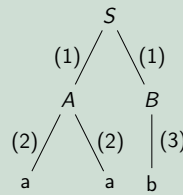


LR elemzések (az LR(0) elemzés)

Fordítóprogramok előadás (A,C,T szakirány)

Emlékeztető: elemzési irányok

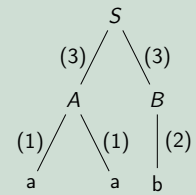
Felülről lefelé



$S \rightarrow^{(1)} AB \rightarrow^{(2)} aaB \rightarrow^{(3)} aab$

- Ez egy *legbal* levezetés!

Alulról felfelé



$aab \leftarrow^{(1)} Ab \leftarrow^{(2)} AB \leftarrow^{(3)} S$

- Ez egy *legjobb* levezetés inverze!

Emlékeztető: alulról felfelé elemzések

- Az elemzendő szöveg összetartozó részeit helyettesítjük nemterminális szimbólumokkal (redukció) és így alulról, a kezdőszimbólum felé építjük a fát.
- Fő kérdés: „Mit redukáljunk?”

Emlékeztető: alulról felfelé elemzési stratégiák

- *visszalépéses keresés (backtrack)*: ha nem sikerül eljutni a startszimbólumig, lépünk vissza, és válasszunk másik redukciót
⇒ visszalépéses alulról felfelé elemzések (nem tananyag)
 - lassú
 - ha hibás a szöveg, az csak túl későn derül ki

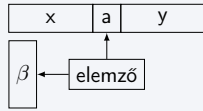
Emlékeztető: alulról felfelé elemzési stratégiák

- *visszalépéses keresés (backtrack)*: ha nem sikerül eljutni a startszimbólumig, lépünk vissza, és válasszunk másik redukciót
⇒ visszalépéses alulról felfelé elemzések (nem tananyag)
 - lassú
 - ha hibás a szöveg, az csak túl későn derül ki
- *precedenciák használata*: az egyes szimbólumok között adjunk meg precedenciarelációkat és ennek segítségével határozzuk meg a megfelelő redukciót
⇒ *precedencia elemzések* (nem tananyag)
 - ma már kevésbé használt
 - operátorokkal felépített kifejezések esetén természetes a használata

Emlékeztető: alulról felfelé elemzési stratégiák

- *visszalépéses keresés (backtrack)*: ha nem sikerül eljutni a startszimbólumig, lépünk vissza, és válasszunk másik redukciót
⇒ visszalépéses alulról felfelé elemzések (nem tananyag)
 - lassú
 - ha hibás a szöveg, az csak túl későn derül ki
- *precedenciák használata*: az egyes szimbólumok között adjunk meg precedenciarelációkat és ennek segítségével határozzuk meg a megfelelő redukciót
⇒ *precedencia elemzések* (nem tananyag)
 - ma már kevésbé használt
 - operátorokkal felépített kifejezések esetén természetes a használata
- *előreolvasás*: olvassunk előre a szövegben valahány szimbólumot, és az alapján döntsünk a redukcióról
⇒ *LR elemzések*
 - minden programozási nyelvhez lehet (LR) elemzőt készíteni
 - majdnem mindegyikhez lehet gyors (LALR) elemzőt készíteni

Az LR elemző felépítése



- xay : bemenet
- β : aktuális mondatforma egy része (veremben tároljuk)
- két lehetséges művelet: léptetés és redukálás
- *LR*: Left to right, using a Rightmost derivation (Balról jobbra, *legjobb* levezetéssel)

5

Fordítóprogramok előadás (A,C,T szakirány) LR elemzések (az LR(0) elemzés)

Léptetés



A bemenet következő szimbólumát a verem tetejére helyezzük.

6

Fordítóprogramok előadás (A,C,T szakirány) LR elemzések (az LR(0) elemzés)

Redukálás

Az $A \rightarrow \alpha$ szabály szerinti redukció.



A verem tetején lévő szabály-jobbaldalt helyettesítjük a megfelelő nemterminális szimbólummal.

7

Fordítóprogramok előadás (A,C,T szakirány) LR elemzések (az LR(0) elemzés)

Kiegészített grammatika

- Az elemzés végét arról fogjuk felismerni, hogy egy redukció eredménye a kezdőszimbólum lett.
- Ez csak akkor lehet, ha a kezdőszimbólum nem fordul elő a szabályok jobboldalán.

