

**O'zbekiston Respublikasi aloqa, axborotlashtirish va
telekommunikatsiya texnologiyalari davlat qo'mitasi**

Toshkent axborot texnologiyalari universiteti

Samarqand filiali

Axborot va pedagogik texnologiyalar fakulteti

Informatika va axborot texnologiyalari kafedrası

KURS ISHI

**Mavzu: Sintaktik tahlilning CY-tarjima tuzilmalarini
tasvirlash**

Bajaruvchi: 301 guruh talabasi

_____Maxkulov J.

Kurs ishi rahbari:

_____Kat. o'qit. Murodov O'.

Samarqand – 2013

Мавзу: Синтактик таҳлилнинг СУ-таржима тузилмаларини тасвирлаш.

Режа: 1. Ишнинг мақсади.

2. Синтактик анализаторнинг вазифаси.

3. Лексемлар ва уларнинг чегараларини аниқлаш муаммолари.

4. Лексем жадвали ва унда мавжуд ахборотлар таҳлили.

5. Синтактик таҳлилнинг таржима тутузилмалари.

6. Лексем турлари модули.

7. Синтактик анализаторларни (сканерларни) куриш.

8. Ишни бажаришга қўйилган талаблар.

9. Топшириқнинг ечим босқичлари.

9.1. Топшириқ учун кирувчи тил грамматикаси.

9.2. Топшириқдаги кирувчи тилнинг лексемасини танлаш учун ўзакли автомат (ЎА) ни тавсифлаш.

10. Хулоса.

11. Фойдаланилган адабиётлар.

1. Ишнинг мақсади.

Регуляр грамматика назариясининг асосий тушунчаларини ўрганиш, синтактик анализаторларнинг (сканерларнинг) ишлаш кўрсатмалари ва жараёнлари билан танишиш, берилган оддий киритиш тили мисолида берилган сканерни куришнинг амалий кўникмаларига эга бўлиш.

2. Синтактик анализаторнинг вазифаси.

Синтактик анализатор (ёки сканер) – бу дастур литерасини бажарилувчи тилда ўқийдиган ва улардан бажарилувчи тил сўзлари (лексемларни) курадиган компилятор қисмидир. Синтактик анализаторга бажарилувчи дастур матни қабул қилинади, чиқувчи ахборот эса синтаксис анализ ва танлаш босқичида компиляторда қайта ишлаш учун узатилади.

Лексем (тилнинг синтактик бирлиги) – тилнинг структурали бирлиги бўлиб, у тилнинг элементар белгиларидан иборат ва ўз ичига тилнинг бошқа структура бирликларини қабул қилмайди. Дастурлаштириш тилларининг лексемлари идентификаторлар, ўзгармаслар, тилнинг калит сўзлари, амал белгилари ва б. ҳисобланади. Аниқ дастурлаштириш тили лексемси таркиби тилнинг синтаксиси билан аниқланади.

Назарий нуқтаи назардан синтактик анализатор компиляторнинг зарур, керакли қисми ҳисобланмайди. Унинг вазифалари синтаксис анализатор босқичларида бажарилиши мумкин. Аммо бир нечта шундай сабаблар бўлиши мумкинки, улардан келиб чикқан ҳолда барча компиляторлар таркибига синтактик анализаторларни қўшиш мумкин. Булар қуйидаги сабаблар:

- синтактик анализатор бажарилувчи дастурга қабул қилинувчи матнни тартиблаштирганлиги ва барча кераксиз ахборотларни ўчирганлиги туфайли синтаксис танлаш босқичида бажарилувчи дастурнинг матни билан ишлаш енгиллашади ва қайта ишланувчи ахборот ҳажми қисқаради;
- матнни ажратиш ва лексемни танлаш учун оддий, самарали ва назарий жиҳатдан яхши қурилган анализ техникасини қўллаш мумкин, бу вақтда эса бажарилувчи тилнинг конструкцияларининг синтаксис босқичида эса танлашнинг бир мунча мураккаб алгоритмлари ишлатилади;
- синтактик анализатор конструкцияси бўйича мураккаб синтаксис анализаторнинг структураси кирувчи тил версиясига боғлиқ равишда қўшиладиган бажарилувчи дастур матни ишидан ажратилади;
- компиляторнинг бундай конструкциясида бир тил версиясидан бошқа тилга ўтишда оддий синтактик анализаторни қуриш мумкин.

Синтактик анализаторда бажариладиган функциялар ва бажариладиган дастур матнида ажратиладиган лексем таркиби компилятор версиясига

боғлиқ ҳолда ўзгариши мумкин. Асосан, синтактик анализаторлар бажарилувчи дастур матнидан изохлар ва белгиланмаган пробеллар, бундан ташқари қуйидаги тур лексемларини ажратадилар: идентификаторлар, сатрли, белгили ва сонли ўзгармаслар, амал белгилари, ажратувчилар ва кирувчи тилнинг калит (хизматчи) сўзлари бажарилади.

Компиляторларнинг кўпчилигида синтактик ва синтаксис анализаторлар – бу ўзаро боғланган қисмлардир. Синтактик ва синтаксис анализнинг қайси қисмида чегара ўтказиш мумкинлигини, қайси конструкцияларни сканер билан, қайсиларини синтаксис анализатор билан анализ қилишни компилятор яратувчиси шуғулланади. Қоидага кўра, агар у бажарилса, ихтиёрий анализ кирувчи дастурнинг синтактик танлаш босқичида бажарилишга ҳаракат қилади. Синтактик анализатор имкониятлари синтаксис анализатор имкониятларига кўра чегараланган, чунки унинг асосида бир мунча оддий механизмлар ётади.

