

**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: Leksemlarni determinirlashgan tugallangan avtomatlar
yordamida aniqlash**

Bajaruvchi: 301 guruh talabasi

_____ Qarshiev A.

Kurs ishi rahbari:

_____ Kat. o'qit. Murodov O'.

Samarqand – 2013

**Мавзу: Лексемларни детерминирлашган тугалланган автоматлар (КА)
ёрдамида аниқлаш.**

Режа: 1. Ишнинг мақсади.

2. Лексик анализаторнинг вазифаси.
3. Лексемлар ва уларнинг чегараларини аниқлаш.
4. Лексем жадвали ва унда мавжуд ахборотлар таҳлили.
5. Детерминирлашган ва детерминирлашмаган тугалланган автоматлар.
6. Кирувчи тилнинг лексемасини таниш учун тугалланган автоматни тавсифлаш.
7. Танувчи дастур модуллари.
8. Ишни бажаришга қўйилган талаблар.
9. Топшириқнинг ечим босқичлари.
10. Хулоса.
11. Фойдаланилган адабиётлар.

1. Ишнинг мақсади.

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

2. Лексик анализаторнинг вазифаси.

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

Лексем (тилнинг лексик 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 нуктаи назардан лексик 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 лексик анализаторларни қўшиш мумкин. Булар қуйидаги сабаблар:

- лексик 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;
- лексик анализатор конструкцияси бўйича мураккаб синтаксис анализаторнинг структураси кировчи тил версиясига боғлиқ равишда қўшиладиган бажарилувчи дастур матни ишидан ажратилади;
- компиляторнинг бундай конструкциясида бир тил версиясидан бошқа тилга ўтишда оддий лексик анализаторни қуриш мумкин.

Лексик анализаторда бажариладиган функциялар ва бажариладиган дастур матнида ажратиладиган лексем таркиби компилятор версиясига боғлиқ ҳолда ўзгариши мумкин. Асосан, лексик анализаторлар бажарилувчи

дастур матнидан изоҳлар ва белгиланмаган пробеллар, бундан ташқари куйидаги тур лексемларини ажратадилар: идентификаторлар, сатрли, белгили ва сонли ўзгармаслар, амал белгилари, ажратувчилар ва кирувчи тилнинг калит (хизматчи) сўзлари бажарилади.

Компиляторларнинг кўпчилигида лексик ва синтаксис анализаторлар – бу ўзаро боғланган қисмлардир. Лексик 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 лексик танлаш босқичида бажарилишга ҳаракат қилади. Лексик анализатор имкониятлари синтаксис анализатор имкониятларига кўра чегараланган, чунки унинг асосида бир мунча оддий механизмлар ётади.

Компиляторда лексик анализатор роли ва унинг лексик анализатор билан ўзаро алоқаси ҳақидаги маълумотларни махсус адабиётларда батафсил кўриб чиқиш мумкин.

3. Лексемлар ва уларнинг чегараларини аниқлаш.

Оддий ҳолда лексик ва синтаксис анализ фазалари компиляторда кетма-кет бажарилиши мумкин. Аммо кўпгина дастурлаштириш тиллари учун лексик 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 лексик анализи босқичма – босқич бажарилади. У ҳолда лексик ва синтаксис анализаторлар параллел равишда, бир – бирига навбати билан мурожаат қилиб бажарилиши керак. Лексик анализатор навбатдаги лексемни топиб, уни синтаксис анализаторга беради, у бажарилган дастурнинг ўқилган қисмини анализ қилишга ҳаракат қилади ва, ёки лексик 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. Лексик ва синтаксис анализаторлар орасидаги ўзаро алоқа схемаси махсус адабиётларда кенг берилган.

Кўриниб турибдики, лексик 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.

Лексик 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)

лексик 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 лексик 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.

Лексик 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 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 лексик анализаторда қайта ишланиладиган иккита турли принципал жадвалларидир.

Лексем жадвали лексик 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 иборат 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 лексик 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. Детерминирлашган ва детерминирлашмаган тугалланган автоматлар.

Лексик 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 тугалланган автоматлар (КА) ҳисобланади.

Грамматикада берилган тиллар занжирини танувчилар ёрдамида давомли грамматика учун тугалланган автоматлар (КА) қурилиши ҳақидаги тўлиқ маълумотларни маъруза дарсида кўрсатилган адабиётлардан топиш мумкин.

Ихтиёрий КАни беш параметр ёрдамида бериш мумкин:

$$M(Q, \Sigma, \delta, q_0, F), \text{ бу ерда:}$$

Q — автомат ҳолатининг тугалланган тўплами;

Σ — мумкин бўлган кирувчи белгиларнинг тугалланган тўплами (КАнинг кирувчи алфавити);

δ — $P(Q)\delta$: қисмий тўплалар тўпламида $Q \cdot \Sigma$ тўпламининг 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.