8

Fordítóprogramok előadás (A,C,T szakirány) LR elemzések (az LR(0) elemzés)

Kiegészített grammatika

- Az elemzés végét arról fogjuk felismerni, hogy egy redukció eredménye a kezdőszimbólum lett.
- Ez csak akkor lehet, ha a kezdőszimbólum nem fordul elő a szabályok jobboldalán.
- Ezt nem minden grammatika teljesíti, de mindegyik kiegészíthető:
 - legyen S' az új kezdőszimbólum
 - legyen $S' \rightarrow S$ egy új szabály
- az *LR* elemzésekhez mindig kiegészített nyelvtanokat fogunk használni

8

Fordítóprogramok előadás (A,C,T szakirány) LR elemzések (az LR(0) elemzés)

Mit kell redukálni?

9

Fordítóprogramok előadás (A,C,T szakirány) LR elemzések (az LR(0) elemzés)

Mit kell redukálni?

- **Egyszerű részmondat** (emlékeztető): α a $\gamma\alpha\beta$ mondatforma egyszerű részmondata, ha $S \Rightarrow^* \gamma A \beta \Rightarrow \gamma \alpha \beta$.

Mit kell redukálni?

- **Egyszerű részmondat** (emlékeztető): α a $\gamma\alpha\beta$ mondatforma egyszerű részmondata, ha $S \Rightarrow^* \gamma A \beta \Rightarrow \gamma \alpha \beta$.
- **Nyel:** a mondatformában a legbaloldalibb egyszerű részmondat.

Mit kell redukálni?

- **Egyszerű részmondat** (emlékeztető): α a $\gamma\alpha\beta$ mondatforma egyszerű részmondata, ha $S \Rightarrow^* \gamma A \beta \Rightarrow \gamma \alpha \beta$.
- **Nyel:** a mondatformában a legbaloldalibb egyszerű részmondat.
- **Épp a nyelet kell megtalálni a redukcióhoz.**

A nyelv meghatározása

- **Probléma: „Mi a nyelv?”**
 - léptetni vagy redukálni kell?
 - ha több lehetőség is van, melyik szabály szerint kell redukálni?

A nyelv meghatározása

- **Probléma: „Mi a nyelv?”**
 - léptetni vagy redukálni kell?
 - ha több lehetőség is van, melyik szabály szerint kell redukálni?
- **LR(k) grammatika:** k szimbólum előreolvasásával eldönthető, hogy mi legyen az elemzés következő lépése.

Magyarázat az LR(k) definíciójához

- Tegyük fel, hogy léptetésekkel és redukálásokkal eljutottunk az $\alpha\beta w$ mondatformához, és itt β a nyelv: $S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta w$.

Magyarázat az LR(k) definíciójához

- Tegyük fel, hogy léptetésekkal és redukálásokkal eljutottunk az $\alpha\beta w$ mondatformához, és itt β a nyél: $S \Rightarrow^* \alpha A w \Rightarrow \alpha\beta w$.
- Tegyük fel, hogy egy ugyanígy kezdődő mondatforma, az $\alpha\beta y$ felbontható $\alpha\beta y = \gamma\delta x$ módon, és ebben δ a nyél, azaz $S \Rightarrow^* \gamma B x \Rightarrow \gamma\delta x$.

Magyarázat az LR(k) definíciójához

- Tegyük fel, hogy léptetésekkal és redukálásokkal eljutottunk az $\alpha\beta w$ mondatformához, és itt β a nyél: $S \Rightarrow^* \alpha A w \Rightarrow \alpha\beta w$.
- Tegyük fel, hogy egy ugyanígy kezdődő mondatforma, az $\alpha\beta y$ felbontható $\alpha\beta y = \gamma\delta x$ módon, és ebben δ a nyél, azaz $S \Rightarrow^* \gamma B x \Rightarrow \gamma\delta x$.
- Az LR(k) tulajdonság azt mondja, hogy w-ből és y-ból előreolvasva k szimbólumot, egyértelműen eldönthető az elemzés következő lépése.

Magyarázat az LR(k) definíciójához

- Tegyük fel, hogy léptetésekkal és redukálásokkal eljutottunk az $\alpha\beta w$ mondatformához, és itt β a nyél: $S \Rightarrow^* \alpha A w \Rightarrow \alpha\beta w$.
- Tegyük fel, hogy egy ugyanígy kezdődő mondatforma, az $\alpha\beta y$ felbontható $\alpha\beta y = \gamma\delta x$ módon, és ebben δ a nyél, azaz $S \Rightarrow^* \gamma B x \Rightarrow \gamma\delta x$.
- Az LR(k) tulajdonság azt mondja, hogy w-ből és y-ból előreolvasva k szimbólumot, egyértelműen eldönthető az elemzés következő lépése.
- Ezért ha $FIRST_k(w) = FIRST_k(y)$, akkor $\alpha\beta w$ és $\alpha\beta y$ esetén is ugyanazt kell csinálni:
 - mivel $\alpha\beta w$ esetén az $A \rightarrow \beta$ szabály szerint redukáltunk,
 - ugyanezt kellett csinálni $\alpha\beta y$ esetén is,
 - vagyis $\alpha = \gamma$, $\beta = \delta$, $A = B$ és $y = x$.