Компиляторда синтактик анализатор роли ва унинг синтактик анализатор билан ўзаро алоқаси ҳақида [1-4, 7] да батафсил кўриб чиқиш мумкин.

3. Лексемлар ва уларнинг чегараларини аниқлаш муаммолари.

Оддий ҳолда синтактик ва синтаксис анализ фазалари компиляторда кетма-кет бажарилиши мумкин. Аммо кўпгина дастурлаштириш тиллари учун синтактик анализ босқичида ахборотлар бир қийматли турни аниқлаш ва навбатдаги лексем чегараси учун етарли бўлмаслиги мумкин.

Бу ҳолни тасвирлаш учун Фортран тилидаги дастур операторини кўрсатиш мумкин, унда *DO 10 I=1...* матн қисмида оператор турини (мос равишда, лексем чегараларини ҳам) аниқлаш мумкин эмас. *DO 10 I=1.15* ҳолда у *D010I* бутун ўзгарувчига 1.15 ўзгармас қиймати билан қўшилади (Фортранда пробеллар ташлаб юборилади), *DO 10 I=1.15* ҳолда эса бу 10 белгигача *I* бутун сонли ўзгарувчи бўйича 1 дан 15 гача ҳисобланади.

Янада замонавий C⁺⁺ дастурлаштириш тилида тасвирланиши – фақат битта тўғри интерпретацияси (агар операцияни бўшлиқлар билан ажратганда): $k = i++ + ++j$; бўлган $k=i+++++j$: қўшиш операторига эга бўлади.

Агар лексем чегараларини аниқлашнинг иложи бўлмаса, у ҳолда бажарилувчи дастурнинг синтактик анализи босқичма – босқич бажарилади. У ҳолда синтактик ва синтаксис анализаторлар параллел равишда, бир – бирига навбати билан мурожаат қилиб бажарилиши керак. Синтактик анализатор навбатдаги лексемни топиб, уни синтаксис анализаторга беради, у бажарилган дастурнинг ўқилган қисмини анализ қилишга ҳаракат қилади ва, ёки синтактик анализатордан кейинги лексемни сўраши ёки унга бир неча қадам орқага қайтишини талаб қилиши ва лексемларни бошқа чегаралар билан ажратишга ҳаракат қилиши мумкин. Бундан кейин эса у қайси лексемни кутиши ҳақида хабар бериши мумкин. Синтактик ва синтаксис анализаторлар орасидаги ўзаро алоқа схемаси махсус адабиётларда кенг берилган.

Кўриниб турибдики, синтактик ва синтаксис анализаторларнинг параллел ишлашини ташкиллаштириш кетма – кет бажарилишига кўра бир мунча мураккаброқ. Бундан ташқари, бундай ёндашув кўплаб ҳисоблаш ресурсларини ва бажарилувчи коднинг ўқилган қисмига қайтишига ҳамда такрорий анализга рухсат берганлиги учун бажарилувчи дастур анализага кўп вақтни талаб этади. Шундан келиб чиқиб, баъзи бир дастурлаштириш тилларида синтаксиснинг мураккаблиги худди шундай ёндашишни талаб этади – юқорида кўриб ўтилган Фортран тилидаги дастур бошқача анализ қилиниши мумкин эмас.

Синтактик ва синтаксис анализаторларнинг биргаликда ишлашидан қочиш учун дастурчилар кўпинча кирувчи тил синтаксисларига чекловлар қўядилар. Масалан, C⁺⁺ дастурлаштириш тили учун лексем чегарасини аниқлашда муаммо келиб чиқса, максимал узунликдаги лексемни танлашга келишиб олинган.

Юқорида кўрилган мисолни $k=i+++++j$ оператор учун кўрганимизда, куйидагига келамиз, лексемнинг иккита вариантларидан бири $+$ тўртинчи белгисини ўқишда ($+$ - C^{++} да қўшиш белгиси, $++$ - эса инкремент оператори) синтактик анализатор энг узун $-++$ (инкремент оператори) ни танлайди ва оператор тўлалигича $k = i++ ++ +j$; каби (операция белгилари бўлиқлар билан ажратилган) танланади, бунда C^{++} тили семантикаси иккита инкремент операторини биргаликда келишини ман этади. Келтирилган мисолдаги каби, операторларнинг хато анализи – компилятор ишлаш самарадорлигига унчалик таъсир этмайди ва тилнинг имкониятларини чекламайди. Аммо худди шу йўлни Фортран тилига қўллаш мумкин эмас.

Кейинчалик барча лексемлар синтактик анализ босқичида сканер билан бир қийматли ажратилган деб ҳисобланади. Барча замонавий дастурлаштириш тиллари учун бу фикр тўғри қўлланилади, қаердаки уларнинг синтаксиси компилятор имкониятларидан келиб чиққан ҳолда ишлаб чиқилган.

4. Лексем жадвали ва унда мавжуд ахборотлар таҳлили.

Синтактик анализатор ишининг натижаси ҳар бир лексем даражалари характеристикаларини ҳисобга олган бажарилувчи дастур матнида топилган барча лексемлар даражаси ҳисобланади. Бу лексем даражасини лексем жадвали деб аталувчи жадвал кўринишида тасвирлаш мумкин. Лексем жадвалида ҳар бир лексемга лексем турига боғлиқ қандайдир уникал шартли код ва қўшимча хизматли ахборот мос келади.

Лексем жадвали ҳар бир сатрида лексем кўриниши, унинг тури ва керак бўлганда, қиймати ҳақидаги ахборотга эга бўлиши бўлиши керак. Бундай жадвални ташкиллаштириш учун хизмат қиладиган маълумотлар структураси, одатда иккита майдонга эга бўлади: биринчиси – лексем тури, иккинчиси – лексем ҳақидаги ахборотга кўрсаткич.

