, \text{"v"})$ .

Eiteata(x, y, "z") tähendab, et fakt Teatab(x, y, "z") kaotab kehtivuse.

Viimast valemit tuleb valemi 1 poolt genereeritud toorteadetele senikaua rakendada, kuni me ei saa enam ühtegi uut teadet. Siis võib saadud teadete hulga lugeda optimaalseks ja lõpuks tõepoolsest teatada. Eelnevas näites toodud toorteadete asendataks valemi 2 abil tõenäoliselt teatega: "Ma nägin ta autot raamatukogu ees".

Valem 2 aitab lahendada ka järgmist situatsiooni: A vajab konspekti ning arvates, et Jaanil on konspekt, küsib B-lt: "Kas Jaan on raamatukogus?". Kui B orienteerub A huvides ning teab, et Jaan andis konspekti äsja ära, tuleb tal vastata umbes nii: "Ta andis konspekti äsja ära" ja jätta A otsene küsimus vastamata.

#### K i r j a n d u s

1. Õim, H. Pragmaatika ja keelelise suhtlemise teooria. - Keel ja Kirjandus, 1986, nr.5, lk. 257-269.
2. Barnden, J.A. Imputations and Explications: Representational Problems in Treatments of Propositional Attitudes. - Cognitive Science. 1986, v. 10, No. 3, pp. 319-365.

3. Davis, R. Meta-Rules: Reasoning about Control. - Artificial Intelligence, 1980, v. 15, No. 3, pp. 179-226.
4. Michalski, R.S., Winston, P.H. Variable Precision Logic. - Artificial Intelligence, 1986, v. 29, No. 2, pp. 121-147.
5. McCarthy, J. Circumscription - A Form of Non-Monotonic Reasoning. - Artificial Intelligence, 1980, v. 13, No.1-2, pp. 27-39.
6. McDermott, D., Doyle, J. Non-Monotonic Logic I. - Artificial Intelligence, 1980, v. 13, No. 1-2, pp. 41-72.
7. Perlis, D. Languages with Self-Reference I: Foundations. - Artificial Intelligence, 1985, v. 25, No. 3, pp. 301-323.
8. Reiter, R. A Logic for Default-Reasoning. - Artificial Intelligence, 1980, v. 13, No. 1-2, pp. 81-132.
9. Etherington, D.W. Formalizing Non-Monotonic Reasoning Systems. Technical Report 83-1, Dept. Computer Science, University of British Columbia.
10. Radical Pragmatics. Ed. P. Cole. New York, Academic Press, 1981.

DEFAULT.-ЛОГИКА И ВОЗМОЖНОСТИ ЕЕ ПРИМЕНЕНИЯ  
В ДИАЛОГОВЫХ СИСТЕМАХ

Т. Таммет

Р е з ю м е

Обсуждаются логические формализмы как средства представления знаний в системах искусственного интеллекта. Выделяется класс немонотонных логик и среди них -DEFAULT - логики, как особенно перспективные для применения в системах моделирования естественных рассуждений, в том числе в системах естественно-языкового диалога пользователя с ЭВМ. Обсуждаются некоторые возможные пути развития и усовершенствования таких логик.