LR(k) definíciója

Definíció: LR(k) grammatika

Egy kiegészített grammatika LR(k) grammatika ($k \geq 0$), ha $S \Rightarrow^* \alpha A w \Rightarrow \alpha\beta w$
 $S \Rightarrow^* \gamma B x \Rightarrow \gamma\delta x$
 $\alpha\beta y = \gamma\delta x$ és $FIRST_k(w) = FIRST_k(y)$ esetén
 $\alpha = \gamma$, $\beta = \delta$ és $A = B$.

LR(k) definíciója

Definíció: LR(k) grammatika

Egy kiegészített grammatika LR(k) grammatika ($k \geq 0$), ha $S \Rightarrow^* \alpha A w \Rightarrow \alpha\beta w$
 $S \Rightarrow^* \gamma B x \Rightarrow \gamma\delta x$
 $\alpha\beta y = \gamma\delta x$ és $FIRST_k(w) = FIRST_k(y)$ esetén
 $\alpha = \gamma$, $\beta = \delta$ és $A = B$.

LR(0) grammatika: előreolvasás nélkül eldönthető az elemzés következő lépése.

Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

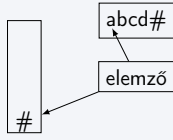
 $S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$

Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$



Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$



Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$



Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$

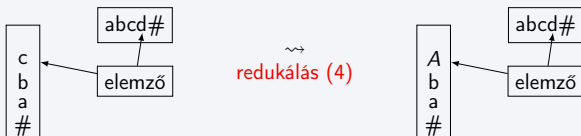


Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$

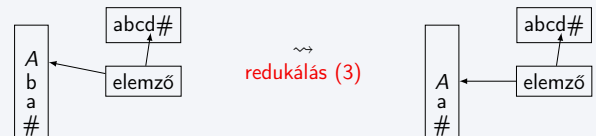


Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$



Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$

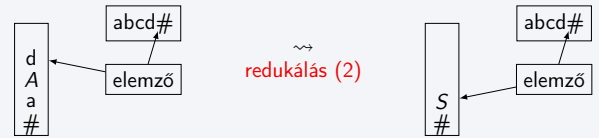


Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$



Példa

Grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

$S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$

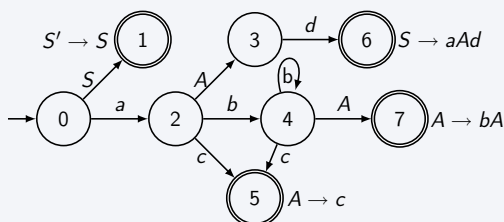


Léptetés vagy redukálás?

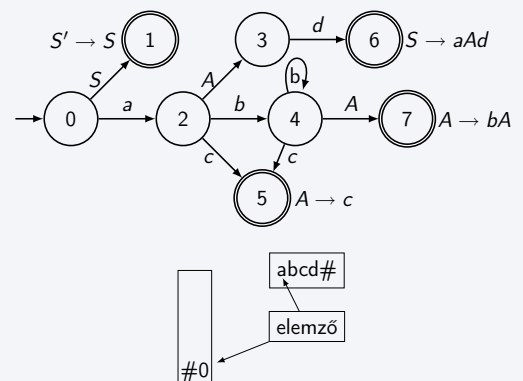
- LR(0) elemzés: előreolvasás nélkül kell döntünk!

Léptetés vagy redukálás?

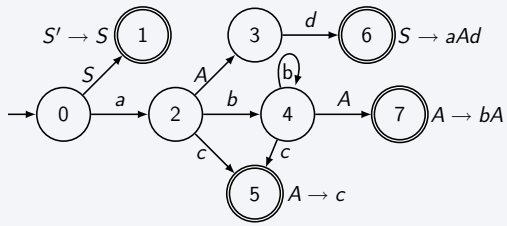
- LR(0) elemzés: előreolvasás nélkül kell döntünk!
- Ötlet: készítsünk egy véges determinisztikus automatát
 - az átmeneteit a verembe kerülő szimbólumok határozzák meg
 - lépteteskör terminális
 - redukáláskör nemterminális
 - amikor elfogadó állapotba jut, akkor kell redukálni



