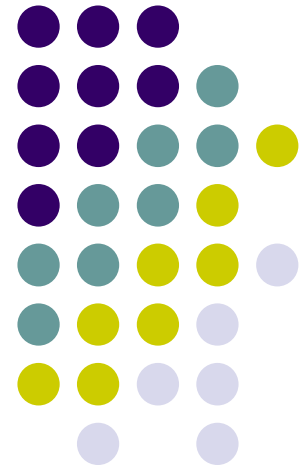
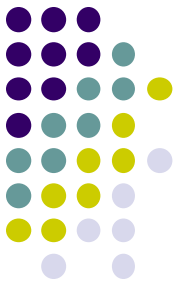


LOOGILINE PROGRAMMEERIMINE *(logic programming)*



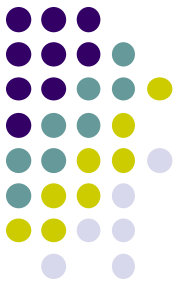
J.Vain





Kursusest üldiselt

- Kood: [ITI0021](#) 5.0 4 2-2-0 H S
 - <http://courses.cs.ttu.ee/>
- Kontakt:
 - Õppejõud: prof. Jüri Vain
 - Vastuvõtuaeg: N kl 16:00 – 17:00 (eelnevalt teatada)
 - E-post: juri.vain@ttu.ee
 - Telefon: 6204190
 - Aadress: ICT-419



Kursuse korraldus

- Loeng – prof. Jüri Vain
 - **IAPB51, IAPB52, IAPB53, IAPB54, IAPB55**
 - N 10:00-11:30 CYB - VEENUS

- Praktikum - Evelin Halling, Sven Nõmm
 - **IAPB51, IAPB52**
 - N: 12:40-14:10 (ICT-401)
 - **IAPB53, IAPB54**
 - N: 14:15-15:45 (ICT-401)

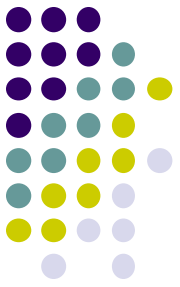
Hindelise arvestuse nõuded



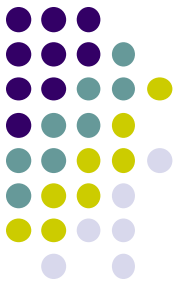
- Praktikumi ülesannete (10 tk) esitamine (tähtajaliselt) – max 10%
- +
Loengute (10 tk) kontrollküsimused – max 10%
- +
Test 1 – põhimõisted - 20%
- +
Test 2 – rekursiooni programmeerimine – 30%
- +
Test 3 – teadmiste esitamine Prologis – 20%
- +
Kodutöö: kabeprogramm (10%) + turniir

NB! Turniiri kohad I-III annavad lisapalle vastavalt 3%, 2% ja 1%

Õppeaine eesmärgid

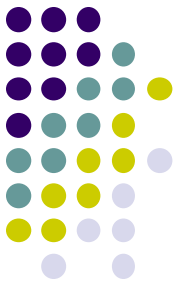


- Teadmiste esitamine kitsenduste süsteemina (Horni laused)
- Järelduste tuletamine teadmusbasisist kasutades *resolutsiooni* meetodit ja *unifitseerimisreegleid*
- Põhilised teadmiste *tuletusmeetodid* ja nende programmeerimine keeles Prolog
- Rakenduslike ülesannete (n. kitsendustega planeerimine) formaliseerimine ja *praktiline programmeerimine*



1.1. Mis on loogiline programmeerimine? (2)

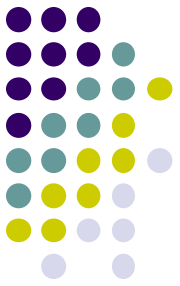
- LP sobib *tehisintellekti rakenduste* programmeerimiseks:
 - loomuliku keele analüüs (DCG grammatikareeglid)
 - ekspertsüsteemid (otsingu- ja järeldusreeglid)
 - kujundituvastus (tuvastusreeglid)
 - kitsendustega planeerimine (logistika, marsruudi otsimine)
 - rekursiivsete funktsioonide püsipunkti arvutus
 - jne
- LP ei sobi rakendustes, kus
 - kiired numbrilised arvutused (n. maatriksarvutused, võrrandid)
 - OOP (kuigi on toetatud mõnes prologis)
 - kasutajaliideste programmeerimine
 - masingraafika



1.1. Mis on loogiline programmeerimine? (3)

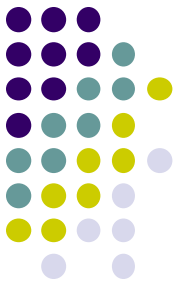
Miks tasub õppida LP?

