Зертханалық жұмыс №2. Visual Prolog логикалық программалау тілінің негізі.

Мақсаты*:* Фактілер, ережелер және сұрауларды орналастыру әдістері.

Ұсыныстар

Пролог тілін құрайтын тек екі түрлі ұғым бар: ереже немесе факті. Бұл ұғымдар прологта ұсыныс (clause) терминімен белгілі. Прологтағы программалардың негізі ұсыныстардан тұрады.

Факті объекттердің арасындағы қатынасты немесе қасиеті көрсетеді. Факті өзіне-өзі жеткілікті. Прологқа айғақтың расталуы үшін қосымша дәлелдеулер керек болмайды және де факті логикалық қорытындының негізі ретінде қолданылды.

Ережелерді қарастырсақ. Проглогта бір фактінің сенімділігінен басқада бірнеше фактілер табуға болады. Ереже – бұл берілгендерді логикалық түрде суреттейтін прологтың құрылымы. Ереже - бұл нақтылы қасиет немесе қатынас, басқада қатынастар белгілі болғанда. Бұл қатынастар үтірлермен бөлінген.

Ережелерге бірнеше мысалдар келтірейік.

1-Мысал**.** Кейбір тағамдардың Диана мәзіріне сәйкес келетіндігіне қорытынды жасай алатын ережелер.

Диана-вегетарианшы ол тек қана өзінің дәрігерінің айтқанымен тамақтанады. (Diane is a vegetarian and eats only what her doctor tells her to eat)

Алдыңғы ереже мен мәзірді біле отырып, мәзірдегі тағамдарды Диана таңдауын алдын ала қорытынды жасай аласыз. Сіз бұл жұмысты іске асыру үшін тағамның берілген шектерге сәйкес келетіндігін тексеруіңіз керек:

• Food_on_menu көкөніс болып табылады ма?

• Food_on_menu дәрігердің тізіміне кіреді ме?

• шешім: егер екі жауап оң болса, Диана Food_on_menu- тапсырыс бере алады.

Прологтағы ұқсас қатынас ережемен анықталуы керек, өйткені қорытынды фактіге сай негізделген. Ережені жазудың бір нұсқасы:

diane_can_eat(Food_on_menu):-vegetable(Food_on_menu), on_doctor_list(Food_on_menu).

vegetable(Food_on_menu) -дан кейін үтір тұратынына көңіл аударыңыз. Ол бірнеше мақсаттардың конъюнкциясын айқындайды және "және" сияқты оқылады; екі ереже де - diane_can_eat(Food_on_menu) үшін vegetable(Food_on_menu) және on_doctor_list(Food_on_menu) ақиқат болуы керек.

2-Мысал**.** Егер Personl Person2-нің ата-анасы болып табылса, сіз ақиқат фактісін жазғыңыз келеді деп болжаймыз. Керек факті осылай көрінеді: parent(paul, samantha).

Бұл айғақ бойынша Пол, Самантаның ата-анасы екендігін білдіреді. Бірақ Visual Prolog мәліметтер қорында факілер бар деп болжайық. Ол

айғақтар әкелік қатынасты қалыптастырады. Мысалы, "Пол Самантаның әкесі":
father(paul, samantha).

Енді сізде аналық қатынас қалыптастырушы айғақ болсын дейік, "Джулия - Самантаның анасы":
mother(julie, samantha).

Егер де сіздерде бірнеше фактілер бар болып, және олар аталық-аналық қарым-қатынастарды қалыптастыратын болса,онда мәліметтер қорындағы әрбір ағайынды қатынасқа уақытты өлтірудің қажеті жоқ.

Сіз білетіндей, Personl Persona-ның ата-анасы, егер Personl-Person2-нің әкесі немесе Personl Person2-нің анасы болса, онда неге бұл шектерді біріктіретін ережелерді жазбасқа? Бұл шарттарды табиғи тілде қалыптастырғаннан кейін оларды пролог ережесі бойынша жазу жеткілікті.
parent(Personl, Person2):- father(Personl, Person2).
parent(Personl, Person2):- mother(Personl, Person2).