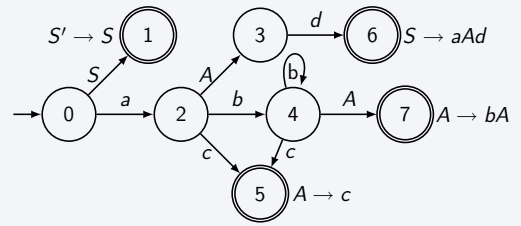
Példa



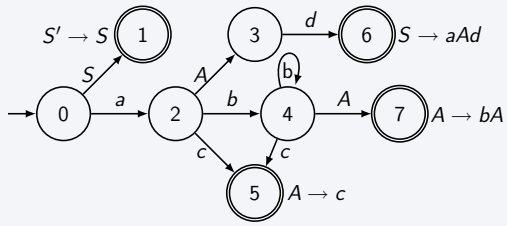
Példa



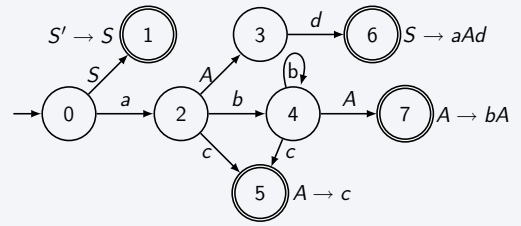
Példa



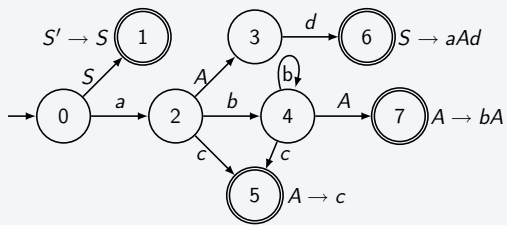
Példa



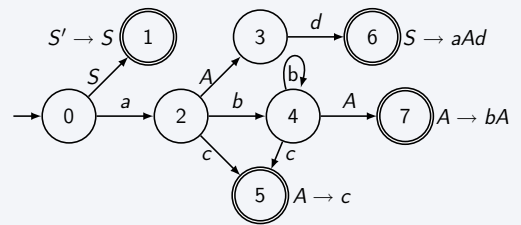
Példa



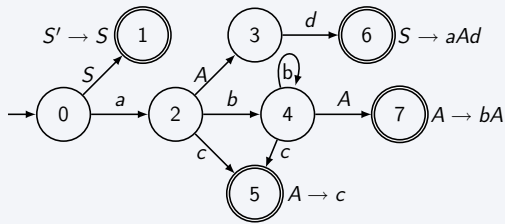
Példa



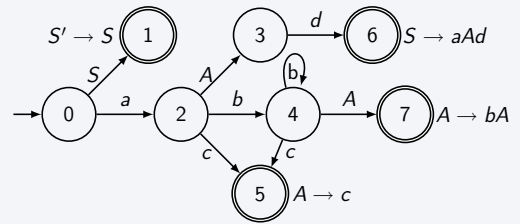
Példa



Példa



Példa



LR(0) elemek

- Hogyan építsük fel az automatát?

LR(0) elemek

- Hogyan építsük fel az automatát?
- Az automata állapotai azt mutatják meg, hogy melyik szabály építésében hol tartunk.
- Például: $[S \rightarrow a.Ad]$ jelentse azt, hogy
 - az a szimbólumot már elemeztük,
 - az Ad rész még hátra van.

LR(0) elemek

- Hogyan építsük fel az automatát?
- Az automata állapotai azt mutatják meg, hogy melyik szabály építésében hol tartunk.
- Például: $[S \rightarrow a.Ad]$ jelentse azt, hogy
 - az a szimbólumot már elemeztük,
 - az Ad rész még hátra van.

Definíció: LR(0) elem

Ha $A \rightarrow \alpha$ a grammatika egy helyettesítési szabálya, akkor az $\alpha = \alpha_1\alpha_2$ tetszőleges felbontás esetén $[A \rightarrow \alpha_1.\alpha_2]$ a grammatika egy LR(0) eleme.

LR(0) elemek

- Hogyan építsük fel az automatát?
- Az automata állapotai azt mutatják meg, hogy melyik szabály építésében hol tartunk.
- Például: $[S \rightarrow a.Ad]$ jelentse azt, hogy
 - az a szimbólumot már elemeztük,
 - az Ad rész még hátra van.

Definíció: LR(0) elem

Ha $A \rightarrow \alpha$ a grammatika egy helyettesítési szabálya, akkor az $\alpha = \alpha_1\alpha_2$ tetszőleges felbontás esetén $[A \rightarrow \alpha_1.\alpha_2]$ a grammatika egy LR(0) eleme.

- Ha a szabály jobb oldala n szimbólumot tartalmaz, akkor $n + 1$ darab LR(0)-elem tartozik hozzá.

$$A \rightarrow .aAd, A \rightarrow a.Ad, A \rightarrow aA.d, A \rightarrow aA\dot{d}$$

A lezárás művelet

- Az automata egy állapotához több $LR(0)$ -elem is tartozhat.
 - Ezeket a halmazokat fogjuk *kanonikus halmazoknak* hívni.