КАни тасвирлашнинг бошқа усули – КА ўтишларининг ҳолатлари ва функциялари тўпламини график тасвирлайдиган ўтишлар графи ҳисобланади. КА ўтишлар графи – бу чўққиси КА ҳолатини берадиган, шохлари бир ҳолатдан бошқасига ўтишни тасвирлайдиган, шохларни белгиловчи белгилар эса КА ўтиш функциясига мос келадиган юкланган бир томонлама йўналган графдир. Агар КА 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 КА занжирлар анализи учун қулай эмас, қаердаки унда бир қийматли бўлмаган ҳолатлар учраши мумкин, яъни шундай ҳолатлар-ки, улардан битта ва фақат шу белги берилган икки ёки ундан кўп шохлар чиқиши мумкин. Бундан кўриниб турибдики, КА ишини дастурлаштириш – уч ўлчамли масала эмас. $M(Q, \Sigma, \delta, q_0, F)$ КАни ишлашини

дастурлаштириш учун у *детерминалланган* бўлиши керак – ихтиёрий кирувчи белги учун шу КА нинг ўтиш функцияси ҳеч бўлмаса битта ҳолатга эга бўлиши керак:

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

Ихтиёрий нодетерминалланган КА уларнинг тили мос тушганда детерминалланган КА га ўзгариши мумкинлиги исботланган (бундай КАларни эквивалент деб айтадилар).

Детерминалланган КАга ихтиёрий КАни ўзгартиришдан ташқари у минималлаштирилиши ҳам мумкин – унга эквивалент ҳолатларнинг минимал мумкин бўлган сони билан детерминалланган КАни куриш мумкин. КАни детерминалланган КА ва минималланган КАга ўзгартириш алгоритмлари махсус адабиётларда тўлиқ тасвирланган. Ихтиёрий детерминалланган КАни ишлашини тасвирловчи функцияни ёзиш мумкин. Бу функцияни дастурлаштириш учун КАнинг жорий ҳолатини тасвирловчи ўзгарувчига эга бўлиши етарли, кирувчи занжирнинг белгилари асосида бир ҳолатдан бошқасига ўтишларни эса танлаш оператори ёрдамида курилиши мумкин. Функция иши кирувчи занжир охирига етгунча давом этади. Функция натижасини ҳисоблаш учун унинг якунида КА 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 лексик анализатор вазифаси лексеманинг белгилари занжирини кирувчи тилга мослигини текширишга қараганда кенгрокдир. У лексема охирини тўғри аниқлаши керак (бу ҳақда юқорида айтилган эди) танилган лексеманинг танилишига кўра у ёки бу амални бажариши керак (масалан, уни лексема жадвалига киритиш). Бажариладиган амаллар тўплами компиляторни ишлатиш орқали сақланади. Одатда бу амаллар танилган лексемалар охирига етган заҳоти бажарилади.

Кирувчи матнда лексемалар махсус белгилар билан чегараланмаган. Лексема чегарасини аниқлаш – бу кирувчи белгилар умумий оқимида таниш

бажариладиган қаторларни белгилашдир. Агар лексема чегаралари ҳамиша аниқланса, у ҳолда уларни берилган терминал белгилар ва кейинги лексема бошидаги белгилар бўйича аниқлаш мумкин. Терминал белгилар – бу бўшлиқлар, амаллар белгилари, изоҳлар белгилари ҳамда ажратувчилар (вергул, нуқтали вергул ва бошқалар). Бундай терминал белгилар тўплами кирувчи тилларга боғлиқ ҳолда ишлатилиши мумкин. Шунини таъкидлаш керакки, амаллар белгиларининг ўзи лексема ҳисобланади ва уларни матнни танишда ташлаб юбормаслик керак.

Шундай қилиб, оддий сканер таниш алгоритмини қуйидагича тасвирлаш мумкин:

- лексемани чегаралайдиган навбатдаги белгини топгунча жорий тилда дастурдаги белгиларнинг кирувчи оқими кўриб чиқилади;
- кирувчи оқимнинг танланган қисми учун лексемани таниш функцияси бажарилади;
- таниш аъло даражада бажарилганда белгиланган лексема ҳақидаги ахборот лексемалар 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 лексемани таниш имконияти берилади (алгоритмнинг биринчи босқичига қайтади).

Сканерлаш (таниш) – дастурининг иши кирувчи оқимдаги жорий тилда дастурнинг барча белгилари кўриб чиқилгунча давом этади.

6. Кирувчи тилнинг лексемасини таниш учун тугалланган автоматни тавсифлаш.

Лексик анализатор вазифаси дастурнинг жорий матнида шу тилнинг барча лексемаларини топиш ва таниш ҳисобланади. Берилган тилнинг лексемалари қуйидагилар:

- тилнинг олти калит сўзлари;
- ажратувчилар; очувчи ва ёпувчи қавслар; нуқтали вергул;

- идентификаторлар;
- белгисиз бутун ўнлик ўзгармаслар.