Бұл пролог ережелері

Personl Person2-нің ата-анасы болады, егер Person1 Person2-нің әкесі болса және

Personl Person2-нің анасы болады,егер Personl Person2-нің анасы болса.

3-Мысал**.** Егер адамға машина (likes) ұнаса және машина сатылса (for sale), онда ол машинаны сатып ала алады.

Бұл ретте қатынас пролог тіліне келесі ереже бойынша ауыстырыла алады:
can_buy(Name, Model):-person(Name),
car(Model), likes(Name, Model), for_sale(Model).

Бұл ережелер келесі қатынастарды сипаттайды:

Name сатып ала алады (can_buy) Model, егер Name адам(person) және Model машина (car) болса, және Model сатылса (for_sale) Name Model –ді ұнатады (likes). Егер бұл ереже 4 шартты қанағаттандырса, онда ол ақиқат болады.

Автомобилді сату проблемасының шешудің жолын іздейтін ch02e02.pro-программасы көрсетілген. Тексеріп көріңіз:

predicates
can_buy(symbol, symbol)
person(symbol)
car(symbol)
likes(symbol, symbol)
for_sale(symbol)
clauses
can_buy(X,Y):-person(X), car(Y), likes(X,Y), for_sale(Y).
person(kelly).

person(]udy).
person(ellen).
person(mark).
car(lemon).
car(hot_rod).
likes(kelly, hot_rod).
ikes(judy, pizza).
likes(ellen, tennis).
likes(mark, tennis).
for_sale(pizza).
for_sale(lemon).
for_sale(hot_rod).

Джуди және Келли не сатып ала алады? Hot_rod-ты кім сатып алуы мүмкін? Келесі мақсаттарды GOAL: -бөлімінде сынап көріңіз. can_buy(Who, What). can_buy(judy, What). can_buy(kelly, What). can_buy(Who, hot_rod).

Бұл программаға басқа фактілер мен жаңа ережелер қосайық. Өзіңіздің алдында құрған сұраныстарыңыз бойынша программаны тексеріп көріңіз. Қорытындының нәтижелері сіздің күткеніңізді ақтады ма, әлде жоқ па?

Жаттығулар**:**
1. Visual Prolog-та келесі ережелеріне сәйкестендіріп жазыңыз:
eats(Who, What):- food(What), likes(Who, What).
pass_class(Who):- did_homework(Who), good_attendance(Who).
owns(Who, What):- bought(Who, What).
2. Берілген сөйлемдерден Visual Prolog ережелерін құрыңыз:
• адамның асқазаны бос болса, онда ол аш;
• егер жақсы ақы төленетін және көңілді жұмыс болса,онда ол барлығына ұнайды;
• автокөлікті кім сатып алса,онда сол басқарады.

Предикаттар

Прологтағы қатынас предикат деп аталады. Аргументтер – бұл осы қатынастар мен байланыстырылатын объекттер; айғақта likes(bill, cindy) қатынас; likes- бұл предикат, ал bill және cindy –аргументтер.

Әртүрлі сандардық аргументтерімен берілген бірнеше предикат мысалдары:

pred (integer, symbol)
person (last, first, gender)
run()
birthday (firstName, lastName, date)

Алдыңғы мысалда көрсетілгендей предикаттарда аргумент болмауы

мүмкін, бірақ ондай предикаттардың қолданылуы шекеулі. Mr. Rosemont атын анықтау үшін person(rosemont, male, male) сұрауларын қолдануы мүмкін. Бірақ сұрауларда run аргументінің дәлелдерінсіз не істейді? Программада run ұсынысы барма анықтайық,және егер run ереженің басы болса, онда берілген ережені есептеуге болады.Кей жағдайларда бұл пайдалы болады. Мысалы, егер сіз әртүрлі соған сәйкес жұмыс істейтін программа құрғыңыз келсе,онда бұл run -ның ұсынысында болады.

Айнымалылар

Қарапайым сұрауда,кімнің теннисті жақсы көретіндігін табу үшін айнымалылар қолданамыз.