- Õpetab mõtlema *probleemikeskselt* ja esitama lahendusi *abstraktsel* kujul
 - Programmi põhifunktsioonid:
 - reaalse maailma probleemide *abstraktne esitamine*,
 - abstraktsioonide teisendamise ja sidumise omavahel
 - abstraktsioonide põhjal arvutamine/otsuste tegemine
 - *Progromeerimiskeel* peab võimaldama
 - kirjeldada ja analüüsida abstraktsioone *arvutile* sobival kujul
 - *Deklaratiivsed* programmeerimiskeeled sobivad
 - abstraktsete objektide ja nende seoste kirjeldamiseks
 - *väldivad protseduurseid detaile*



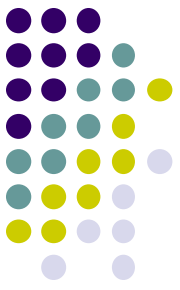
1.1. Mis on loogiline programmeerimine? (4)

- Universaalne keel omaduste/seoste abstraktseks kirjeldamiseks on loogika
- → LP on programmeerimine loogika keeles!
- Prolog – *programming in logic*
- LP \neq Prolog



1.2 LP ajalugu

- Prolog (1972)
 - Alain Colmerauer, Phillipe Roussel;
- Edinburgh Prolog (1980 algus)
 - David Warren;
- 1980 – 2015 – laiendamine teiste programmeerimisparadigmadega
 - paralleelsus, OO, andmetüübid jm
 - palju Prologi dialekte



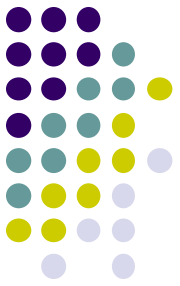
1.3 LP meetod

- Piiritleda valdkond:
 - reaalse maailma modelleeritav situatsioon (domain, use-cases)
 - määratleda sellega seotud põhimõisted
 - defineerida mõisteid iseloomustavad atribuudid ja nende omadused
 - defineerida seosed atribuutide vahel
 - Formaliseerida valdkonna objektid ja seosed LP keeles
 - tulemusena tekib faktide/tuletusreeglite struktuur
- Näide:
- raamatukogu asub 2. korrusel, <-- fakt
 - Sokrates on kreeklane <-- fakt
 - kõik kreeklased on inimesed <-- reegel
- Saadud teadmiste struktuuridel formuleerida päringud
- Näide:
- Kas Sokrates on inimene?



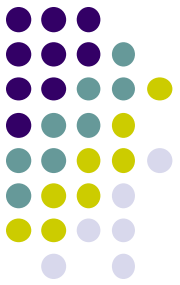
LP “õrnad” kohad

- Teadmiste esitamine on *otsingureeglite tundlik*
 - päringu tulemus oleneb otsingureeglist ja faktide järjestusest teadmusbaasis
 - tagasivõtu (*backtracking*) mehhanismist arusaamine nõuab otsingumootori tundmist
- Keeruline saavutada “puhast deklaratiivsust”
- Efektiivsuse saavutamiseks peab tundma Prologi otsingumootorit
- Praktilises programmeerimises vaja ka nn “madala taseme” käske:
 - kasutajaliidese juhtimine,
 - failisüsteemi käsud,
 - stringioperatsioonidjms.



LP edasiarendused

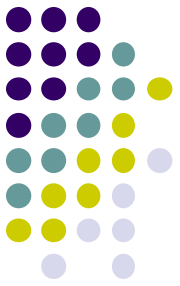
- Laiendamine teiste programmikeelte paradigmadega
 - Functional logic programming
 - <http://www.informatik.uni-kiel.de/~mh/FLP/>
 - keeled [Curry](#) and [Mercury](#).
- Efektiivsuse suurendamine
 - Concurrent prolog
 - [Curry](#), [ToonTalk](#), [Janus](#), [Alice](#)
- Probleem-orienteerituse suurendamine
 - Constraint Logic Programming
 - http://en.wikipedia.org/wiki/Constraint_logic_programming)
 - Semantiline veeb
 - <http://hcs.science.uva.nl/projects/SWI-Prolog/articles/mn9c.pdf>



Kursuse sisu

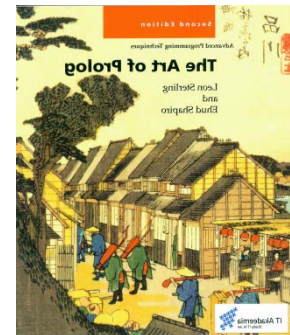
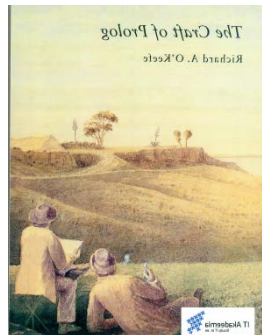
- Alusmõisteid loogikast
 - Loogikasüsteem (-keel, -arvutus)
 - Termide unifitseerimine ja võrdlemine
 - Tõestusmeetod - resolutsioon
- LP andmestruktuurid (listid, semantilised võrgud, freimid)
- LP deduktiivne ja protseduurne semantika
- Prologi süntaks ja operaatorid
- Prologi otsingumootor, otsingu juhtimine
- Kitsenduste süsteemide kirjeldamine ja lahendamine Prologis
- Rakendusnäiteid:
 - teadmusbasisid ja reisiplaani koostamine
 - loomulike keelte analüüs (parsimine)
 - kujundituvastus ja keerdülesanded
- Näpunäiteid praktiliseks programmeerimiseks: integreerimine Java ja C++ga.