A lezárás művelet

- Az automata egy állapotához több $LR(0)$ -elem is tartozhat.
 - Ezeket a halmazokat fogjuk *kanonikus halmazoknak* hívni.
- Milyen $LR(0)$ elemek tartoznak egy halmazba?
 - Például: Ha $[A \rightarrow a.Ad]$ állapotban vagyunk, akkor az $A \rightarrow bA$ és $A \rightarrow c$ szabályokat kezdhethetjük építeni, azaz $[A \rightarrow .bA]$ és $[A \rightarrow .c]$ is hozzátartozik a halmazhoz.

A lezárás művelet

- Az automata egy állapotához több $LR(0)$ -elem is tartozhat.
 - Ezeket a halmazokat fogjuk *kanonikus halmazoknak* hívni.
- Milyen $LR(0)$ elemek tartoznak egy halmazba?
 - Például: Ha $[A \rightarrow a.Ad]$ állapotban vagyunk, akkor az $A \rightarrow bA$ és $A \rightarrow c$ szabályokat kezdhethetjük építeni, azaz $[A \rightarrow .bA]$ és $[A \rightarrow .c]$ is hozzátartozik a halmazhoz.

Definíció: lezárás (closure)

Ha \mathcal{I} a grammatika egy $LR(0)$ elemhalmaza, akkor $closure(\mathcal{I})$ a *legsűkebb* olyan halmaz, amely az alábbi tulajdonságokkal rendelkezik:

- $\mathcal{I} \subseteq closure(\mathcal{I})$
- ha $[A \rightarrow \alpha.B\gamma] \in closure(\mathcal{I})$, és $B \rightarrow \beta$ a grammatika egy szabálya, akkor $[B \rightarrow .\beta] \in closure(\mathcal{I})$.

Az olvasás művelet

- Hogyan lépünk át az automata egyik állapotából a másikba?

Az olvasás művelet

- Hogyan lépünk át az automata egyik állapotából a másikba?
- Ha $[A \rightarrow a.Ad]$ állapotban vagyunk, és A kerül a verem tetejére, akkor $[A \rightarrow aA.d]$ állapotba jutunk.

Az olvasás művelet

- Hogyan lépünk át az automata egyik állapotából a másikba?
- Ha $[A \rightarrow a.Ad]$ állapotban vagyunk, és A kerül a verem tetejére, akkor $[A \rightarrow aA.d]$ állapotba jutunk.

Definíció: olvasás (read)

Ha \mathcal{I} a grammatika egy $LR(0)$ elemhalmaza, X pedig terminális vagy nemterminális szimbóluma, akkor $read(\mathcal{I}, X)$ a *legsűkebb* olyan halmaz, amely az alábbi tulajdonsággal rendelkezik:

- ha $[A \rightarrow \alpha.X\beta] \in \mathcal{I}$, akkor $closure([A \rightarrow \alpha.X\beta]) \subseteq read(\mathcal{I}, X)$.

LR(0) kanonikus halmazok

Definíció: LR(0) kanonikus halmazok

- 1 $closure([S' \rightarrow .S])$ a grammatika egy kanonikus halmaza.
- 2 Ha \mathcal{I} a grammatika egy kanonikus elemhalmaza, X egy terminális vagy nemterminális szimbóluma, és $read(\mathcal{I}, X)$ nem üres, akkor $read(\mathcal{I}, X)$ is a grammatika egy kanonikus halmaza.
- 3 Az első két szabállyal az összes kanonikus halmaz előáll.

35

Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

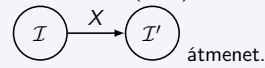
LR(0) kanonikus halmazok

Definíció: LR(0) kanonikus halmazok

- 1 $closure([S' \rightarrow .S])$ a grammatika egy kanonikus halmaza.
- 2 Ha \mathcal{I} a grammatika egy kanonikus elemhalmaza, X egy terminális vagy nemterminális szimbóluma, és $read(\mathcal{I}, X)$ nem üres, akkor $read(\mathcal{I}, X)$ is a grammatika egy kanonikus halmaza.
- 3 Az első két szabállyal az összes kanonikus halmaz előáll.

Az automata felépítése:

- A $closure([S' \rightarrow .S])$ legyen a kezdőállapot.
- Ha $\mathcal{I}' = read(\mathcal{I}, X)$, akkor az automatában legyen



átmenet.

- A végállapotok azok a kanonikus halmazok, amelyekben olyan elemek vannak, ahol a pont a szabály végén van.