Мысалы:

likes. (X, tennis)

Бұл сұрауда X әріпін белгісіз адамды табу үшін айнымалы ретінде қолданамыз. Айнымалының аты Visual Prolog кейін (бас немесе кіші) әріптердің кез келген саны тұра алатын (немесе астын сызу символы) цифрлар немесе асты сызулы символдардан басталуы керек. Дұрыс айнымалының аты төменде келтірілген:

My_first_correct_variable_name Sales_10_ll_86

келесі үш - терісі:

1stattempt

second_attempt

"Disaster "

Атауларын әртүрлі регистрдің айнымалы әріптерімен қолданған ыңғайлы:

IncoraeAndExpenditureAccount

Мағынасы түсінікті таңдау айнымалының атын оқу үшін программаны өте ыңғайлы істейді.

Мысалы:

likes(Person, tennis)

жақсы,оған қарағанда

likes(X,tennis).

өйткені Person x қа қарағанда көбірек мағына береді. Мақсатты енді сынап көріңіз:

GOAL likes (Person, tennis).

Visual Prolog жауап береді:

Pers on=ellen Person=mark

2 Solutions

Өйткені, мақсат екі шешім ellen және marktің мәндерімен айнымалы Personдарды дәйекті түрде салыстыра ала алады, атап айтқанда.

Айнымалылардың белгілеу

Сіз прологтың меншіктеу операторы бола алмайтындығын байқай алдыңыз. Бұл прологтың басқа программалау тілдерінен маңызды айырмашылықтары. Прологта айнымалысы, айғақтар немесе ережелердегі

тұрақтылармен салыстыруда аты-жөнін көрсетеді.

Белгілеуге дейін айнымалы еркін; иемденулерден кейін оның мәнімен байланыста болады. Егер сұраулар бойынша айнымалысы шешім қабылдаса, онда сабақтас болып қалады, содан соң ол прологты босатады және басқа шешімді іздейді.

Мәліметті айнымалың мәнін беріп сақтауға болмайды. Мәліметтің қоймасы емес, айнымалы үдерістің бір бөлігі ретінде шешімді іздестіру үшін қолданылады.

Ch02e03.pro -нің программасы қарап шыға алады, сонымен қатар айнымалы өз мәндерін алады.

```
predicates
likes(symbol, symbol)
clauses
likes(ellen, reading).
likes(John, computers).
likes(John, badminton).
likes(leonard, badminton).
likes(eric, swimming).
likes(eric, reading).
```

Сұрауларды қарап шығамыз: жүзу мен оқуды жақсы көретін адам бар ма?

```
likes(Person, reading), likes(Person, swimming).
```

Пролог программадағы айғақтарды басынан соңына дейін іздестіру арқылы сұраудың екі бөлігін де шеше бастады. Сұраудың бірінші бөлімі:

```
likes(Person, reading)
```

Person еркін айнымалы; пролог шешімді табар алдында оның мәні белгісіз. Басқа жағынан, екінші аргумент, reading атымен белгілі. Пролог бірінші сұраулардың бір бөлігіне сәйкес келетін айғақ іздейді.

Программадағы бірінші айғақ

```
likes (ellen, reading)
```

Демек пролог еркін Person айнымалысын ellenның мәнімен байланыстырады, (айғақтағы reading сұраулардағы readingге сәйкес келеді) бірінші сұраулардың бір бөлігін қанағаттандырады айғақты тиісті мәнге ұластырады. Дәл сол уақытта пролог көрсеткіші айғақтардың тізіміне бағытталады, ол іздеу процедурасының қаншалықты алға басқандығын көрсетеді.

Бұдан әрі, (жузу мен оқуды ұнататын адамға іздеу салу) сұрауларды толық шешу үшін екінші сұраудың бір бөлігі шешілуі керек. Person ellen мен байланысты болғандықтан,проглог айғақты іздеуі тиіс.

```
likes (ellen, swimming)
```

Пролог бұл айғақты программаның басынан іздейді,бірақ сәйкестік жоқ (өйткені программада мұндай айғақ жоқ). Егер Person ellen мәні болса

сұраудың екінші бөлімі жалған болады.