Бундан ташқари, бажарилувчи дастурда топилган қандайдир лексем турлари ҳақидаги ахборот идентификаторлар жадвалида жойлашиши керак

(агар компилятор турли тилдаги лексемлар учун турли идентификаторлар жадвалини кўриб чиқса, у ҳолда идентификаторлар жадвалидан бирида).

Лексем жадвали ва идентификаторлар жадвалини адаштирмаслик керак – улар синтактик анализаторда қайта ишланиладиган иккита турли принципал жадвалларидир.

Лексем жадвали синтактик анализаторда қайта ишланадиган бажарилувчи дастурнинг тўла матнидан ташкил топади. Унга барча мумкин бўлган лексем турлари киради, бундан ташқари ихтиёрий лексем унда ихтиёрий марта аниқланиши мумкин. Идентификаторлар жадвали фақат аниқланган лексем турлари – идентификаторлар ва ўзгармаслардан ташкил топади. Унга кирувчи тилнинг калит (хизматчи) сўзлари, операция белгилари каби лексемлар тушади. Бундан ташқари, ҳар бир лексем (идентификатор ёки ўзгармас) идентификаторлар жадвалида бир марта қатнашиши мумкин. Шунинг ҳам эътиборга олиш керакки, лексемлар жадвалида лексемлар бажарилувчи дастурдаги каби лексемлар тушади. Бундан ташқари, ҳар бир лексем (идентификатор ёки ўзгармас) идентификаторлар жадвалида бир марта қатнашиши мумкин. Шунинг ҳам эътиборга олиш керакки, лексемлар жадвалида лексемлар бажарилувчи дастурдаги каби жойлашса (бунда лексем тартиби бузилмайди), идентификаторлар жадвалида эса лексемлар излашга энгиллик бериш учун ихтиёрий тартибда жойлашади.

Мисол сифатида, Object Pascal тилининг бажарилувчи кодидан қандайдир фрагментни ва 2.1 жадвалда кўрсатилган унга мос лексем жадвални кўриш мумкин:

```
...  
begin  
for i:=1 to N do  
fg:= fg * 0.5  
...
```

2.1 жадвал. Pascal тилидаги дастур фрагментининг лексемлари

Лексем	Лексем тури	Қиймати
--------	-------------	---------

begin	Калит сўз	x_1
for	Калит сўз	x_2
i	Идентификатор	$i : 1$
:=	Муносабат белгиси	s_1
1	Бутун сонли ўзгармас	1
to	Калит сўз	x_3
N	Идентификатор	$n : 2$
do	Калит сўз	x_4
fg	Идентификатор	$fg : 3$
*	Арифметик операция	s_1
0.5	белгиси	$fg : 3$
	Ҳақиқий ўзгармас	

2.1 жадвалда “қиймат” майдони синтактик анализатор ишининг натижасида якуний лексем жадвалга жойлаштириладиган қандайдир кодли қийматли беради. Албатта, мисолда ёзилган қийматлар шартли деб ҳисобланади. Аниқ кодлар дастурчилар томонидан компиляторни яратиш жараёнида танланади. Шунини ҳам таъкидлаб ўтиш керакки, лексемлар жадвалларининг идентификаторлар жадваллари билан алоқаси ўрнатилади (мисолда у идентификатордан кейин келадиган ':' белгили қандайдир индекс билан тасвирланади, аниқ компиляторда эса у яратиш жараёнида аниқланади).

5. Синтактик таҳлилнинг таржима тузилмалари.

Синтактик анализаторни ишлатувчи модулар икки гуруҳга бўлинади:

- Дастур коди кирувчи тилга боғлиқ бўлмаган модулар;
- Дастур коди кирувчи тилга боғлиқ бўлган модулар;

Биринчи гуруҳга қуйидаги модулар киради:

- *LexElem* — лексем жадвалининг берилган элементлари структурасини тасвирлайди;
- *FormLab2* — фойдаланувчи билан интерфейсни тасвирлайди.

Иккинчи гуруҳга қуйидаги модулар киради:

- *LexType* — номланган ва матнли ахборот билан боғланган кирувчи лексем турларини аниқлайди;
- *LexAuto* — ЎА ишини ташкиллаштиради.

Модуллари шу кўринишда бўлиш кирувчи тилни ўзгартиришда синтактик танувчини ташкиллаштириш учун маълумотлар структурасини ишлатишга рухсат беради.

№2-лаборатория ишини бажариш учун бу модуллари ташқари, №1-лаборатория ишининг бажарилишида яратилган идентификаторлар жадвалини комбинацияли ишлатишга рухсат берадиган (*TblElem* и *FncTree*) дастур модуллари ҳам ишлатилади. Бу икки модулли ҳам кирувчи тилга боғлиқ бўлмайди.

Синтактик анализаторни ташкиллаштириш учун ишлатиладиган дастур модуллари таркибини қисқача ёритамиз:

6. Лексем турлари модули.

LexType модули учун майда изоҳларга ҳожат йўқ. Унда ҳар лексемнинг номланиши ва белгиланишига мос келадиган барча мумкин бўлган лексем турлари (*TLexType* маълумотлар тури) келтирилган. Лексем номини чиқаришни *LexTypeName* функцияси, белгилашларни чиқаришни эса — *LexTypeInfo* функцияси таъминлайди. Айтиб ўтиш керакки, лексем топшириғида санаб ўтилганлардан ташқари, қатор охирини билдирадиган яна битта қўшимча ахборот лексеми (*LEX_START*) ишлатилади.

