LL elemzések

Fordítóprogramok előadás (A,C,T szakirány)

LL elemzések

- felülről lefelé elemzés
- alapötlet: k terminális szimbólum előreolvasásával döntünk az alkalmazandó szabályról
- név: Left to right, using a Leftmost derivation
  (balról jobbra, legbal levezetéssel)

Ellenpélda

Nem minden grammatika esetén alkalmazható!

Nem elemezhető LL módszerrel

$S \to A \mid B$
$A \to a \mid aA$
$B \to ab \mid aBb$

Ellenpélda

Nem minden grammatika esetén alkalmazható!

Nem elemezhető LL módszerrel

$S \to A \mid B$
$A \to a \mid aA$
$B \to ab \mid aBb$

Akárhogy rögzítjük az előreolvasandó szimbólumok számát (k), a

$$\underbrace{aa\ldots a}_{k}\ldots$$

inputra nem tudjuk eldönteni, hogy $S \to A$ vagy $S \to B$ a jó választás.

FIRST halmazok

$FIRST_k(\alpha)$: az α mondatformából levezethető terminális sorozatok k hosszúságú kezdőszeletei

(ha a sorozat hossza kisebb mint k, akkor az egész sorozat eleme $FIRST_k(\alpha)$-nak, akár $\epsilon \in FIRST_k(\alpha)$ is előfordulhat)

FIRST halmazok

$FIRST_k(\alpha)$: az α mondatformából levezethető terminális sorozatok k hosszúságú kezdőszeletei

(ha a sorozat hossza kisebb mint k, akkor az egész sorozat eleme $FIRST_k(\alpha)$-nak, akár $\epsilon \in FIRST_k(\alpha)$ is előfordulhat)

Definíció: $FIRST_k(\alpha)$

$$FIRST_k(\alpha) = \{x \mid \alpha \Rightarrow^* x\beta \wedge |x| = k\} \cup \{x \mid \alpha \Rightarrow^* x \wedge |x| < k\}$$

$LL(k)$ grammatikák

$LL(k)$ grammatika: a levezetés tetszőleges pontján a szöveg következő k terminálisa meghatározza az alkalmazandó levezetési szabályt

$LL(k)$ grammatikák

$LL(k)$ grammatika: a levezetés tetszőleges pontján a szöveg következő k terminálisa meghatározza az alkalmazandó levezetési szabályt

Definíció: $LL(k)$ grammatika

Tetszőleges

$$S \Rightarrow^* wA\beta \Rightarrow w\alpha_1\beta \Rightarrow^* wx$$

$$S \Rightarrow^* wA\beta \Rightarrow w\alpha_2\beta \Rightarrow^* wy$$

levezetéspárra $FIRST_k(x) = FIRST_k(y)$ esetén $\alpha_1 = \alpha_2$.

LL elemzések

- példa szoftverek:
  - ANTLR parser generátor: http://www.antlr.org/
  - Coco/R: http://www.ssw.uni-linz.ac.at/coco/
- gyakorlatban: az $LL(1)$ elemzést könnyű megvalósítani
  - egyszerű $LL(1)$
  - epszilonmentes $LL(1)$
  - általános

Egyszerű $LL(1)$ nyelvtan

Definíció: Egyszerű $LL(1)$

Olyan $LL(1)$ grammatika, amelyben a szabályok jobboldala terminális szimbólummal kezdődik (ezért ϵ-mentes is).
(Az összes szabály $A \to a\alpha$ alakú.)

Egyszerű $LL(1)$ nyelvtan

Definíció: Egyszerű $LL(1)$

Olyan $LL(1)$ grammatika, amelyben a szabályok jobboldala terminális szimbólummal kezdődik (ezért ϵ-mentes is).
(Az összes szabály $A \to a\alpha$ alakú.)

Következmény: Egyszerű $LL(1)$ grammatika esetén az azonos nemterminálishoz tartozó szabályok jobboldalai különböző terminálissal kezdődnek.