Енді пролог айнымалы Personдарды босатады және бірінші сұраулардың бір бөлігінің басқа шешімін табуға талаптанады. Басқа айғақты іздестіру бірінші бөлімнің сұранысын қанағаттандырады, (белгіленген позицияға қайта оралу, қайта іздеу деп аталады) айғақтардың тізіміндегі нұсқ ағыштан басталады.

Пролог оқуды жақсы көретін және likes(eric, reading) айғағын табатын адамды іздейді. Person айнымалысы қазір eric мәнімен байланысты, және пролог екінші бөлімнің сұранысы бойынша айғақтар программасында қайта іздеуді _____бастайды.
likes(eric, swimming)

Пролог сәйкестендірілген (программадағы соңғы ұсыныс) және сұраныс толығымен қанағаттандырылады.
Person=eric. 1 Solution.

Анонимді айнымалы

Анонимді айнымалылар біздің программаларымызды ретке келтіруге мүмкіндік береді. Егер сізге нақтылы мәлімет керек болса, анонимді айнымалыларды керек емес мәліметтерді жою үшін пайдалануға болады. Прологта анонимді айнымалыларастын сызу символы арқылы белгіленеді (_).

Келесі "отбасылық" мысал анонимді айнымалыларды пайдалану арқылы іске асады. TestGoalға ch02e04.pro программасының жобасын жүктеңіз.

predicates
male(symbol)
female(symbol)
parent(symbol, symbol)
clauses
male(bill).
male(]oe).
female(sue).
female(tammy).
parent(bill,joe).
parent(sue,joe).
parent(joe,tammy).

Аноним айнымалысы кез келген басқа өзгерістердің орынында қолданылады және оған ешқандай мағына берілмейді.

Мысалы, бізге келесі сұрауда қандай адамдар ата-ана бола алатындығын білу керек бірақ, сізге олардың балалары қызықты емес. Прологтағы әрбір сұраныста сізге сызылған символды падаланған кезде айнымалының мағынасы туралы ақпараттың қажеті жоқ.
goal

parent(Parent, _).

Мұндай сұранысты алғаннан кейін пролог былай деп жауап береді:
(Test Goal)
Parent=bill Parent=sue Parent=joe 3 Solutions

Бұл жағдайда пролог үш ата-ананы тауып береді, бірақ ол parent -тың ұсынысында екінші дәлелмен сабақтас мәндерді бермейді.

Анонимді айнымалыларды айғақтар ретінде пайдалануға болады. Прологтің келесі айғақтары:
owns(_, shoes). eats (_).

Табиғи тілде бекітулерді өрнек үшін қолдана алды:

Әрбір адамда аяқ киім бар. (Everyone owns shoes) Әрбір адам тамақтана алады. (Everyone eats)

Аноним айнымалысы кез келген мәліметтерге қарама-қарсы келеді.

Құрама мақсаттар**:** конъюнкция және дизъюнкция
Көріп отырғаныңыздай, құралған мақсаттарды іздеудің нәтижесі үшін қолдануға болады, екі ішкі мақсат А және В ақиқат (конъюнкция), ішкі мақсат үтір арқылы ажыратамыз. Егер іздеудің нәтижесінде ақиқат А және В ішкі мақсат (дизъюнкция) болса бұл ішкі мақсат нүктелі үтір арқылы ажыратамыз.

Төменде программаның мысалы көрсетілген:

```
predicates
car(symbol,long,integer,symbol,long)
truck(symbol,long,integer,symbol,long)
vehicle(symbol,long,integer,symbol,long)
clauses
car(Chrysler,130000,3,red,12000).
car(ford,90000,4,gray,25000).
car(datsun,8000,1,red,30000).
truck(ford,80000, 6,blue,8000).
truck(datsun,50000,5,orange, 20000) .
truck(toyota,25000,2,black,25000).

vehicle(Make,Odometer,Age,Color,Price):-car(Make,Odometer,Age,Color,Price);
truck(Make,Odometer,Age,Color,Price).
```

TestGoal-дың жобасына осы программаны жүктеңіз, содан соң мақсат қойыңыз:
Goal
car(Make, Odometer, Years_on_road, Body, 25000).