LexElem модули (*TLexem*) лексем жадвалининг элементи маълумотлар структурасини ва (*TLexList*) лексем жадвалининг ўзини ҳамда барча улар билан боғланганларни тасвирлайди.

7. Синтактик анализаторларни (сканерларни) қуриш.

Синтактик анализатор ўзгармас ва идентификаторлар каби турли кўринишдаги объектлар билан иш кўради (охиргисига калит сўзлар ҳам киради). Ўзгармас ва идентификаторларни тасвирлаш тили кўп ҳолларда давомли ҳисобланади, яъни давомли грамматика ёрдамида тасвирланиши

мумкин. Давомли тиллар учун танувчилар ўзакли автоматлар (ЎА) ҳисобланади.

Шу грамматикада берилган тиллар занжирини танувчилар ёрдамида давомли грамматика учун ўзакли автоматлар (ЎА) қурилиши ҳақидаги тўлиқ маълумотни махсус адабиётлардан топиш мумкин.

Ихтиёрий ЎА ни беш параметр ёрдамида бериш мумкин: $M(Q, \Sigma, \delta, q_0, F)$, қаердаки:

Q — автомат ҳолатининг ўзакли тўплами;

Σ — мумкин бўлган кировчи белгиларнинг ўзакли тўплами (ЎА нинг кировчи алфавити);

δ — $P(Q)\delta$: қисмий тўпламлар тўпламида $Q \cdot \Sigma$ тўпамининг берилган тасвири;

$Q \cdot \Sigma \rightarrow P(Q)$ (баъзан δ ни автоматнинг ўтиш функцияси деб атайдилар);

$q_0 \in Q$ - автоматнинг бошланғич ҳолати;

$F \subseteq Q$ —автоматнинг якуний ҳолатлари тўплами.

ЎА тасвирлашнинг бошқа усули – ЎА ўтишларининг ҳолатлари ва функциялари тўпамини график тасвирлайдиган ўтишлар графи ҳисобланади. ЎА ўтишлар графи – бу чўққиси ЎА ҳолатини берадиган, шохлари бир ҳолатдан бошқасига ўтишни тасвирлайдиган, шохларни белгиловчи белгилар эса ЎА ўтиш функциясига мос келадиган юкланган бир томонлама йўналган графдир. Агар ЎА ўтишлар функцияси бир неча белгилар бўйича q дан q' ҳолатга ўтишни кўриб чиқса, у ҳолда улар орасида q дан q' га ўтишни бажарадиган барча белгиларни кўрсатадиган битта шох қурилади.

Нодетерминалланган ЎА занжирлар анализи учун қулай эмас, қаердаки унда бир қийматли бўлмаган ҳолатлар учраши мумкин, яъни шундай ҳолатлар-ки, улардан бита ва фақат шу белги берилган икки ёки ундан кўп шохлар чиқиши мумкин. Бундан кўриниб турибдики, ЎА ишини дастурлаштириш – уч ўлчамли масала эмас. $M(Q, \Sigma, \delta, q_0, F)$ ЎА ни ишлашини дастурлаштириш учун у детерминалланган бўлиши керак – ихтиёрий

кирувчи белги учун шу $\mathcal{A}$ нинг ўтиш функцияси ҳеч бўлмаса битта ҳолатга эга бўлиши керак:

$$\forall a \in V, \forall q \in Q: \text{либо } \delta(a, q) = \{r\} r \in Q \text{ либо } \delta(a, q) = \emptyset$$

Ихтиёрий нодетерминалланган $\mathcal{A}$ уларнинг тили мос тушганда детерминалланган $\mathcal{A}$ га ўзгариши мумкинлиги исботланган [3, 7, 26] (бу $\mathcal{A}$ ларни эквивалент деб айтадилар).

Детерминалланган $\mathcal{A}$ га ихтиёрий $\mathcal{A}$ ни ўзгартиришдан ташқари у минималлаштирилиши ҳам мумкин – унга эквивалент ҳолатларнинг минимал мумкин бўлган сони билан детерминалланган $\mathcal{A}$ қуриш мумкин. $\mathcal{A}$ ни детерминалланган $\mathcal{A}$ ва минималланган $\mathcal{A}$ га ўзгартириш алгоритмлари [3,7,26] да тўлиқ тасвирланган. Ихтиёрий детерминалланган $\mathcal{A}$ ни ишлашини тасвирловчи функцияни ёзиш мумкин. Бу функцияни дастурлаштириш учун $\mathcal{A}$ нинг жорий ҳолатини тасвирловчи ўзгарувчига эга бўлиши етарли, кирувчи занжирнинг белгилари асосида бир ҳолатдан бошқасига ўтишлар эса танлаш оператори ёрдамида қурилиши мумкин. Функция иши кирувчи занжир охирига етгунча давом этади. Функция натижасини ҳисоблаш учун унинг якунида $\mathcal{A}$ ҳолати анализ қилиниши керак. Агар бу охирги ҳолатлардан бири бўлса, у ҳолда функция аъло даражада тугатилади ва кирувчи занжир қабул қилинади, акс ҳолда кирувчи занжир берилган тилга тегишли эмас.

Умумий ҳолда синтактик анализатор вазифаси лексемнинг белгилари занжирини кирувчи тилга мослигини текширишга қараганда кенгроқдир. У лексем охирини тўғри аниқлаши керак (бу ҳақда юқорида айтилган эди) танилган лексемнинг эсланишига кўра у ёки бу амални бажариши керак (уни лексем жадвалига киритиш). Бажариладиган амаллар тўплами компиляторни ишлатиш орқали сақланади. Одатда бу амаллар танилган лексем охирига етган заҳоти бажарилади.

