

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС
ТАЪЛИМ ВАЗИРЛИГИ**

БУХОРО МУҲАНДИСЛИК-ТЕХНОЛОГИЯ ИНСТИТУТИ

**“ИНФОРМАТИКА ВА АХБОРОТ ТЕХНОЛОГИЯЛАРИ”
КАФЕДРАСИ**

“ТИЗИМЛИ ДАСТУРИЙ ТАЪМИНОТ” ФАНИДАН

КУРС ИШИ

Мавзу: Хомский грамматикаси иерархияси

Бажарди:

**12-11 МИАТ гуруҳ талабаси
Бахронова Гулираъно**

Қабул қилди:

Ибрагимов У.М.

Бухоро-2013 й.

**БУХОРО ЮҚОРИ ТЕХНОЛОГИЯЛАР МУҲАНДИСЛИК-
ТЕХНИКА ИНСТИТУТИ**

**“ТЕХНОЛОГИК ЖАРАЁНЛАРНИ АВТОМАТЛАШТИРИШ”
ФАКУЛЬТЕТИ**

**“ИНФОРМАТИКА ВА АХБОРОТ ТЕХНОЛОГИЯЛАРИ”
КАФЕДРАСИ**

ТАСДИҚЛАЙМАН
Кафедра мудири
Доц. Раззоқов Ш.И.

« ____ » _____ 2013 й.

(Талабанинг гуруҳи ва Фамилияси, исми ҳамда шарифи)

Фанидан

ТОПШИРИҚ

Мавзу: _____

1. Курс ишининг топшириш учун муддати _____

2. Курс ишини бажариш учун бошланғич маълумот _____

3. Тушунтириш хатининг мазмуни _____

4. График материаллар: _____

5. Топшириқ берилган сана: _____

6. Топшириқни бажариш учун қабул қилдим: _____

7. Рахбар: _____

Мундарижа

Кириш

«Тизимли дастурий таъминот» фани Ўзбекистон Республикаси Президенти И.А. Каримов томонидан қўйилган малакали мутахассисларни тайёрлаш миллий таълим тизимини ривожлантириш бўйича муҳим масалаларни ечишда катта аҳамият касб этади.

Мазкур фан замонавий электрон ҳисоблаш машиналарининг тизимли дастурий таъминотини ўрганган ҳолда, уни жамиятнинг турли соҳаларида самарали қўллаш муаммолари билан шуғулланади.

Фан таркибида ҳисоблаш тизимининг дастурий таъминоти, уни ташкил этиш, бошқариш асослари ва ундан амалиётда фойдаланиш масалаларини ўрганиш кўзда тутилган.

Тизимли дастурий таъминот таркибига ҳисоблаш тизимини бошқарувчи дастурлар, хизматчи дастурлар, дастурлаш тизимлари ва операцион тизимлар киради. Ҳозирги вақтга келиб эса, ушбу юқорида келтирилган дастурларни ўз соҳасида қўлламаётган мутахассисни топиш мумкин эмас. Демак, ушбу дастурлардан фойдаланиш кўникмасига эга бўлиш нафақат ушбу соҳада хизмат қилаётган мутахассислар учун, балки, жамиятимиздаги барча соҳа мутахассислари учун зарур ҳисобланади.

G граматикадан ҳосил бўлувчи тил бу граматиканинг бошланғич рамзидан чиқувчи терминал занжирлар тўпламига айтилади. Ёки формал кўринишда, $L(G)=\{\alpha \in T^* \mid S \rightarrow_G^* \alpha\}$

Шундай қилиб, тилнинг ихтиёрий занжирини чиқаришни бошланғич нотерминал рамздан бошлашимиз керак. Агар грамматика қоидаларига кўра ихтиёрий чекли ўрнига қўйишдан сўнг ҳосил бўлган занжир фақат терминал рамзлардан ташкил топган бўлса, бу занжир тилнинг грамматикасидан ҳосил бўлган занжир ҳисобланади. Фақат терминаль, яқуний рамзлар тил жанжирида учраши мумкин. Нотерминал – бу қўшимча, ёрдамчи рамзлар бўлиб, тилни чекли қоидалар билан бериш учун керак.