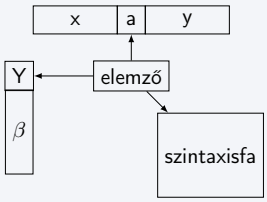
Tétel

Egy grammatika pontosan akkor egyszerű $LL(1)$, ha

- csak $A \to a\alpha$ alakú szabályai vannak
- és ha $A \to a_1\alpha_1$ és $A \to a_2\alpha_2$ két különböző szabály, akkor $a_1 \neq a_2$.

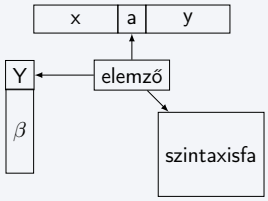
(A fordítóprogramok könyvben ez szerepel definícióként!)

Egyszerű $LL(1)$ elemzés



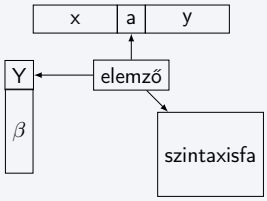
- xay: bemenet
- $Y\beta$: aktuális mondatforma (veremben tároljuk)

Egyszerű $LL(1)$ elemzés



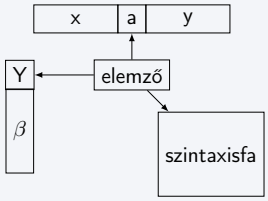
- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba

Egyszerű $LL(1)$ elemzés



- ha a verem tetején nemterminális szimbólum (A) van:
  - ha van $A \to a\alpha$ szabály: A helyére $a\alpha$ és bejegyzés a szintaxisfába
  - különben: hiba

Egyszerű $LL(1)$ elemzés



- ha a verem üres:
  - ha a szöveg végére értünk: OK
  - különben hiba

Elemző táblázat

Az egyes akciók táblázatba foglalhatók.

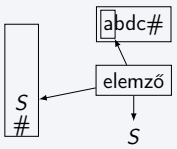
Példa nyelvtan

$S \to aS$ (1) | bAc (2)
$A \to bAc$ (3) | d (4)

- #: veremalja és fájlvége
- üres helyek: hiba

	a	b	c	d	#
S	$S \to^{(1)} aS$	$S \to^{(2)} bAc$			
A		$A \to^{(3)} bAc$		$A \to^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

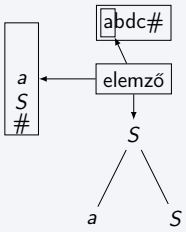


- $(abdc\#, S\#, \epsilon)$

Példa: abdc

	a	b	c	d	#
S	$S \to^{(1)} aS$	$S \to^{(2)} bAc$			
A		$A \to^{(3)} bAc$		$A \to^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

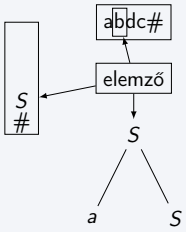


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

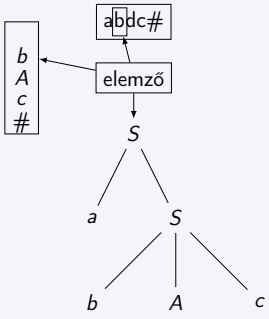


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

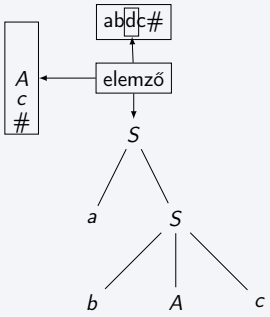


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$
- $(bdc\#, bAc\#, 12)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

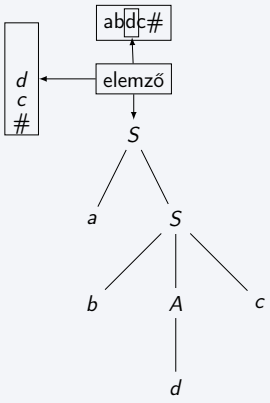


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$
- $(bdc\#, bAc\#, 12)$
- $(dc\#, Ac\#, 12)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