Берілген мақсат айтылған ұсыныстардан бағасы $25 000 тұратын машина (car) тұратын машина табуға тырысады. Пролог былай жауап береді:
Make=ford, Odometer=90000,
Years_on_road=4,

Body=gray
1 Solution.

Дегенмен берілген мақсат біршама жасанды. Жасанды болғандықтан тез арада мынандай типті сұрақ қойылады:
Тізімде $25000-дан кем тұратын машина бар ма?(Is there a car listed that costs less than $25000)

Мұндай мақсатты шешу үшін сіз Visual Prolog-та келесі құралған мақсатты бере аласыз:
Car (Make, Odometer, Years_on_road, Body, Cost) % ішкі мақсат А және Cost < 25000. % ішкі мақсат В.

Бұл конъюнкция болып табылады. Құралған мақсатты шешу үшін пролог ішкі мақсаттарды кезек бойынша рет-ретімен шешуге тырысады. Бастапқыда ол мынаны шешуге талаптанады:
Car (Make, Odometer, Years_on_road, Body, Cost), содан соң
Cost < 25000.
Cost айнымалысы екі подцельдерде ұқсас мәндерге ие. Енді осының барлығын Test Goalмен істеп көріңіздер.

Ескерту

Cost < 25 000 ішкі мақсаты Visual Prolog жүйесінде құралған «кем» қатынасына сәкес келеді. Кем қатынасы басқа да екі сандық объектілерге пайдаланылатын қатынастардан ешқандай айымашылығы жоқ, бірақ та, ол қатынастымына символмен (<) және де екі объектінің арасында қатыстыра отырып жазған дұрыс.

Қарастырсақ келесі өрнек ақиқат болып табыла ма? Табиғи тілде ол былай деп сипатталады: Тізімде $25 000-дан кем тұратын автомобиль немесе $20 000-дан кем тұратын жүк машинасы бар ма?

Мына тапсырмада келесі құралған мақсаттарда пролог тиісті нәтижені іздестіруді қайта орындайды:
Car (Make, Odometer, Years_on_road, Body, Cost), Cost<25000% ішкі мақсат А немесе
truck (Make, Odometer, Years_pn_road, Body, Cost), 20000<Cost. % ішкі мақсат В.

Құралған мақсаттағы бұл тип дизъюнкция болып табылады. Берілген мақсат екі альтернативті ішкі мақсаттарды, бір ережеге екі ұсыныстың біріктірілгеніне сәйкес келуі мүмкін келеді деп есептейді. Пролог екі ішкі мақсаттарды да қанағаттандыратын нәтижелердің барлығын іздейді.

Рұхсат етілген құралған мақсатта пролог келесі ішкі мақсаттардан құралған бірінші ішкі мақсаты іздеуге тырысады. ("автомобильді іздеу"):
car(Make, Odometer, Years_on_road, Body, Cost) және
Cost < 25000.

Егер автомобиль табылса мақсат-ақиқат,егер табылмаса(жүк машинасын іздеу) – пролог келесі ішкі мақсатан тұратын екінші құрама мақсаттарды шешуге тырысады:
truck(Make, Odometer, Years_on_road, Body, Cost),

және
Cost < 20000.

Visual Prolog программасы

Visual Prolog синтаксисі қасиеттер және өзара байланыстарды анықтау үшін жасалған. Көбінесе (айғақтар және ереже) ұсыныс, предикаттар, айнымалы және мақсаттарды қарастардыңыз.

Прологтың басқа болжамдардан айырмашылықтары, Visual Prolog - бақылаушы типтердің компиляторы: әрбір предикат үшін қолданылатын объекттердің типін хабарлайды. Сонымен бірге, Visual Prolog бағдарламаларына бұл типтерді орындайтын жылдамдығы машина кодтарына сәйкес келуге мүкіндік береді,ал басқа жағдайда - Pascalдың тілдеріндегі ұқсас бағдарламалардың жылдамдығынан асады.