Масалан, bbb занжири тилга тегишли бўлиб, G_0 граматикадан ҳосил бўлган. Ҳақиқатдан ҳам: $S \rightarrow aSbAc \rightarrow bbAc bAc \rightarrow bBbAc \rightarrow bBB \rightarrow bbB \rightarrow bbb$.

Хомскининг ҳосил қилувчи грамматикаси тилларни формал бериш учун хизмат қилади ва грамматика берилса, бу грамматикадан ҳосил бўлувчи турли занжирларни олиш мумкин. Амалиётда эса тескари масала учрайди: тил грамматикасини тилнинг “тўғри” занжирлари мисоллари сони ва баъзи тил конструкцияларини тўғрилиги ҳақида билиш асосида тузиш. Аниқки, тилнинг барча занжирларини бериб бўлмайди – улар сони чексиз. Шунинг учун граматикани яратиш – ноформал иш бўлгани учун, уни яратишни мисоллар орқали ўрганиш мумкин.

Иккита грамматика эквивалент ҳисобланади, агар улар бир тилни ҳосил қилсалар.

Мисол: $V=\{a,b,c,\dots,0,1,\dots,[,],\dots,\text{begin},\text{end},\text{for},\dots\}$

$L=\text{Паскаль тили}$

Замонавий юқори даражадаги дастурлаш тиллари ҳосил қилувчи Хомский грамматикаси орқали берилади. Бу грамматикалар бир неча ўнлаб (баъзан минглаб) қоидалардан иборат. Қулайлик ва тушунарлилик учун бу қоидалар янада қулай воситалар-синтаксис диаграммалар (Паскаль тилидай) ёки кенгайтирилган БНФ формаларда берилади. Тилларнинг беришнинг иккала модели ҳам қуйида кўриб ўтамиз, у ернинг ўзида бу моделлар Хомский ҳосил қилувчи грамматикасига эквивалент эканлиги исботланади.

Дастурлаш тили грамматикаси терминал рамзлари бу дастур ташкил топган рамзлар ҳисобланади. Дастурлаш тиллари грамматикаси ҳосил қилувчи нотерминал рамзлар табиати қандай экан?

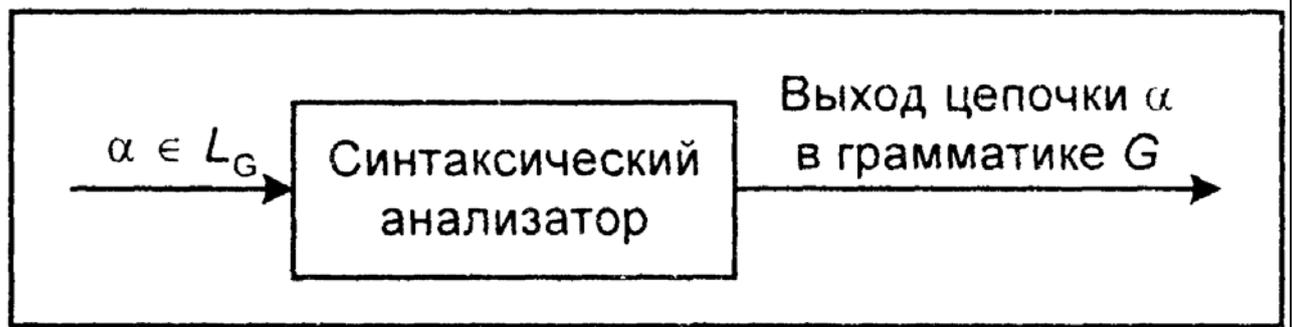
Аниқки, нотерминал рамзлар дастурлаш тилларини изоҳлаш учун ишлатилиши керак, лекин тил дастурларида учрамайди. Булар қанақа рамзлар? Аниқки улар тил конструкцияси номланишини беришлари керак. Дастурлаш тилларида бу конструкцияларин қуйидагича номлаш мумкин: ДАСТУР, БЛОК, ОПЕРАТОР ва ҳоказо. ДАСТУР нотерминал албатта бошланғич нотерминал ҳисобланади, бу энг абстракт конструкция ҳисобланади. Худди мана шу нотерминалдан ихтиёрий дастур чиқарилади. Грамматика қоидаларида нотерминал рамзлар терминал рамзлар билан адаштириб юбормаслик учун ажратилади, маслан, хизматчи сўзлар билан. Баъзан улар бурчакли кавсларга олинади, масалан:

$\langle \text{ИДЕНТИФИКАТОР} \rangle \rightarrow \langle \text{ҲАРФ} \rangle | \langle \text{ИДЕНТИФИКАТОР} \rangle \langle \text{ҲАРФ} \rangle | \langle \text{ИДЕНТИФИКАТОР} \rangle \langle \text{РАҚАМ} \rangle$

Баъзан нотерминаллар кавсиз ёзилади, терминал рамзлар эса қўштирноққа олинади, масалан:

$\text{expression} \rightarrow \text{expression} \text{ ‘?’ } \text{expression} \text{ ‘:’ } \text{expression}$

Оддий мисол сифатида содда дастурлаш тили учун ҳосил қилувчи грамматика яратишни кўрамиз, қайсики бу тилни Милан (инг. Тилда – Mini language). Келинги, Милан тилида дастур қуйидагича ёзилсин:



2-расм. Синтаксис анализ масаласи.

Аниққки агар синтаксис анализнинг умумий алгоритми мавжуд бўлса, у Тьюринг машинаси орқали амалга оширилиши мумкин. Аммо ундай алгоритм учун Тьюринг машинаси ихтиёрий кириш лентасида бу машина тўхташини (ёки якуний ҳолатда ёки продукцияни қўллай олмаслик ҳолатида) ва алгоритм иш вақтини баҳолай олмаймиз.

Шундай қилиб, умумий ҳолда синтаксис анализ масаласини Хомский ҳосил қилувчи грамматика модели синфида эффектив еча олмаймиз.

Бундай ҳолатда тилларни трансляциялаш бу қарашини кўриб ўтишни тугатишимиз мумкин эди, чунки бу фактдан умумий назарияни ярата олмаслик ва ҳар бир ҳолат учун индивидуал синтаксис анализ алгоритмини яратиш керак эди. Бироқ, илмда кўпинча бўладигандек умумий кўринишда ечиб бўлмас ёки қийин ечимга эга муаммолар хусусий берилган ҳолатларда эффектив ечилади. Бундан ташқари бу муаммоларнинг хусусий берилиши амалиётда муҳим ҳолатларнинг кўпгина юзага келишини қоплайди.

Формал тиллар ва грамматика назариясида худди мана шу ҳолат мавжуд. Трансляциялаш методларида муҳим рольни хусусий ҳолатлар ўйнайди, улар Хомский умумий грамматика ичкисинфлари ва хусусан таркибан-муस्ताқил грамматикалар. Уларнинг қоидалари Хомский умумий грамматикаси кўриниши қоидаларига нисбатан содда.

Аниққки, грамматика қоидалари формаси (уларнинг сони эмас) синтаксис анализ алгоритми учун муҳим аҳамиятга эга. $\alpha \rightarrow \beta$ кўринишидаги қоидада чап ва ўнг қисмлар қандай кўринишга эга эканлигага қараб Хомский 4 тип(оила)ни аниқлади. Хомский бу грамматикалардан ҳосил бўлувчи 4 та турли тиллар оиласини айратди ва ҳар бир оила учун натижада абстракт ўтказувчилар (автоматлар) синфи топилди, яъни шу синф берилган α занжири ва G грамматикасига кўра G да чиқиш занжирини топиш. Берилган грамматика ички синфи қоидаларига қанча чегаралаш кўп қўйилса бу синф учун синтаксис анализ алгоритми шунча эффектив бўлади. 1-жадвалда Хомский 4 типидagi грамматикаси келтирилган ва шу 4 тип грамматика учун мумкин йўл қўйиладиган қоидала, бу синфлар номи ва улар келтириб чиқарадиган тиллар ҳамда умумий таниб олиш алгоритми амалга оширилиши мумкин бўлган абстракт қурилма келтирилган.