35

Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

35

Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$



$$\mathcal{I}_0 = closure([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$$

36

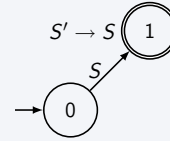
Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$



$$\mathcal{I}_0 = closure([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$$

$$\mathcal{I}_1 = read(\mathcal{I}_0, S) = closure([S' \rightarrow S]) = \{[S' \rightarrow S]\}$$

37

Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

37

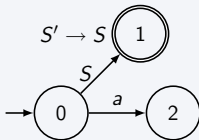
Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$



$$\mathcal{I}_0 = closure([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$$

$$\mathcal{I}_1 = read(\mathcal{I}_0, S) = \{[S' \rightarrow S]\}$$

$$\mathcal{I}_2 = read(\mathcal{I}_0, a) = closure([S \rightarrow a.Ad])$$

$$= \{[S \rightarrow a.Ad], [A \rightarrow .bA], [A \rightarrow .c]\}$$

38

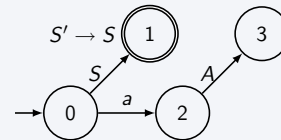
Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$



$$\mathcal{I}_0 = closure([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$$

$$\mathcal{I}_1 = read(\mathcal{I}_0, S) = \{[S' \rightarrow S]\}$$

$$\mathcal{I}_2 = read(\mathcal{I}_0, a) = \{[S \rightarrow a.Ad], [A \rightarrow .bA], [A \rightarrow .c]\}$$

$$\mathcal{I}_3 = read(\mathcal{I}_2, A) = closure([S \rightarrow aA.d]) = \{[S \rightarrow aA.d]\}$$

39

Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

39

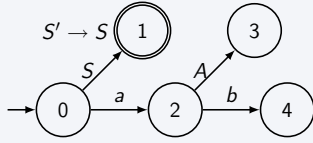
Fordítóprogramok előadás (A,C,T szakirány)

LR elemzések (az LR(0) elemzés)

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

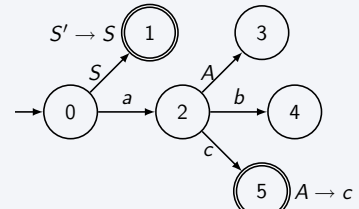


$\mathcal{I}_0 = \text{closure}([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$
 $\mathcal{I}_1 = \text{read}(\mathcal{I}_0, S) = \{[S' \rightarrow S]\}$
 $\mathcal{I}_2 = \text{read}(\mathcal{I}_0, a) = \{[S \rightarrow a.Ad], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_3 = \text{read}(\mathcal{I}_2, A) = \{[S \rightarrow aA.d]\}$
 $\mathcal{I}_4 = \text{read}(\mathcal{I}_2, b) = \text{closure}([A \rightarrow b.A])$
 $= \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\}$

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

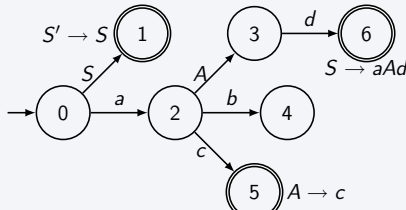


$\mathcal{I}_0 = \text{closure}([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$
 $\mathcal{I}_1 = \text{read}(\mathcal{I}_0, S) = \{[S' \rightarrow S]\}$
 $\mathcal{I}_2 = \text{read}(\mathcal{I}_0, a) = \{[S \rightarrow a.Ad], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_3 = \text{read}(\mathcal{I}_2, A) = \{[S \rightarrow aA.d]\}$
 $\mathcal{I}_4 = \text{read}(\mathcal{I}_2, b) = \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_5 = \text{read}(\mathcal{I}_2, c) = \text{closure}([A \rightarrow c.]) = \{[A \rightarrow c.]\}$

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

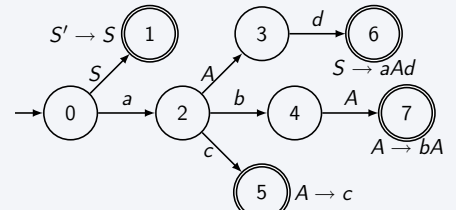


$\mathcal{I}_0 = \text{closure}([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$
 $\mathcal{I}_1 = \text{read}(\mathcal{I}_0, S) = \{[S' \rightarrow S]\}$
 $\mathcal{I}_2 = \text{read}(\mathcal{I}_0, a) = \{[S \rightarrow a.Ad], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_3 = \text{read}(\mathcal{I}_2, A) = \{[S \rightarrow aA.d]\}$
 $\mathcal{I}_4 = \text{read}(\mathcal{I}_2, b) = \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_5 = \text{read}(\mathcal{I}_2, c) = \{[A \rightarrow c.]\}$
 $\mathcal{I}_6 = \text{read}(\mathcal{I}_3, d) = \text{closure}([S \rightarrow aAd.]) = \{[S \rightarrow aAd.]\}$