Кирувчи матнда лексемлар махсус белгилар билан чегараланмаган. Лексем чегарасини аниқлаш – бу кирувчи белгилар умумий оқимида таниш бажариладиган қаторларни белгилашдир. Агар лексем чегаралари ҳамиша

аниқланса (юқорида шу қарорга келинган эди), у ҳолда уларни берилган терминал белгилар ва кейинги лексем бошидаги белгилар бўйича аниқлаш мумкин. Терминал белгилар – бу бўшлиқлар, амаллар белгилари, изоҳлар белгилари ҳамда ажратувчилар (вергул, нуқтали вергул ва б.). бундай терминал белгилар тўплами кирувчи тилларга боғлиқ ҳолда ишлатилиши мумкин. Шунини таъкидлаш керакки, амаллар белгиларининг ўзи лексем ҳисобланади ва уларни матнни танишда тадлаб юбормаслик керак.

Шундай қилиб, оддий сканернинг ишлаш алгоритмини қуйидагича тасвирлаш мумкин:

- лексемни чегаралайдиган навбатдаги белгини топгунча жорий тилда дастурдаги белгиларнинг кирувчи оқими кўриб чиқилади;
- кирувчи оқимнинг танланган қисми учун лексемни таниш функцияси бажарилади;
- таниш аъло даражада бажарилганда белгиланган лексем ҳақидаги ахборот лексемлар жадвалига юборилади ва алгоритм биринчи босқичга қайтади;
- таниш бажарилмаганда, хато ҳақидаги хабар чиқарилади, кейинги ҳаракатлар эса сканер ишига боғлиқ бўлади; ё унинг бажарилиши тўхтатилади, ё кейинги лексемни таниш имконияти берилади (алгоритмнинг биринчи босқичига қайтади).

Сканер – дастурнинг иши кирувчи оқимдаги жорий тилда дастурнинг барча белгилари кўриб чиқилгунча давом этади.

8. Ишни бажаришга қўйилган талаблар.

Курс иши қуйидаги тартибда бажарилиши керак:

1. Талаба ўқитувчидан масала бўйича топшириқ олинади;
2. Идентификаторлар жадвалини ташкиллаштиришда берилган усуллар учун қўлланиладиган маълумотлар структурасини қўллаш амалга оширилади;
3. Ҳисобот тайёрланади ва ҳимоя қилинади;

9. Топширик вариантлари.

1. Кирувчи тил; (нуктали вергул) белгиси билан ажратилган арифметик ифодадан иборат. Арифметик ифодалар идентификаторлар, сузувчи нуктали ўнлик сон (оддий ва логарифмик шаклда), муносабат белгиси ($:=$), $+$, $-$, $*$, $/$ амаллар белгилари ва қавслардан ташкил топади.

Курс ишидаги топширикни бажаришга мисол сифатида ;(нуктали вергул) белгиси билан ажратилган `if... then ... else` ва `if... then` турли шартли операторлар тўпламидан ташкил топган кирувчи тилни оламиз. Бу операторлар шарт сифатида операндалари идентификаторлар ва белгисиз бутун ўнли ўзгармаслар ҳисобланадиган `or`, `xor` ва `and` амаллари ёрдамида қурилган мантикий ифодаларни ўз ичига олади. Бажарилиш қисмида бу оператор ё мантикий ифода ўзгарувчисининг муносабат операторидан ё бошқа шартли оператордан иборат бўлади.

Изоҳ очувчи фигурали қавс (`{}`) билан бошланиб, ёпувчи фигурали қавс (`}`) билан тугалланадиган белгилар кетма-кетлиги кўринишида ташкиллаштирилади. Изоҳлар ихтиёрий алфавит-рақамли белгилардан, шу билан биргаликда миллий алфавитдан ташкил топиши мумкин.

9.1. Топширик учун кирувчи тил грамматикаси.

Юқорида берилган кирувчи тил P қоидалари билан $G(\{if, then, else, a, :=, or, xor, and, (,), ;\}, \{S, F, E, D, C\}, P, S)$ КО-грамматикаси билан қурилиши мумкин.

$S \rightarrow F;$

$F \rightarrow if\ E\ then\ T\ else\ F \mid if\ E\ then\ F \mid a := E$

$T \rightarrow if\ E\ then\ T\ else\ T \mid a := E$

$E \rightarrow E \text{ or } D \mid E \text{ xor } D \mid D$

$D \rightarrow D \text{ and } C \mid C$

$C \rightarrow a \mid (E)$

Грамматика тавсифи Бэкус—Наур шаклида қурилган. Грамматикада вақоидаларда қуюқ шрифтлар билан терминал белгилар белгиланган.

Мисол сифатида тасвирланган тил ва унинг берилган грамматикаси юқоридаги вариантлардан биронтаси билан мос тушмайди. Бошқа томондан, бу мисолда синтактик, кейинчалик турли вариантлар билан келувчи синтаксис танувчиларнинг кўпгина хусусиятларини кўриш мумкин. У жорий дастурнинг у ёки бу жойига бошқаришни узатиш билан боғланган шартли оператор каби, мантикий ифодаларни ҳисоблаш шаклидаги чизиқли операциялардан ташкил топади. Шунинг учун бу мисол №2-лаборатория иши учун бажарилиш сифатида танланган, кейинчалик эса у №3-лаборатория ишида ишлатилади.

9.2. Топшириқдаги кирувчи тилнинг лексемасини танлаш учун ўзакли автомат (ЎА) ни тавсифлаш.