Енді Visual Prolog программасының негізгі төрт бөлімін талқылаймыз – жарияланатын және сипатталатын предикаттар,сонымен қатар аргумент типтері ,берілген және анықталған ережелер программаның мақсаты болып табылады. Бұдан әрі ережелердің синтаксисін және хабарламаларды толығырақ қарап шығамыз . Соңында ,программаның басқа да бөлімдерін қысқаша қорытындылап сипаттаймыз: деректер қоры, тұрақтылар, әр түрлі глобалді бөлімдер және компилятордың нұсқауы.

Visual Prolog – програмасының негізгі бөлімдері:

Prolog Visual тіліндегі программа төрт негізгізгі программалық реттен тұрады. Оларға мыналар жатады:

• clauses (ұсыныстар) бөлімі;
• predicates (предикаттар) бөлімі;
• domains (домендер) бөлімі;
• goal (мақсаттар) бөлімі.

Clauses бөлімі- Visual Prolog программасының негізгі бөлігі; нақ осы бөлімде айғақтар және ережелер жазылады, Visual Prolog операциясы программаның мақсатын шешуге тырысады.

Predicates бөлімі бұл (Visual Prologта кірістірілген предикаттарды жарияламауға да болады) предикаттарды және домендердің (түрлер) аргументтері жарияланады.

Domains бөлімі сіздер қолданылатын барлық домендерді хабарлау үшін қызмет көрсетеді, олар (стандарты домендерді жариялау міндетті емес) Visual Prolog-тың стандартты домендері бола алмайды.

Goal бөлімі - бұл сізге Visual Prolog-программасының мақсатын сыйғызып алады.

Ұсыныстардың бөл***і***м***і***

Программаны құру барысында сіз clauses -тың (ұсыныстар) бөлімінде барлық айғақтар мен ережелерлерді сыйғыза аласыз. Негізгі тарауда программадағы ұсыныстарға(айғақтар және ережелер) басты назар аудырылады: олар қалай жазу керектігін білдіреді.

Егер сіз айғақтарды және ережелерді олардың прологта көрсетілуін түсінсеңіз, онда сіз clausestың бөліміндегі әрбір нақты предикат үшін барлық ұсыныс бірге орналасуы керек екендігін білесіз. Бір предикатты анықтайтын ұсыныстардың тізбегі процедура деп аталады.

(clauses бөліміндегі бірінші ұсыныстан бастап) Visual Prolog әрбір айғақты және әрбір ережені табуға ұмтылады. Төмен қарай clausestың бөлімі, ол бірінші ішкі нұсқауға ұсынысты шешімінің бір бөлігі ретінде орнатады. Егер келесі ұсыныс логикалық жолдың бөлігі болып табылмаса, онда Visual Prolog қойылған нұсқағышқа қайтарылады және кезекті салыстыруды (бұл процесс қайтарумен іздестіру деп аталады) нұсқағышқа орын алмастыра жақындай іздейді.

Предикаттардың бөл***і***м***і***

Егер сіз clausestың бөліміндегі Visual Prolog тіліндегі программаның меншігіндегі предикаттарды сипаттасаңыз, онда сіз predicatestің (предикаттар) бөлімінде жариялауыңыз керек; басқа жағдайда Visual Prolog түсінбейді. Сіз предикатты хабарлаудың нәтижесінде домендерге (түрлерге) бұл предикаттың аргументіне жататынын хабарлаңыз.

Visual Prolog кірістірілген (олардың жариялауға болмайды) предикаттардың толық жиынымен әкелінеді,анықтама кітабы олардың толық сипатын көрсетеді.

Предикаттар айғақтарға және ережелерге тапсырма береді. Барлық предикаттар predicates бөлімінде олардың аргумент типтерінің (домендер) нұсқауы есептелінеді.

Сіз айғақтарыңыз бен ережелеріңіздің жұмысын істейтін (дәлелдер) объектілер типін жарияласаңыз Visual Prolog жұмысының тиімділігі едәуір өседі.

