LFTC – **Seminar 9**

1. **FIRST1, FOLLOW1**
2. Determinați FIRST1 and FOLLOW1

pentru neterminalele gramaticii următoare:

S → abA

S → e

A → Saa

A → b

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, e |  |
| A | b |  |

Explicații vagi First1 (» primul terminal de după →, se aplică recursiv):

S → abA (1)

S → e (2)

A → Saa (3)

A → b (4)

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, e |  |
| A | b |  |

Pentru producția (3), care începe cu S, se va înlocui S cu “First1” său

A → Saa (3) devine A → aaa (3’), respectiv A → aa (3”)

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, e |  |
| A | b, a |  |

Explicații vagi Follow1 (» primul terminal de după Neterminaul curent din membrul drept al producțiilor (din dreapta →), se aplică recursiv, se folosește First1):

Implicit avem $ pentru Neterminalul de pornire

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, e | $, a |
| A | b, a |  |

S → abA

S → e

A → **S**aa

A → b

Pentru un neterminal care se află pe ultima poziție într-o regulă de producție, sau care este urmat de o secvență de neterminale care îl au pe e în First1 (deci ajunge pe ultima poziție), se adaugă Follow1 al Neterminalulului producției curente. DeoareceS ce va urma după neterminaul respectiv va urma după neterminaul de pe ultima poziție.

S → ab**A**

Deci vom adăuga în Follow1(A) pe Follow1(S), deoarece în momentul în care îl vom înlocui pe S cu această producție, tot ce va urma după S va urma după A.

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, e | $, a |
| A | b, a | $, a |

Observație! Deși veți avea impresia că $-ul apare tot timpul în Follow1, el este tot timpul în Follow1(S), este doar o coincidență că ajunge să fie inclus în Follow1 al altor neterminale.

**2. Analiza sintactică descendentă : LL(1)**

1. Dându-se gramatica:

S → if c then S endif

S → if c then S else S endif

S → stmt

Dacă înlocuim**: *if c then*** cu ***a***, **else** cu ***b***,

***endif*** cu ***c***, și ***stmt*** cu ***i*** avem:

S → a S c

S → a S b S c

S → i

Pentru una dintre cele 2 gramatici de mai sus:

a) Verificați dacă gramatica este LL(1).

b) Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

c) Folosind un analizor descendent verificați dacă secvența: if c then if c then stmt else stmt endif endif (sau echivalenta ei scrisa cu a,b,c,i) aparține limbajului generat de gramatică.

...

1. Fie gramatica ambiguă:

S → if c then S else S

S → if c then S

S → stmt

Dacă înlocuim**: *if c then*** cu ***a***, ***else*** cu ***b***,

***endif*** cu ***c***, și ***stmt*** cu ***i*** avem:

S → a S b S

S → a S

S → i

Pentru una dintre cele 2 gramatici de mai sus:

a) Verificați dacă gramatica este LL(1).

b) Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

c) Discutați, împreună cu cadrul didactic, cum se poate modifica tabelul de analiză astfel încât să se elimine conflictele.

d) Folosind analizorul LL(1) dacă secvența: if c then if c then stmt else stmt (sau echivalenta ei scrisa cu a,b,c,i) aparține limbajului generat de gramatică.

***Rezolvare:***

1. Verificați dacă gramatica este LL(1). (*Adică să nu avem conflicte – 2 elemente în aceeași a tabelei de analiză LL(1)*)

S → a S b S (1)

S → a S (2)

S → i (3)

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, i | $, b |

Dreapta regulii de producție

S → a S b S (1)

S → a S (2)

S → i

S este și ultimul (umat de e), deci vom adăuga în Follow1(S) întreg al Follow1(S), deci nu se modifică nimic.