Синтактик анализатор вазифаси дастурнинг жорий матнида шу тилнинг барча лексемларини топиш ва танлаш ҳисобланади. Берилган тилнинг лексемлари қуйидагилар:

- тилнинг олтита калит сўзлари;
- ажратувчилар; очувчи ва ёпувчи қавслар; нуқтали вергул;
- идентификаторлар;
- белгисиз бутун ўнлик ўзгармаслар.

Санаб ўтилган лексемлардан ташқари, танувчи қуриш принципи юқорида тавсифланган кирувчи тилдан изоҳларни аниқлашни ва ўчиришни бажариши керак. Изоҳларни калит сўзлар билан белгилаш учун очувчи ва ёпувчи фигурали қавслар бўлиши керак.

Санаб ўтилган лексемлар ва изоҳлар учун регуляр грамматикани қуриш, кейинчалик эса у асосида ЎА яратиш мумкин. Бу кўринишда қурилган грамматика бир томондан элементар оддий, иккинчи томондан кам

ахборотли ва кичик бўлади. Шунинг учун ЎА ни лексем тавсифи бўйича қуриш йўли билан бориши керак. Бунинг учун фақат идентификаторлар ва белгисиз ўнлик ўзгармасларни тасвирлаш керак:

- идентификатор – бу катта ва кичик лотин алфавитининг харфларидан (А дан Z гача, а дан z гача), рақамлар (0 дан 9 гача) ва ҳарф ёки қуйи чизик белгиси билан бошланадиган () қуйи чизик белгиси кетма – кетлиги;
- белгисиз ўнлик сон – бу ихтиёрий рақам билан бошланувчи (0 дан 9 гача) рақамлар кетма – кетлиги;

Берилган танувчи учун лексем чегараси сифатида пробел, табуляция белгиси, қаторларни ўтказиш ва қареткага қайтиш белгилари, бундан ташқари қавслар, очувчи фигурали қавс, нуқтали вергул ва иккита нуқта белгиси хизмат қилади. Шу ерда эслатиб ўтиш керакки, қавслар ва нуқтали вергулларнинг ўзи лексем ҳисобланади, очувчи фигурали қавс изоҳни бошлайди, икки нуқта белгиси эса лексемнинг чегараси, шу вақтнинг ўзида бошқа лексемнинг боши – муносабат белгиси ҳисобланади.

Берилган тилда синтактик анализатор ҳамма вақт лексем чегарасини бир қийматли аниқлаши мумкин, шунинг учун унинг синтаксис анализатор ва компиляторнинг бошқа элементлари билан бўлиқлигига ҳожат йўқ.

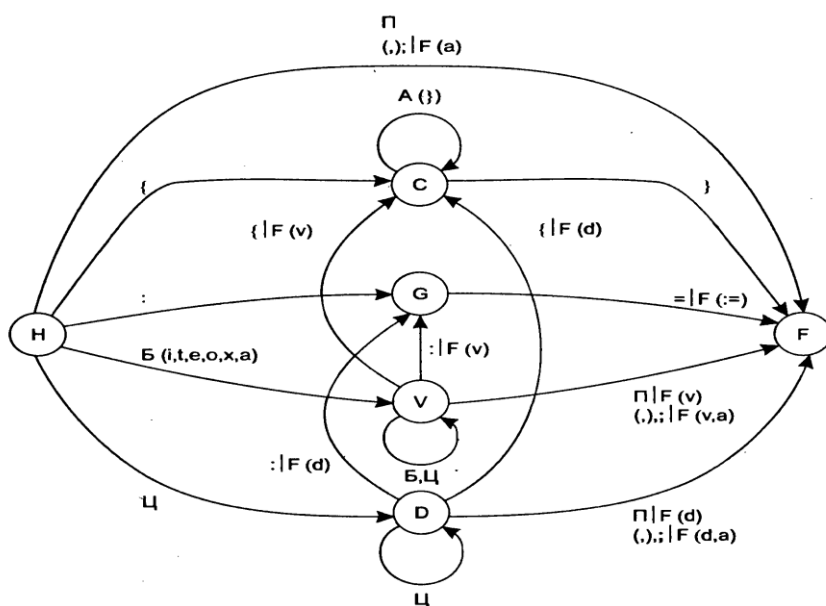


Рис. 2.1. Фрагмент графа переходов КА для распознавания всех лексем, кроме ключевых слов

2.1-расм. Калит сўзлардан ташқари барча лексемларни таниш учун ЎА ўтишлари графига фрагмент

ЎА ўтишнинг тўлиқ графини кўриш учун катта ва ноқулай бўлганлиги учун, уни бир неча фрагментлар орқали кўриб чиқамиз. 2.1-расмда ажратувчилар, изоҳлар, муносабат белгилари, ўзгарувчилар ва ўзгармасларни (калит сўзлардан ташқари кирувчи тилнинг барча лексемлари) танишга жавоб берадиган ЎА ўтиш графининг фрагменти тасвирланган.

2.2-расмда if ва then калит сўзларини танишга жавоб берадиган ЎА ўтиш графининг фрагменти тасвирланган: (бу фрагмент 2.1-расмда тасвирланган ҳолатда ҳаволага эга). Ўхшаш фрагментларни бошқа калит сўзлар учун қуриш мумкин.