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

Санаб ўтилган лексемалар ва изоҳлар учун регуляр грамматикани қуриш, кейинчалик эса у асосида КА яратиш мумкин. Бу кўринишда қурилган грамматика бир томондан элементар оддий, иккинчи томондан кам ахборотли ва кичик бўлади. Шунинг учун КА ни лексема тавсифи бўйича қуриш йўли билан бориши керак. Бунинг учун фақат идентификаторлар ва белгисиз ўнлик ўзгармасларни тасвирлаш керак:

- идентификатор – бу катта ва кичик латин алфавитининг ҳарфларидан (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 лексема 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 лексема ҳисобланади, очувчи фигурали қавс изоҳни бошлайди, икки нуқта белгиси эса лексеманинг чегараси, шу вақтнинг ўзида бошқа лексеманинг боши – муносабат белгиси ҳисобланади.

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

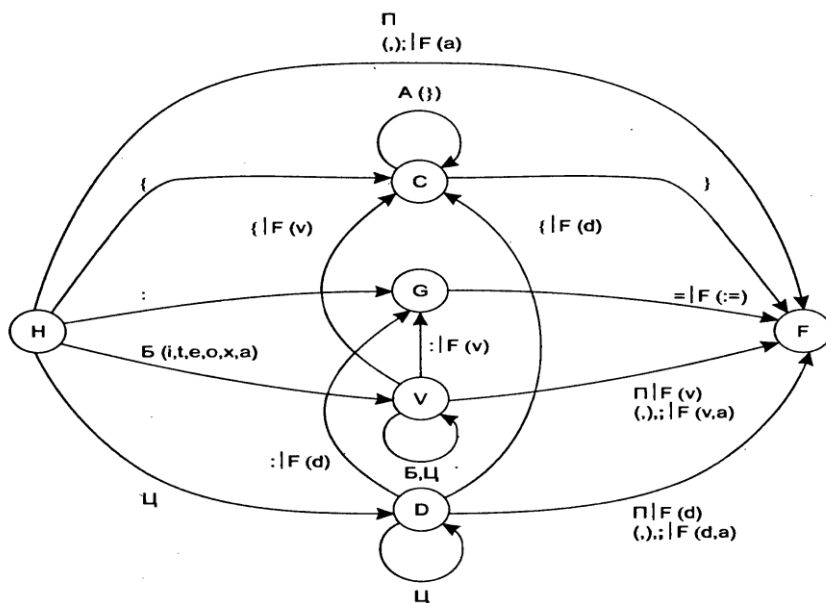


Рис. 2.1. Фрагмент графа переходов КА для распознавания всех лексем, кроме ключевых слов

2.1-расм. Калит сўзлардан ташқари барча лексемаларни таниш учун ўтиш графиги.

7. Танувчи дастур модуллари.

Юқорида кўрсатиб ўтилган модуллардан ташқари, интерфейсларни фойдаланувчи билан боғлайдиган яна бир модул зарур. Бу модул (*FormLab2*) *VCL* кутубхонасида *TForm* синф асосида *TLab2Form* график ойнани ишлаб чиқади ва қуйидагилардан иборат бўлади:

- Дастур коднинг файли (*FormLab2.pas* файли);
- Фойдаланувчи интерфейси ресурсларини тасвирлаш файли (*FormLab2.dfm* файли).

Интерфейс шакли ва унинг бошқарув органларини тасвирлашдан ташқари *FormLab2* модули лексема жадвалида ссылкарни ёзадиган (*listLex*) ўзгарувчисидан ташкил топади.

Модулда тасвирланган интерфейс шакли қуйидаги асосий бошқарув органларидан ташкил топади:

- Мос ҳолда “жорий файл” ва “лексема жадвали” исмли иккита (*SheetFile* ва *SheetLexems*) ва иккита тузилган (*PageControll*) кўп сатрли илова;
- *SheetFile* тузилма;

- (EditFile) файл исмларни киритиш майдони, (BtnFile) файл тизими каталогидаги файл исмини танлаш тугмаси, (BtnLoad) файлни ўқиш тугмаси;
- (ListIds) ўқилган файлни сақлаш учун кўп сатрли майдон;
- SheetLexems тузилмаси;
 - ўқилган лексемлар ҳақидаги маълумотларни тасвирлаш учун 3 устунли (GridLex) тўри;
 - (BtnExit) дастури билан ишни тугатиш тугмаси.

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. Топшириқ вариантлари.

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

Курс ишидаги топшириқни бажаришга мисол сифатида ;(нуқтали 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 кавс ({) билан бошланиб, ёпувчи фигурали кавс (}) билан тугалланадиган белгилар кетма-кетлиги кўринишида ташкиллаштирилади. Изоҳлар ихтиёрий алфавит-рақамли белгилардан, шу билан биргаликда миллий алфавитдан ташкил топиши мумкин.

10. Х у л о с а.

Курс иши натижасида тугалланган автомат асосида if ва then калит сўзлар учун лексик анализатор қуриш амалга оширилган. Қурилган лексик 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).

Лексик 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 лексик анализатор хато ҳақидаги хабар беради ва кейинги анализни тўхтади. Анализатор бир неча хатолар сони кўп ҳолларда улардан фақат бирини аниқлайди.

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

11. Фойдаланилган адабиётлар.

11.1. 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.

11.2. 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.

11.3. Информацион-техник воситалар.

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