Tabelul de analiză LL(1)

S → a S b S (1)

S → a S (2)

S → i (3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | i | $ |
| S | (aSbS,1)  (aS,2) |  |  |  |
| a |  |  |  |  |
| b |  |  |  |  |
| i |  |  |  |  |
| $ |  |  |  |  |

Avem conflict (2 în aceeași celulă) => Gramatica nu e de tip LL(1)

1. Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

S → a S b S

S → a S

S → i

După factorizarea la stânga (se păstrează doar parte comună + un neterminal nou. Neterminal nou va genera partea lipsă):

S → a S A (factorul comun + un neterminal nou, A)

S → i

A → bS (restul din (1))

A → e (restul din (2))

Renumerotăm:

S → a S A (1)

S → i (2)

A → bS (3)

A → e (4)

Determinăm First & Follow

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, i | $, b |
| A | b, e |  |

Implicit la Follow1(S) este $

După S urmează A, deci vom pune First1(A), dar nu și e

**Pentru e, respectiv cazul în care X se află pe ultima poziție a unei producții a lui Y, Y ® aX, se include în Follow1(X) tot Follow1(Y).**

Se repetă recursiv până ce nu se mai modifică nimic.

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | a, i | $, b |
| A | b, e | $, b |

S → a S A (1)

S → i (2)

A → bS (3)

A → e (4) se va adăuga în coloanele Follow1(lui A)

Tabelul de analiză LL(1):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | i | $ |
| S | (aSA,1) |  | (i,2) |  |
| A |  | (bS,3) (e,4) |  | (e,4) |
| a |  |  |  |  |
| b |  |  |  |  |
| i |  |  |  |  |
| $ |  |  |  |  |

Avem conflict, deci din păcate, factorizarea la stânga nu ne-a ajutat.

1. Discutați, împreună cu cadrul didactic, cum se poate modifica tabelul de analiză astfel încât să se elimine conflictele.

În urma discuției, am convenit ca else-ul să țină doar de cel mai apropiat if, deci eliminăm cealaltă variantă:

Tabelul de analiză LL(1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | i | $ |
| S | (aSA,1) | err | (i,2) |  |
| A |  | (bS,3) ~~(e,4)~~ | - | (e,4) |
| a | pop |  |  |  |
| b |  | pop |  |  |
| i |  |  | pop |  |
| $ |  |  |  | acc |

d) Folosind analizorul LL(1) dacă secvența: if c then if c then stmt else stmt (sau echivalenta ei scrisa cu a,b,c,i) aparține limbajului generat de gramatică.

(aaibi$,S$,e) |- (push 1) (aaibi$, aSA$,1) |- (pop) (aibi$, SA$,1) |- (push 1) (aibi$, aSAA$,11) |- (pop)   
(ibi$, SAA$,11) |- (push 2) (ibi$, iAA$,112) |- (pop) (bi$, AA$,112) |- (push 3) (bi$, bSA$,1123) |- (pop)(i$, SA$,1123) |- (push 2)(i$, iA$,11232) |- (pop) ($, A$,11232) |- (push 4) ($,$,112324) |- (acc) acc  
Þ aaibiÎ L(G) (Da), și șirul producțiilor utilizate pentru a obține aaibi este 112324

1. Dându-se gramatica:

E → T + E | T

T → T \* F | F

F → (E) | a

a) Verificați dacă gramatica este LL(1).

b) Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

c) Folosind un analizor descendent verificați dacă secvența: a+a aparține limbajului generat de gramatică.

...

1. Fie gramatica:

List → id

List → id sep List

a) Verificați dacă gramatica este LL(1).

b) Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

1. Fie gramatica:

S → begin SList end

S → stmt

SList → S

SList → S ; SList

a) Verificați dacă gramatica este LL(1).

b) Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

...

1. Fie gramatica:

S → begin SList end (1)

S → stmt (2)

SList → S (3)

SList → S ; SList (4)

a) Verificați dacă gramatica este LL(1).

b) Încercați să transformați gramatica în una echivalentă LL(1) aplicând factorizarea la stânga. Verificați dacă noua gramatică este LL(1).

|  |  |  |
| --- | --- | --- |
|  | FIRST1 | FOLLOW1 |
| S | begin, stmt | $, ;, end |
| SList | begin, stmt | end |

Tabelul LL(1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | begin | end | stmt | ; | $ |
| S | (begin SList end,1) |  | (stmt,2) |  |  |
| SList | (S,3)  (S ; SList,4) |  | (S,3)  (S ; SList,4) |  |  |
| begin |  |  |  |  |  |
| end |  |  |  |  |  |
| stmt |  |  |  |  |  |
| ; |  |  |  |  |  |
| $ |  |  |  |  |  |

Avem conflicte, nu e de tip LL(1)