Тип	Қоидалар кўриниши	Грамматика номи	Таниб олувчи абстракт қурилма
0	$\alpha \rightarrow \beta$	Чекланмаган грамматика Мустақил (free) грамматикалар	Тьюринг машинаси
1	$\gamma A \delta \rightarrow \gamma \beta \delta$	Таркибан боғлиқ грамматикалар Қисқармайдиган граматиклар Тўғридан тўғри иборат бўлган грамматика ТИ-грамматика	Чизиқли чекланган автоматлар
2	$A \rightarrow \beta$	Таркибан мустақил грамматика ТМ грамамтика	Дўконли хотирали автоматлар
3	$A \rightarrow aB$ $A \rightarrow a$ $A \rightarrow \epsilon$	Автомат грамматика А грамматика Регуляр грамматика Чекли сонли ҳолатдаги грамматика	Чекли автоматлар

1-жадвал.

Формат тилларни формал грамматикасини бериш учун муҳим тил керак. Бундай тил метатил деб аталади. Бу ТМ грамматикани қоидаларини қулай ифодалаш учун келишув ҳисобланади. Бу қоидаларни албатта бу тилнинг иборалари сифатида тўғридан тўғри ишлатиш мумкин, лекин кўпинча нотерминал рамзлар улар билан боғлиқ маънога эга ва грамматикада нетерминал рамзлар учун абстракт белгилашларни ишлатиш эмас мос конструкцияларни ишлатиш қулай.

Тарихан ТМ грамматикани ифодалаш учун биринчи метатил Бекас-Наура нотацияси ёки БНФ нотацияси ҳисобланади. Бу ерда нотерминал рамзлар бурчак кавсларга ёзилади, ::= рамзи стрелка ўрнига ишлатилади. Худди шу ерда қисқартириш мақсадида қоиданинг ўнг томонида | белгиси билан у ёки бу нотерминалнинг ўзаро алмаштиргичлари киритилди. Бу белги “ёки2 маъносини беради.

Масалан, Турбо паскал тилида ўзгарувчиларни эълон қилиш синтаксис қуйидагича бўлади:

```
<variable-declaration-part> ::= var <variable-declaration>
{; <variable-declaration>}
```

Амалий қисм

```

program leks;
label 10,20,30,40;
const
  tts:array[1..17] of string[7]=
    ('program','input','output','var','end','begin','while','do','char',
     ':',';','_',':=','<','>', '(' ,'' );
  r:set of char=[':','<','>', ';', ',', ':', '=', '(', ')', ',', '"', '+', '-'];
  c:set of char=['0','1','2','3','4','5','6','7','8','9'];
type ms=array[1..100]of string[10];
   str=string[15];
var
  f,ff:text;
  tl ,tsi:ms; kod:array[1..100,1..2]of integer;
  i,nm,k,l,t:integer;
  sl,name:str;st:string[100];
PROCEDURE poisk(s:str;lim,nt:integer;var nom:integer);
var z:integer;
begin nom:=0;
if nt=1 then
  for z:=1 to lim do if tts[z]=s then nom:=z;
if nt=2 then for z:=1 to lim do if tsi[z]=s then nom:=z;
if nt=3 then for z:=1 to lim do if tl[z]=s then nom:=z;
end;
PROCEDURE zapis(nt,nom:integer);
begin
  kod[k,1]:=nt;kod[k,2]:=nom;
  inc(k);sl:="";
end;
PROCEDURE output(tabl:ms;lim:integer);
var z:integer;
begin for z:=1 to lim do begin
  if z<>lim then write(ff,'<z,',' ,tabl[z],> ');
  else begin write(ff,'<z,',' ,tabl[z],>');writeln(ff) end end;
end;
BEGIN
writeln('ÿÿ. Ë¬п д ©« ');
readln(name);
assign(f,name);
reset(f);
writeln('ÿÿĬĬĬ Ë¬п д ©« ¼«п аĬ §Ĭ«мв в®ÿ');
readln(name);
assign(ff,name);rewrite(ff);
l:=0;k:=1;t:=0;sl:="";
while not eof(f) do BEGIN
  readln(f,st);
  i:=1;
  sl:="";

