Зертханалық жұмыс №7

AMZI! Prolog қабықшасымен таныстыру. Пролог тілінде программалаудың негізін үйрету. Деректер арасындағы қатынастар

Негізгі түсініктер

Пролог программалау тілі өзіне үлгілерді топтастыратын, бұтақтәрізді деректер құрылымын ұсынатын және автоматты түрде кері қайтарылатын мезанизмдердің шектеулі жиыныда топтастырылады. Осы шағын жинақтың өзі қуатты, әрі икемді программалық аппаратты құрайды.

Пролог тілінде программалау – логика терминдерінде программалау болып табылады. Прологтағы программалар декларативті стильде жазылады, бұл дегеніміз – Пролог – машинамен программаларды орындау келесі принцип бойынша жүргізіледі:

Сен айт: " Не істеу керек", мен жасаймын "Қалай істеу керектігін". Басқа сөзбен айтқанда, программалаудың императивті тілдерімен салыстырғанда, Прологта программалардың орындалу ретін интерпретататордың өзі таңдайды.

Әсіресе, Пролог объектілер мен олардың арасындағы қатынастар басты орында болатын есептерді шешуге жақсы ыңғайланған. Мұндай есептерге қатынастарды сипаттайтын есептер мысал бола алады. Прологта (X,Y) негізгі болатын қатынас сияқты, қатынастарды анықтау оңай, егер X негізгі болса РАС, әйтпесе Y негізгі болса ЖАЛҒАН.

Пролог тіліндегі программа сөйлемдерден тұрады. Әр сөйлем нүктемен аяқталады.

Қатынастардың аргументтері атомдар (тұрақты объектілер) немесе айнымалылар бола алады. Программа мәтінінде айнымалылардың атаулары бас әріптерден, ал тұрақтылардың атаулары кіші әріптерден баьталады.

Прологта қатынастар нақты деректер немесе ережелер түрінде жазылуы мүмкін. Нақты деректер түрінде қарапайым қатынастар жазылады.

Мысалы:

human(oleg).
fruit(orange).

Ережелер белгілі қатынастардан жаңа қатынастар шығару үшін қажет.

Мысалы:

father(X,Y):-parent(X,Y),man(X).

X Y-тің әкесі болып табылады, ЕГЕР X Y –тің ата-анасы және X еркек болса.

Жүйеге қойылатын сұрақтар айнымалылардан тұратын мақсаттар түрінде құрылады. Бұл жағдайда, Пролог – машина мүмкін болатын барлық айнымалылар жағдайларын тексереді және солардың ішінен мақсатқа жететінін тауып алады.

Жаңа деректерге әкелетін ережеге нақты деректерді қайталап қолдану, тікелей талқылау (*forward chaining*) деп аталады. Кері талқылау (*backward chaining*) кезінде, мақсатқа сәйкес қорытындыларды (ережелер тақырыбын) іздеу жүргізіледі. Егер ондай қорытынды табылса, онда өз кезегінде берілген ережелерге сәйкес мақсаттар дәлелденуі керек. Прологта кері талқылау қолданылады.

Прологта іздеу тереңнен жүргізіледі. Іздеу алгоритмі 2 суретте көрсетілген.

Мұнда үш тізім қатыстырылған. Біріншісі, ұсыныстардан тұратын ҰСЫНЫСТАР тізімі. Екіншісінде, қанағаттандырылатын мақсаттар МАҚСАТТАР тізімі. Үшінші тізім, ШЕШІМДЕР, ол мақсатқа жету үшін қолданылатын қайту нүктелері мен ұсыныстардың жолын сақтайды. Бұл процедура шешімді іздеу жолының трассасы сияқты.

ІЗДЕУ процедурасы мақсаттар тізімін қарастырады. Егер ол бос болса, дәлелдеуді қажет ететін бір де бір мақсаттың қалмады және іздеу сәтті деп есептеледі. Әйтпесе, МАҚСАТТАР тізіміндегі бірінші мақсат жойылады да, сканерлеу сәйкес ұсынысты ізздеу мақсатында ҰСЫНЫСТАР тізімі бойынша жүргізіледі.

Егер ондай ұсыныс табылмаса (ҰСЫНЫСТАР тізімінде), онда осы ұсынысқа нұсқау (ҰСЫНЫСТАР тізіміндегі), мақсатымен қоса ШЕШІМДЕР тізіміне барып қосылады. Көрсеткіш керекті сөйлемді табу алдында ҰСЫНЫС тізімінде ІЗДЕУ процедурасы қаншама алға жылжығанын белгілейді. Ары қарай таңдалған ұсыныстың мақсаттары тексеріледі. Егер олардың арасында тым біреуі ғана орындалмаса, көрсеткіш ҰСЫНЫС тізімдегі келесі ыңғайлы ұсынысқа дейін жылжыйды және оның мақсаттары МАҚСАТ тізіміне енгізіледі. Сәтсіздікке мұндай әрекет кайтарым деп аталады. ШЕШІМДЕР тізімінде ескі көрсеткіш жаңасына ауысады және кайтарым орындалған сайын ҰСЫНЫСТАР тізімі бойынша ары қарай жылжыйды. ҰСЫНЫСТАР тізімін соңына көрсеткіштің жеткені берілген мақсаттарды орындайтын ұсыныс жоқтығын білдіреді.