Қолданбалы предикатты жариялау

Предикаттың жариялануы сол предикаттың атымен басталады,содан соң дөңгелек ашылатын жақша ішінде нөл немесе предикаттың үлкен доменді(тип) аргументі жалғасады:

arguraent_typel OptionalNamel, argument_type2 OptionalName2 predicateName ....., argument_typeN OptionalName3

үтір дәлелдің (түр)

Әрбір доменді аргументтен кейін,соңғы аргумент типінен кейін жабылатын жақша қойылады.Атап өтеміз, clausestың бөліміндегі ұсыныстан айырмашылығы декларация предикаты нүктемен аяқталмайды, предикаттың доменді аргументі стандартты домен немесе domaines бөлімінде жарияланған домен болады. Аргументтің атын OptionalNameK деп нұсқауға болады - бұл прогрмманың оқылуын жақсартады, компилятор аяққа басатындықтан оның орындалу жылдамдығы білінбейді.

Предикаттардың аттары

Предикаттың аты реттеле орналасқан әріптер, цифрлар және астын сызылған таңбалармен басталуы керек. Әріптердің регистрі ешқандай мағына бермейді, дегенмен сізге біз (прологтың басқа болжамы да бұған жол бермейді және болашақта Visual Prolog- - болжам 6, ол да бұған тыйым салады) бірақ біз сізге предикаттың атының бірінші әрпін бас әріппен жазуға кеңес бермейміз. Предикаттың аты 250 символымен шектеледі.

Домендерд***і***ң бөл***і***м***і***

Дәстүрлі прологта тек – терм типі бар. Visual Prologta доменді барлық аргументтің предикаты ретінде жариялаймыз.

Бірнеше мысалдарды қарап шығамық.

1. Осы мысал предикаттардың домендерін қалай құжаттауға көмектесетінін көрсетеді:

Франк - 45 жасқа толған еркек.

Стандартты домендерді қолана отырып , сіз тиісті домен предикатын қолдана аласыз:

person.(symbol, symbol, integer) .

Көп жағдайларда мұндай жарияланулар өте жақсы жұмыс істейді. Дегенмен сіз программа жазылғаннан кейін бірнеше айлардан соң түзеткіңіз келді дейік.Предикаттың ұқсас хабарламасы сізге ештеңе айтпауы мүмкін.Керісінше,төменде көрсетілгендей бұл предикаттың декларациясы осы предикаттың аргументтерін ретке келтіруге көмектеседі:

```
Domains
name,sex = symbol
age = integer predicates
person (name, sex, age)
```

Меншікті домендердің хабарлауының бас артықшылықтарының бірі, Visual Prolog типтердің қатесін зерттеп отыра алған, мысалы:

```
same_sex (X, Y):- person (X, Sex, _), person (Sex, Y, _).
```

name және sex symbol ретінде көрсетілгеніне қарамастан, олар бір біріне эквивалентті емес. Егер сіз оларды шатыстырсаңыз, Visual Prolog сол қатені анықтауға мүмкіндік береді.Бұл сіздің бағдарламаңыз өте үлкен және күрделі болған жағдайларда пайдалы болмақ.

```
Domains
product, sum = integer
predicates
add_em_up (sum, sum, sum)
multiply_em (product, product, product)
clauses
add_em_up (X, Y, Sum):-Sum=X+Y.
multiply_em (X, Y, Product):-Product=X*Y.
```

Бұл прогрмма екі операцияны орындайды: жинақтайы және көбейтеді.Оған мынадай мақсат беріп көрейік:

add_em_up (32, 54, Sum).
Visual Prolog (Test Goal) былай жауап береді:
Sum=86
1 Solution
сіз программаға екі бүтін санның қосындысын жібердіңіз.

Декларация және ережелер

Visual Prologта бірнеше кірістірілген стандартты домендер бар. Оларды domains бөлімінде декларарация типінде және сипатталмаған предикаттардың аргументінде қолдануға болады.