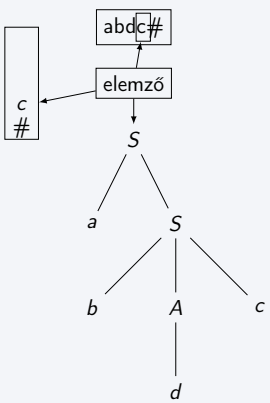


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$
- $(bdc\#, bAc\#, 12)$
- $(dc\#, Ac\#, 12)$
- $(dc\#, dc\#, 124)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

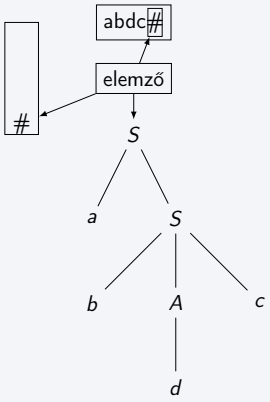


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$
- $(bdc\#, bAc\#, 12)$
- $(dc\#, Ac\#, 12)$
- $(dc\#, dc\#, 124)$
- $(c\#, c\#, 124)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc

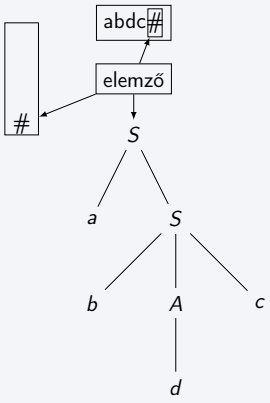


- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$
- $(bdc\#, bAc\#, 12)$
- $(dc\#, Ac\#, 12)$
- $(dc\#, dc\#, 124)$
- $(c\#, c\#, 124)$
- $(\#, \#, 124)$

Példa: abdc

	a	b	c	d	#
S	$S \rightarrow^{(1)} aS$	$S \rightarrow^{(2)} bAc$			
A		$A \rightarrow^{(3)} bAc$		$A \rightarrow^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

Példa: abdc



- $(abdc\#, S\#, \epsilon)$
- $(abdc\#, aS\#, 1)$
- $(bdc\#, S\#, 1)$
- $(bdc\#, bAc\#, 12)$
- $(dc\#, Ac\#, 12)$
- $(dc\#, dc\#, 124)$
- $(c\#, c\#, 124)$
- $(\#, \#, 124)$
- OK

ϵ-mentes LL(1) nyelvtan

Definíció: ϵ-mentes LL(1)

Olyan $LL(1)$ grammatika, amely ϵ-mentes.
(Nincs $A \rightarrow \epsilon$ szabály.)

ϵ-mentes LL(1) nyelvtan

Definíció: ϵ-mentes LL(1)

Olyan $LL(1)$ grammatika, amely ϵ-mentes.
(Nincs $A \rightarrow \epsilon$ szabály.)

Következmény: ϵ-mentes $LL(1)$ grammatika esetén az egy nemterminálishoz tartozó szabályok jobboldalainak $FIRST_1$ halmazai diszjunktak.

Tétel

Egy grammatika pontosan akkor ϵ-mentes $LL(1)$, ha

- ϵ-mentes
- és ha $A \rightarrow \alpha_1$ és $A \rightarrow \alpha_2$ két különböző szabály, akkor $FIRST_1(\alpha_1) \cap FIRST_1(\alpha_2) = \emptyset$.

(A fordítóprogramok könyvben ez szerepel definícióként.)

ϵ-mentes $LL(1)$ elemzés

- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba

ϵ-mentes $LL(1)$ elemzés

- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba
- ha a verem tetején nemterminális szimbólum (A) van (és a szövegben a következik):
  - ha van $A \to \alpha$ szabály, amelyre $a \in FIRST_1(\alpha)$: A helyére α és bejegyzés a szintaxisfába
  - különben: hiba

ϵ-mentes $LL(1)$ elemzés

- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba
- ha a verem tetején nemterminális szimbólum (A) van (és a szövegben a következik):
  - ha van $A \to \alpha$ szabály, amelyre $a \in FIRST_1(\alpha)$: A helyére α és bejegyzés a szintaxisfába
  - különben: hiba
- ha a verem üres:
  - ha a szöveg végére értünk: OK
  - különben hiba

ϵ-mentes $LL(1)$ elemző táblázat

- ugyanolyan szerkezetű, mint az egyszerű $LL(1)$-es
- az $A \to \alpha$ szabályt az A sorába és a $FIRST_1(\alpha)$ elemeinek oszlopaiba kell beírni