2.2-расмда if ва then калит сўзлари учун ЎА ўтиш графининг фрагменти

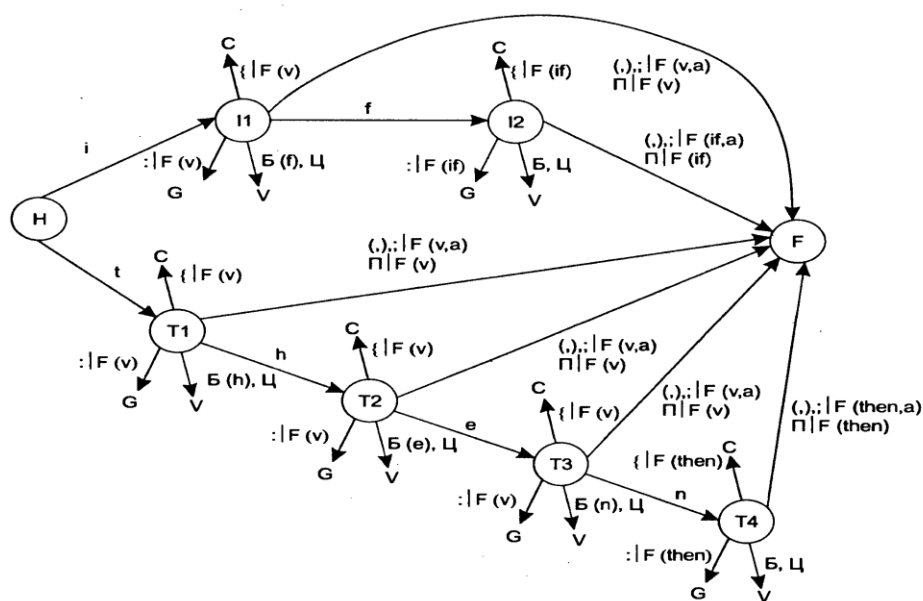


Рис. 2.2. Фрагмент графа переходов КА для ключевых слов if и then

2.1 ва 2.2-расмларларда тасвирланган ЎА ўтиш графининг фрагментлари учун қуйидаги белгилашлар қабул қилинган.

- A – ихтиёрий алфавитли – рақамли белгилар;
- A(*) – қавсларда санаб ўтилганлардан ташқари, ихтиёрий алфавитли – рақамли белгилар;
- П – ихтиёрий кўринмас белги (пробел, табуляция белгиси, қаторга ўтиш, қареткага қайтиш);

- Б – англиз алфавитининг ихтиёрий ҳарфи (катта ёки кичик) ёки () қуйи чизик белгиси;

- Б(*) - кавсларда санаб ўтилганлардан ташқари, англиз алфавитининг ихтиёрий ҳарфи (катта ёки кичик) ёки () қуйи чизик белгиси;

- Ц – 0 дан 9 гача ихтиёрий рақамлар;

- F — ЎА ўтишларининг бир ҳолатдан бошқасига чақирадиган лексем жадвалини чақириш функцияси. Унинг аргументларини тасвирлаш:

- v — ЎА билан ишлашда сақланган ўзгарувчи;

- d — ЎА билан ишлашда сақланган ўзгармас;

- a — ЎА нинг жорий кирувчи белгиси.

Бу белгилашларни ҳисобга олган ҳолда ЎА ни тўлиқ ҳолда қуйидаги кўринишда тасвирлаш мумкин:

$M(Q, \Sigma, \delta, q_0, F)$:

$Q = \{H, C, G, V, D, I, I_2, T_1, T_2, T_3, T_4, E_1, E_2, E_3, E_4, O_1, O_2, X_1, X_2, X_3, A_1, A_2, A_3, F\}$

$Z = A$ (барча рухсат этилган алфавитли-рақамли белгилар);

$q_0 = H$;

$F = \{F\}$;

ЎА нинг бошланғич ҳолатидан «i», «t», «e», «o», «x» ва «a» литералари калит сўзларга мос келадиган ҳолатлар занжирининг бошига келтирилади:

- I_1, I_2 ҳолати —if калит сўзига;
- T_1, T_2, T_3, T_4 ҳолати —then калит сўзига;
- E_1, E_2, E_3, E_4 ҳолати—else калит сўзига;
- O_1, O_2 ҳолати —or калит сўзига;
- X_1, X_2, X_3 ҳолати —хoг калит сўзига;
- A_1, A_2, A_3 ҳолати—and калит сўзига.

Қолган литералар V – ўзгарувчисига (идентификаторга) мос келадиган ҳолатга келтирилади.

Агар қандайдир занжирда калит сўз ёки рақамга мос келмайдиган литера учраса, у ҳолда ЎА ҳам V ҳолатган ўтади, агарда лексем чегараси учраса – калит сўзнинг ўқилган қисмини ўзгарувчи сифатида сақлаб қолади (калит сўзларнинг боши билан мос тушадиган «i» ёки «els» идентификаторлар каби тўғри белгиланиши учун).

Рақамлар D – кировчи ўзгарсмасга мос келувчи ҳолатга келтирилади. Очувчи фигурали қавс изохни топишга мос келадиган C ҳолатга келтирилади – бу ҳолатдан ЎА фақат киришда ётувчи фигурали қавсни олса, чиқади. Яна бир ҳолат – G – “муносабат белгиси” лексемсига мос тушади. Унга киришга икки нуқтани олиб, ЎА ўтади ва бу ҳолатда “тенглиак” белгисини кутади.

H ҳолати — ЎА нинг бошланғич ҳолати, F ҳолат эса — унинг охириги ҳолати. ЎА ўзакли ҳолатга ўтганда лексемларнинг узлуксиз оқими билан ишлаганлигидан, у шу вақтнинг ўзида навбатдаги лексемни таниш учун бошланғич ҳолатга қайтиши керак. Шунинг учун дастурни моделлаштиришда бу икки ҳолатни бирлаштириш мумкин.

Графда ва ўтишлар функциясини тасвирлашда, шусиз ҳам қийин бўлган граф ва функцияни мавҳумлаштирмаслик учун “хато” хабари белгиланмаган. Бу ҳолатга ЎА жорий ҳолатдан ўтиб бўлмаганда киришга белгини олганда ўтади.