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

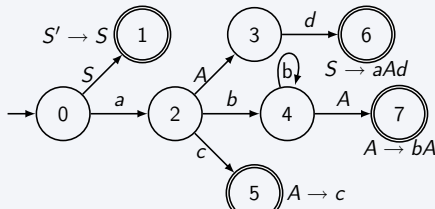


$\mathcal{I}_0 = \text{closure}([S' \rightarrow .S]) = \{[S' \rightarrow .S], [S \rightarrow .aAd]\}$
 $\mathcal{I}_1 = \text{read}(\mathcal{I}_0, S) = \{[S' \rightarrow S]\}$
 $\mathcal{I}_2 = \text{read}(\mathcal{I}_0, a) = \{[S \rightarrow a.Ad], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_3 = \text{read}(\mathcal{I}_2, A) = \{[S \rightarrow aA.d]\}$
 $\mathcal{I}_4 = \text{read}(\mathcal{I}_2, b) = \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_5 = \text{read}(\mathcal{I}_2, c) = \{[A \rightarrow c.]\}$
 $\mathcal{I}_6 = \text{read}(\mathcal{I}_3, d) = \{[S \rightarrow aAd.]\}$
 $\mathcal{I}_7 = \text{read}(\mathcal{I}_4, A) = \text{closure}([A \rightarrow b.A]) = \{[A \rightarrow b.A]\}$

Példa

Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$

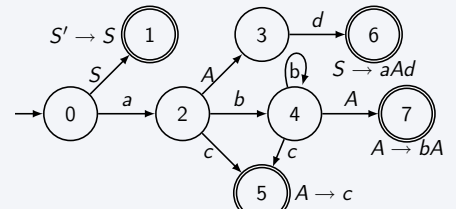


$\mathcal{I}_4 = \text{read}(\mathcal{I}_2, b) = \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_5 = \text{read}(\mathcal{I}_2, c) = \{[A \rightarrow c.]\}$
 \dots
 $\text{read}(\mathcal{I}_4, b) = \text{closure}([A \rightarrow b.A])$
 $= \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\} = \mathcal{I}_4$

Példa

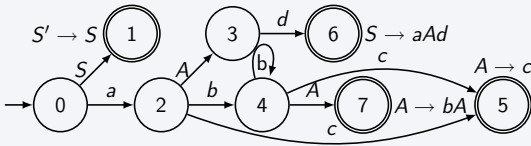
Példa grammatika

- 1 $S' \rightarrow S$
- 2 $S \rightarrow aAd$
- 3 $A \rightarrow bA$
- 4 $A \rightarrow c$



$\mathcal{I}_4 = \text{read}(\mathcal{I}_2, b) = \{[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]\}$
 $\mathcal{I}_5 = \text{read}(\mathcal{I}_2, c) = \{[A \rightarrow c.]\}$
 \dots
 $\text{read}(\mathcal{I}_4, b) = \mathcal{I}_4$
 $\text{read}(\mathcal{I}_4, c) = \text{closure}([A \rightarrow c.]) = \{[A \rightarrow c.]\} = \mathcal{I}_5$

LR(0) elemző táblázat



állapot	akció	S	A	a	b	c	d
0	léptetés	1		2			
1	OK						
2	léptetés		3		4	5	
3	léptetés						6
4	léptetés		7		4	5	
5	redukálás ($A \rightarrow c$)						
6	redukálás ($S \rightarrow aAd$)						
7	redukálás ($A \rightarrow bA$)						

Véges-e az elemző létrehozása?

- a grammatikának véges sok szabálya van
- véges sok LR(0) eleme van

Véges-e az elemző létrehozása?

- a grammatikának véges sok szabálya van
- véges sok LR(0) eleme van
- a *closure* függvény kiszámítása véges sok lépésben befejeződik
- a *read* függvény kiszámítása véges sok lépésben befejeződik

Véges-e az elemző létrehozása?

- a grammatikának véges sok szabálya van
- véges sok LR(0) eleme van
- a *closure* függvény kiszámítása véges sok lépésben befejeződik
- a *read* függvény kiszámítása véges sok lépésben befejeződik
- a véges sok LR(0)-elem hatványhalmaza is véges
- a lehetséges kanonikus halmazok száma is véges

Véges-e az elemző létrehozása?

- a grammatikának véges sok szabálya van
- véges sok LR(0) eleme van
- a *closure* függvény kiszámítása véges sok lépésben befejeződik
- a *read* függvény kiszámítása véges sok lépésben befejeződik
- a véges sok LR(0)-elem hatványhalmaza is véges
- a lehetséges kanonikus halmazok száma is véges

Az elemző táblázat (az automata) létrehozása véges sok lépésben befejeződik.

Helyes-e az elemző?