Példa nyelvtan és a FIRST halmazok

$S \to aS \mid A$
$A \to bAc \mid d$

$FIRST_1(aS) = \{a\}$ $FIRST_1(A) = \{b, d\}$
$FIRST_1(bAc) = \{b\}$ $FIRST_1(d) = \{d\}$

	a	b	c	d	#
S	$S \to^{(1)} aS$	$S \to^{(2)} A$		$S \to^{(2)} A$	
A		$A \to^{(3)} bAc$		$A \to^{(4)} d$	
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

$LL(1)$

Definíció: $LL(1)$ grammatika

Tetszőleges

$$S \Rightarrow^* wA\beta \Rightarrow w\alpha_1\beta \Rightarrow^* wx$$

$$S \Rightarrow^* wA\beta \Rightarrow w\alpha_2\beta \Rightarrow^* wy$$

levezetéspárra $FIRST_1(x) = FIRST_1(y)$ esetén $\alpha_1 = \alpha_2$.

$LL(1)$

Definíció: $LL(1)$ grammatika

Tetszőleges

$$S \Rightarrow^* wA\beta \Rightarrow w\alpha_1\beta \Rightarrow^* wx$$

$$S \Rightarrow^* wA\beta \Rightarrow w\alpha_2\beta \Rightarrow^* wy$$

levezetéspárra $FIRST_1(x) = FIRST_1(y)$ esetén $\alpha_1 = \alpha_2$.

Probléma:
$A \to \alpha_1 \mid ... \mid \alpha_n$ szabályban előfordulhat, hogy $\alpha_i \Rightarrow^* \epsilon$ valamelyik α_i-re.
Ezért az előreolvasott szimbólum a $S \Rightarrow^* wA\beta$ levezetésesetén a β-ból származhat.
Meg kell tehát vizsgálni azt is, hogy milyen terminálisok követhetik az A-t!

$FOLLOW$ halmazok

$FOLLOW_k(\alpha)$: a levezetésekben az α után előforduló k hosszúságú terminális sorozatok

(ha a sorozat hossza kisebb mint k, akkor az egész sorozat eleme $FOLLOW_k(\alpha)$-nak,
ha α után vége lehet a szövegnek, akkor $\# \in FOLLOW_k(\alpha)$)

$FOLLOW$ halmazok

$FOLLOW_k(\alpha)$: a levezetésekben az α után előforduló k hosszúságú terminális sorozatok

(ha a sorozat hossza kisebb mint k, akkor az egész sorozat eleme $FOLLOW_k(\alpha)$-nak,
ha α után vége lehet a szövegnek, akkor $\# \in FOLLOW_k(\alpha)$)

Definíció: $FOLLOW_k(\alpha)$

$$FOLLOW_k(\alpha) = \{x \mid S \Rightarrow^* \beta\alpha\gamma \wedge x \in FIRST_k(\gamma) \setminus \{\epsilon\}\} \cup \{\# \mid S \Rightarrow^* \beta\alpha\}$$

$LL(1)$ elemzést megalapozó tétel

Tétel

Egy grammatika pontosan akkor $LL(1)$ grammatika, ha bármely $A \to \alpha_1$ és $A \to \alpha_2$ különböző szabályok esetén $FIRST_1(\alpha_1 FOLLOW_1(A)) \cap FIRST_1(\alpha_2 FOLLOW_1(A)) = \emptyset$.

$FIRST_1(\alpha FOLLOW_1(A))$ jelentése: α-hoz egyenként konkatenáljuk $FOLLOW_1(A)$ elemeit és az így kapott halmaz minden elemére alkalmazzuk a $FIRST_1$ függvényt.