Негізгі домендер 4 кестеде сипатталған

Домен	Сипаттaлуы	Іске асуы
short	Қысқа, таңбалы, сандық	Барлық платформалар (- 32 768—32 767) 16 бит
ushort	Қысқа, таңбасыз,сандық	(0—65 535) 16 биттік платформалар
long	Ұзын, таңбалы, сандық	барлық платформалар (- 2 147 483 648-2 147 483 647) 32 бит
ulong	Ұзын, таңбасыз, сандық	барлық платформалар (0-4 294 967 295) 32 бит
integer	Таңбалы, сандық, тәуелді көлемді платформаға ие	(- 32 768-32 767) 16 биттік платформалар (- 2 147 483 648-2 147 483 647) 32 биттік платформалар
unsigned	Таңбасыз, сандық,тәуелді көлемді платформаға ие	(0—65 535) 16 биттік платформалар (0-4 294 967 295) 32 биттік платформалар
byte		Барлық платформалар (0— 55) 8 бит
word		Барлық платформалар (0—65 535) 16 бит
dword		Барлық платформалар (0—4 294 967 295) 32 бит

Идентификаторла және жолдар сіздің программаңызда бір-бірін ауыстыра алады, бірақта

Visual Prologта олар бөлек сақталады. Идентификаторлар, идентификатор кестелерінде сақталады, ал олардың көрінісіне оның кестедегі индексі ғана қолданылады, бірақ жолдардың идентификаторы емес. Бұл идентификаторлардың қарама-қарсы қойылуы өте жылдам орындалатындығын білдіреді, ал олар программада біренше жерде кездессе,ондо олардың сақталуы да тығыз болады. Жолдар іздеу кестесінде сақталынбайды және қарама-қарсы қоюдың міндеттілігінде Visual Prolog оның символдарын біртіндеп тесереді.керек болса нышанға олардың нышаны тексереді. Сіз қай домен әрбір прграммада жақсы қолданылатынын өзіңіз шешесіз.

Тапсырмалар

Visual Prolog аяқталған телефондық анықтама секілді программаларды көрсетеді. Тек стандартты домендер пайдаланылғандықтан оған domains бөлімінің программада қажеті жоқ.

```
Predicates
phone_number (symbol, symbol)
clauses
phone_number ("Albert " , "EZY-3665 " ).
phone_number ("Betty " , "555-5233 " ) .
phone_number ("Carol " , "909-1010 " ) .
phone_number ("Dorothy " , "438-8400 " ) .
goal
```

программаны енгізіп, орындауға жібергеннен кейін рет-ретімен мақсаттарды еңгізіңіз.

```
phone_number ("Carol " , Number) .
phone_number ("438-8400 " Who, ).
phone_number ("Albert " , Number).
phone_number (Who, Number).
```

Енді ұсыныстарды өзгертіңіз. Kim және dorothy бір телефон номеріне ие деп есептейік. Бұл айғақты clauses бөліміне енгізіп, келесі мақсатты енгіземіз:
Phone_number ("438-8400 " Who,).
Бұл сұраныстан сіз екі нәтиже алуыңыз керек:
Who=dorothy
Who=kim
2 Solutions.

Програмасындағы char доменін сипатау үшін isletter предикаты пайдаланылады. Тапсырама барысында оған төменлегідей мақсаттар қойылған:
isletter(%).
isletter(Q).
"Yes" немесе "No" сәйкес келетін мағынасына қарай қайтарады

```
Predicates
isletter(char)
clauses
= > белгісі % таңбасын анықтайды.
% мына теңдік "Алфавиттің алдында қойылады"
isletter (Ch):-‘ a’< = Ch, Ch<=’z’.
isletter (Ch ):-‘A’<= Ch, Ch<=’Z’
```

Программасын енгізіп, Test Goal -да әрбір мақсатты рет-ретімен сынап көріңіз:

a) isletter (‘x’). d) isletter (a) .
b) isletter (‘ 2 ‘). e) isletter (X).
c) isletter ("Hello ").
(c) және (d) мақсаттары қателіктің түріне алып келеді,ал (e) мақсаты және "Free variable " (байланыспаған айнымалы) хабарламасын қайтарады. Сіз берілмеген объектінің a немесе z-ке қатынасын тексере алмайсыз.

Бақылау сұрақтары

3. Логикалық программалау тілдерінің мүмкіндіктері.
4. Тілдің құрылымы

Есеп беру мазмұны және формасы: Тапсырмалар нұсқасын құрастыру. Тапсырманы орындау реті. Орындау нәтижелері. Қортынды.