```

```

20: while (i<=length(st))and (not (st[i] in r)) do begin
  sl:=sl+st[i];inc(i);end;
  if st[i]="'" then begin
    if sl<>" then begin
      poisk(sl,17,1,nm);if nm<>0 then zapis(1,nm)
        else begin poisk(sl,t,2,nm);
          if nm<>0 then zapis(2,nm)
            else begin inc(i);tsi[i]:=sl;zapis(2,t);
              end;end;end;
poisk("'",17,1,nm);zapis(1,nm);inc(i);
  while st[i]<>"'" do begin
    sl:=sl+st[i];inc(i) end;
    if length(sl)>10 then begin
writeln(ff,'Т
  writeln(ff,sl,' Ү««МИГ 10');close(ff);goto 40;end;
  poisk(sl,1,3,nm);
  if nm<>0 then zapis(3,nm)
    else begin inc(l);tl[l]:=sl;zapis(3,1);end;
poisk("'",17,1,nm);zapis(1,nm);inc(i);goto 30;
end;
if (st[i] in r) and (sl=") then goto 30;
if sl[1] in c then begin poisk(sl,1,3,nm);if nm<>0 then zapis(3,nm)
else begin inc(l);tl[l]:=sl;zapis(3,1);end;goto 30;end;
poisk(sl,17,1,nm);
if nm<>0 then zapis(1,nm)
else begin poisk(sl,t,2,nm);
  if nm<>0 then zapis(2,nm)
    else begin inc(t);tsi[t]:=sl;zapis(2,t);end;end;
30: if i>length(st) then goto 10;
  if st[i] in r then begin
    if (st[i]=':') and (st[i+1]='=')then begin sl:='=';i:=i+2;end;
    if (st[i]='<') and (st[i+1]='>')then begin sl:='<>';i:=i+2; end
    else begin if (st[i]<>' ')and(st[i]<>"')then begin sl:=st[i];inc(i);end
      else begin inc(i);sl:=";goto 20; end;end;
  poisk(sl,17,1,nm);if nm<>0 then zapis(1,nm)
    else begin poisk(sl,t,2,nm);
      if nm<>0 then zapis(2,nm)
        else begin inc(t);tsi[t]:=sl;zapis(2,t);end;end;
  if i>length(st) then goto 10 else goto 20;end;
10:END;
close(f);
writeln(ff,'В Ү«.''' :');
writeln(ff,'<N бва.,бГa-.бЭ-Ү«>');
for i:= 1 to 17 do begin
  if i mod 5=0 then writeln(ff);
  write(ff,'< 'i',' ',tts[i],>');
  if i<>17 then write (ff,' ; ')end;
writeln(ff);

```

```
writeln(ff,'В Ÿ«.'€:');
writeln(ff,'<N бВа.,Ë¬Π>');
output(tsi,t);
writeln(ff);
writeln(ff,'В Ÿ«.'<');
writeln(ff,'<N бВа.,«ËБΓа <>');
output(tl,l);
writeln(ff);
writeln(ff,'ЉТ,,> <...Љ‘...Н :');
for i:=1 to k-1 do begin
  if i mod 5=0 then writeln(ff);
  if i<>k-1 then
    write(ff,'<',kod[i,1],',',kod[i,2],> ; ')
  else write(ff,'<',kod[i,1],',',kod[i,2],>');
end;
close(ff);
40:END.□
```

Хулоса

Ушбу “мавзу” курс ишини бажариб Тизимли дастурий таъминоти фанидан катта таъсуротларга ва натижаларга эришдим.

Адабиётлар:

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. -СПб: Питер, 2003.-396 с.
2. Афанасьев А.Н. Формальные языки и грамматики.: Учебная школа: УлГТУ. 1997.- 84 с.
3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -. Мир, 1979.- 487с.
4. Браун С. Операционная система UNIX - М.: Мир. 1986.-463 с.
5. Вендров А.М. Case - технология. Современные методы и средства проектирования информационных систем. -М: Финансы и статистика, 1998 -361с
6. Вирт И. Алгоритмы и структуры данных - М.; МИР. 1989. - 360 с.
7. Гордеев А.З., Молчанов А.Д. Системное программное обеспечение. -Спб-Петер. 2002.-734с.
8. Дворогин А.И. Основы трансляции. Учебное пособие. — Волгоград. ВолГТУЛ 999.80г.