Példa

Példa nyelvtan és a FIRST, FOLLOW halmazok

$S \to aS \mid A$
$A \to bAc \mid d \mid \epsilon$

$FOLLOW_1(S) = \{\#\}$
$FOLLOW_1(A) = \{c, \#\}$

$FIRST_1(aS\ FOLLOW_1(S)) = \{a\}$
$FIRST_1(A\ FOLLOW_1(S)) = \{b, d, \#\}$

$FIRST_1(bAc\ FOLLOW_1(A)) = \{b\}$
$FIRST_1(d\ FOLLOW_1(A)) = \{d\}$
$FIRST_1(\epsilon\ FOLLOW_1(A)) = \{c, \#\}$

$LL(1)$ elemző táblázat

Mint az egyszerű és az ϵ-mentes $LL(1)$ esetén. Az $A \to \alpha$ szabályt az A sorába és a $FIRST_1(\alpha FOLLOW_1(A))$ halmaz oszlopaiba kell írni.

	a	b	c	d	#
S	$S \to^{(1)} aS$	$S \to^{(2)} A$		$S \to^{(2)} A$	$S \to^{(2)} A$
A		$A \to^{(3)} bAc$	$A \to^{(5)} \epsilon$	$A \to^{(4)} d$	$A \to^{(5)} \epsilon$
a	pop				
b		pop			
c			pop		
d				pop	
#					OK

$LL(1)$ elemzés

- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba

$LL(1)$ elemzés

- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba
- ha a verem tetején nemterminális szimbólum (A) van (és a szövegben a következik):
  - ha van $A \to \alpha$ szabály, amelyre $a \in FIRST_1(\alpha FOLLOW_1(A))$: A helyére α és bejegyzés a szintaxisfába
  - különben: hiba

$LL(1)$ elemzés

- ha a verem tetején terminális szimbólum van:
  - ha egyezik a szöveg következő karakterével: pop és lépés a szövegben
  - különben: hiba
- ha a verem tetején nemterminális szimbólum (A) van (és a szövegben a következik):
  - ha van $A \to \alpha$ szabály, amelyre $a \in FIRST_1(\alpha FOLLOW_1(A))$: A helyére α és bejegyzés a szintaxisfába
  - különben: hiba
- ha a verem üres:
  - ha a szöveg végére értünk: OK
  - különben hiba

Rekurzív leszállás

- az (általános) $LL(1)$ elemzés egy másik implementációja
- minden nemterminálishoz egy eljárást készítünk
- az eljáráshívásokon keresztül a futási idejű verem valósítja meg az elemzés vermét

Az *elfogad* eljárás

A terminális szimbólumok ellenőrzéséhez:

```
void elfogad( terminalis t )
{
  if( aktualis_szimbolum == t )
    aktualis_szimbolum = lexikalis_elemzo.kovetkezo();
  else
    hiba(...);
}
```

(A lexikális elemző *kovetkezo* függvénye visszaadja a soron következő lexikális elemet.)

$A \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n$ programja

```
void A()
{
  if(aktualis_szimbolum ∈ FIRST_1(α_1 FOLLOW_1(A)))
  {
    // alfa_1 programja
  }
  ...
  else if(aktualis_szimbolum ∈ FIRST_1(α_n FOLLOW_1(A)))
  {
    // alfa_n programja
  }
  else
  {
    hiba(...);
  }
}
```


A szabályalternatívák programja

Az $\alpha = \epsilon$ alternatíva programja a „skip”.

Az $\alpha = X_1X_2...X_n$ alternatíva programja az X_1, X_2,...,X_n szimbólumokhoz tartozó utasítások szekvenciája.

- ha $X_i = a$ (terminális), akkor az X_i-hez tartozó utasítás

  ```
  elfogad(a);
  ```

- ha $X_i = B$ (nemterminális), akkor

  ```
  B();
  ```


Hibakezelés

- hiba detektálása esetén:
  - segítő hibajelzést kell adni
  - meg kell próbálni folytatni az elemzést (az elemzőt szinkronizálni kell a bemenettel)
- a szinkronizáció szimbólumok kihagyását jelenti egy olyan szimbólumig, ahonnan folytatni lehet az elemzést
  - „pánik módszer”: az elemző pánikszerűen menekül a hiba helyétől
- például: a rekurzív leszállás eljárásai elején megvizsgálhatjuk, hogy megfelelő-e az `aktualis_szimbolum`, és szükség esetén addig ugorjuk át a szimbólumokat a bemeneten, amíg megfelelő nem lesz