F функцияси лексем жадвалига маълумотларнинг ёзилишини бажаришга мос келади. Функция аргументлари A нинг жорий ҳолатига боғлиқ бўлади. ЎА ишини моделлаштирувчи дастурнинг бажарилишида бу функция жорий ҳолат ва кировчи белгига боғлиқ ҳолда чақириладиган бир неча функцияларга мос келиши керак. Эслатиб ўтиш керакки, лексем жадвалига ўзгармас ва ўзгарувчиларни тўғри ёзиш учун ЎА уларга мос келувчи белгилар занжирини сақлаб қолиши керак. Буни у ҳар гал H ҳолатида бўлганда ЎА бошчасининг ҳисоблаш жойини сақлаб қолган ҳолда бажариши керак.

Айтиб ўтиш лозимки, ЎА ўтиш функцияси бир мунча мураккаб ва ўз вақтида оддий ҳосил бўлади (барча калит сўзлар учун у бир турли бўлади).

ЎА ни ишлатишда калит сўзлар алоҳида ҳолатларни белгиланиб, V ҳолатда ЎА га киришда ҳарфларни ҳамма вақт учратганда ўтиш керак. У ҳолда ўқилган қатор калит сўз ёки идентификатор эканлигини текшириш, лексем жадвалига қаторларни таққослашнинг стандарт операциялари ёрдамида ёзиш вақтида бажариш мумкин. ЎА ўтишлар графи бу вариантда бир мунча яхши бўларди – у 2.1-расмда тасвирланган фрагмент каби бўлади. Уни “қисқартирилган” ЎА ўтиш графи (ёки “қисқартирилган ЎА”) деб аташ мумкин.

Аммо шуни таъкидлаш керакки, ўзининг кенг имкониятлари ва оддийлигига қарамай, қисқартирилган ЎА янада яхшироқ бўлади, чунки жадвалга лексемни ёзиш вақтида у лексемни барча маълум калит сўзлар билан таққослашни бажариши керак бўлади (берилган ҳолда олтига калит сўзни аниқлаш керак бўлади – бундан қаторларнинг олтига таққосланиши бажарилади). Бундан эса ЎА кирувчи занжирнинг ўқилган қисмини бир неча марта такроран кўриб чиқиши керак бўлади. Қисқартирилган ЎА ни ишлатишда аниқ кўриниб турибдики, бу амал мавжуд эмас, у барибир қаторларни таққослашнинг кутубхонали функциясини чақиришда бажарилади. Шундай қилиб, қисқартирилган ЎА ҳолатлар сонига кўра кам ва ишлатишда оддий бўлса ҳам, барча кирувчи лексем анализида қурилган тўлиқ ЎА га кўра қулайдир. Барча ҳолларда ҳам ЎА ни ишлатишнинг иккала варианты ҳам талаб этилган синтактик анализаторни қуришни таъминлайди. Қайси бирини танлашни дастурчининг ўзи танлайди.

10. Х у л о с а.

Курс иши натижасида ўзакли автомат асосида синтактик анализатор куриш амалга оширилган. Курилган синтактик анализатор лексемларни жорий этиш дастурининг матнида қуйидаги турларни ажратишга рухсат беради:

- калит сўзлар (if, then, else, or, xor ва and);
- идентификаторлар (бунда идентификаторлар исмида катта ва кичик инглиз ҳарфлари фарқланади);
- таққослаш амаллари белгилари;
- белгисиз бутун ўнлик ўзгармаслар;
- ажратгичлар (қавслар ва нуқтали вергул).

Синтактик анализаторга кирувчи дастурнинг матнида бўшлиқлар, табуляция белгилари ва сатрларни қайтариш ҳамда фигурали қавсларда ажратилган изоҳларни ташлаб юборади.

Хато лексемлар (масалан, ҳарфлар билан ёзилган сон), ёпилмаган изоҳ ёки тугалланмаган лексемлар (бундай лексемда фақат “.” белгиси бўлади) бўлган ҳолларда синтактик анализатор хато ҳақидаги хабар беради ва кейинги

анализни тўхтади. Анализатор бир неча хатолар сони кўп ҳолларда улардан фақат бирини аниқлайди.

Синтактик анализнинг бажарилиш натижаси лексем жадвалини тасвирлайдиган маълумотлар структураси ҳисобланади. Қурилган синтактик анализатор натижасидаги лексемлар жадвали синтаксис анализ ва код генерацияси билан боғланган кейинги босқич ишларни бажариш учун керакли маълумотларни тайёрлашга мўлжалланган.

11. Фойдаланилган адабиётлар.

8.1. Асосий адабиётлар.

1. Вирт Н. Алгоритмы и структуры программы. М., Мир, 1985.
2. Берзтисс А.Т. Структуры данных. М. Статистика, 1984.
3. Трамбле Ж., Соренсон П. Введение в структуры данных. М., Машиностроение, 1982.
4. Ленгсам Й. Структуры данных для персональных ЭВМ. М., Мир, 1989.

8.2. Қўшимча адабиётлар.

1. 6. Дж Ульман, и др. Введение в системы баз данных. М., Лори , 2000.
2. Д. Мейер, Теория реляционных баз данных, М, Мир, 1987.

3. Салахутдинов В.Х.. Конспект лекций по курсу “Структуры и организация данных в ЭВМ ” для студентов по специальности В-52200 “Информатика и информационные технологии” , Ташкент, 2000.

8.3. Информацион-техник воситалар.

Курс ишини бажариш учун ТАТУ Самарқанд филиалининг компьютерлар билан жихозланган ўқув синфларидан фойдаланилди.