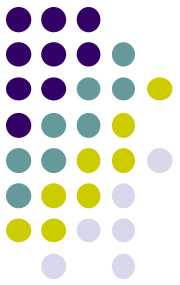
Õppematerjal



- Õpikud TTÜ raamatukogus:

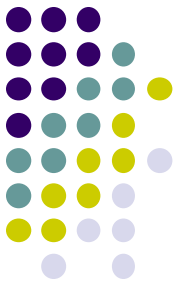
- Tõnu Tamme. Loogilise programmeerimise meetod. Tartu Ülikool 2003. (algajatele)
- R.A.O'Keefe The Craft of Prolog, MIT Press (sissejuhatav)
- L. Sterling, E. Shapiro, The Art of Prolog. (edasijõudnutele)
- I. Bratko, "Prolog Programming for Artificial Intelligence", Addison–Wesley Ltd. (rakendusprogrammerijatele)





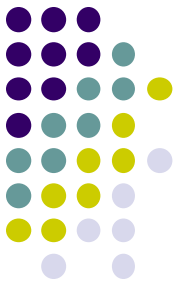
Lisamaterjal

- Ajakirjad:
 - The Journal of Logic and Algebraic Programming
 - (<http://www.informatik.uni-trier.de/~ley/db/journals/jlp/jlap.html>)
 - Theory and Practice of Logic Programming
 - (<http://www.cwi.nl/projects/alp/TPLP/>)
- SWI prologi help



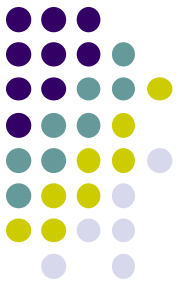
Veel kasulikke linke

- Peter Hancox. Prolog and Logic Programming. School of Computer Science in the University of Birmingham, UK.
 - http://www.cs.bham.ac.uk/~pjh/prolog_course/sem242.html
- [Ischislenie Vyskazyvaniy I Logicheskoe Programmirovanie](#)
 - 2012/5/2 Vladimir Kulakov **Vene keeles!**
- The World Wide Web Virtual Library: Logic Programming
- Guide to Prolog Programming
 - <http://kti.mff.cuni.cz/~bartak/prolog/implementations.html>
- Object-Oriented Prolog
 - http://www.cetus-links.org/oo_prolog.html
- [Jonathan Bowen](#) Logic Programming page
 - http://formalmethods.wikia.com/wiki/Logic_programming



Kuidas hankida oma Prolog?

- Unix, Windows, Linux:
 - ALS (Applied Logic Systems, Inc.) Prolog compiler
 - BinProlog, BinNet Corp. See also Jinni (Java INference Engine and Networked Interactor).
 - GNU Prolog compiler - free Prolog compiler with constraint solving over finite domains.
 - IF/Prolog system. IF Computer. Unix, Windows 95/98/NT.
 - IT ProLog. IT Masters. (Unix and Windows NT).
 - LPA WIN-PROLOG, MacProlog32 and Prolog++. Logic Programming Associates Ltd.
 - Quintus Prolog. For Unix and MS Windows.
 - SICStus Prolog (commercial, portable) Unix machines, Windows.
 - **SWI-Prolog**. Unix and MS Windows. Portable.
- PC Prologid:
 - YAP Prolog System (Yet Another Prolog) – kiire Prologi kompilaator, Linux/Solaris/Windows NT, 95, 98. Akadeemiline litsents vaba.
 - Amzi Prolog + Logic Server. (Commercial). Windows 3.x, 95, WFW, NT 3.5x, DOS, Extended-DOS. Allows embedding of Prolog components in C/C++, Visual Basic, Delphi, Access, etc. <http://www.amzi.com/download/freedist.htm>
 - ADA Prolog (aeglane) ja ESL Prolog (hea, kiire).
 - LPA WIN-Prolog. Windows 3.1, Macintosh ja MS-DOS.
 - Qu-Prolog. Support symbolic computation for mathematical notations and languages such as Z.
 - Visual Prolog from the Prolog Development Center. DOS, Windows 3.1/95/98, NT, Linux.



Kuidas hankida oma Prolog?

- Praktikumides ainult SWI prolog!!
 - <http://www.swi-prolog.org/>

Küsimused?