- Az automata pontosan akkor jut-e végállapotba, ha redukálni kell?
- A megfelelő a redukciót írja-e elő?

Járható prefix

Definíció: járható prefix

Ha az $\alpha\beta x$ mondatforma nyele β , akkor az $\alpha\beta$ prefixeit az $\alpha\beta x$ járható prefixeinek nevezzük.

Járható prefix

Definíció: járható prefix

Ha az $\alpha\beta x$ mondatforma nyele β , akkor az $\alpha\beta$ prefixeit az $\alpha\beta x$ járható prefixeinek nevezzük.

- A járható prefixeket olvassuk végig a nyelv végének eléréséhez.
- Példa: $S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$
Az $abAd$ mondatforma nyele a bA .
A járható prefixei: a, ab, abA

Járható prefix

Definíció: járható prefix

Ha az $\alpha\beta x$ mondatforma nyele β , akkor az $\alpha\beta$ prefixeit az $\alpha\beta x$ járható prefixeinek nevezzük.

- A járható prefixeket olvassuk végig a nyelv végének eléréséhez.
- Példa: $S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abcd$
Az $abAd$ mondatforma nyele a bA .
A járható prefixei: a, ab, abA
- **Maximális járható prefix:** olyan járható prefix, amihez nem lehet újabb szimbólumot hozzávenni, hogy járható prefixet kapjunk.
- Egy járható prefix épp akkor maximális, ha a végén van a nyelv.

Járható prefixre érvényes LR(0) elemek

Definíció: járható prefixre érvényes LR(0) elemek

A grammatika egy $[A \rightarrow \alpha.\beta]$ LR(0)-eleme érvényes a $\gamma\alpha$ járható prefixre nézve, ha

$$S' \Rightarrow^* \gamma Ax \Rightarrow \gamma \alpha \beta x$$

Járható prefixre érvényes LR(0) elemek

Definíció: járható prefixre érvényes LR(0) elemek

A grammatika egy $[A \rightarrow \alpha.\beta]$ LR(0)-eleme érvényes a $\gamma\alpha$ járható prefixre nézve, ha

$$S' \Rightarrow^* \gamma Ax \Rightarrow \gamma \alpha \beta x$$

- Az érvényes LR(0) elemek az adott járható prefix „lehetséges folytatásait” adják meg.
- Példa: az ab járható prefixre érvényes LR(0)-elemek:
 $[A \rightarrow b.A], [A \rightarrow .bA], [A \rightarrow .c]$
 $S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd$
 $S' \Rightarrow S \Rightarrow aAd \Rightarrow abAd \Rightarrow abbAd$
 $S' \Rightarrow S \Rightarrow aAd \Rightarrow abcd$
- A maximális járható prefixekre érvényes LR(0) elemek azok, ahol a pont a szabály végén van.

Az LR(0) elemzés helyessége

Tétel: az LR(0) elemzés nagy tétele

Egy γ járható prefix hatására az elemző automatája a kezdőállapotból olyan állapotba kerül, amelyhez tartozó kanonikus halmaz éppen a γ járható prefixre érvényes LR(0) elemeket tartalmazza.

Az $LR(0)$ elemzés helyessége

Tétel: az $LR(0)$ elemzés nagy tétele

Egy γ járható prefix hatására az elemző automatája a kezdőállapotból olyan állapotba kerül, amelyhez tartozó kanonikus halmaz éppen a γ járható prefixre érvényes $LR(0)$ elemeket tartalmazza.

- Az elemző tehát addig fog léptetést előírni, amíg a veremben lévő járható prefix maximális nem lesz.
- Ha a járható prefix maximális, akkor az elemző redukálást ír elő.

Konfliktusok a táblázatban

- Hol használtuk ki, hogy $LR(0)$ a grammatika?

Konfliktusok a táblázatban

- Hol használtuk ki, hogy $LR(0)$ a grammatika?
- **Konfliktus:** Ha egy \mathcal{I}_k kanonikus halmaz alapján nem lehet egyértelműen eldönteni, hogy az adott állapotban milyen akciót kell végrehajtani.
 - léptetés/redukálás konfliktus: az egyik elem léptetést, egy másik redukálást ír elő
 - redukálás/redukálás konfliktus: az egyik elem az egyik szabály szerinti, a másik egy másik szabály szerinti redukciót ír elő

Konfliktusok a táblázatban

- Hol használtuk ki, hogy $LR(0)$ a grammatika?
- **Konfliktus:** Ha egy \mathcal{I}_k kanonikus halmaz alapján nem lehet egyértelműen eldönteni, hogy az adott állapotban milyen akciót kell végrehajtani.
 - léptetés/redukálás konfliktus: az egyik elem léptetést, egy másik redukálást ír elő
 - redukálás/redukálás konfliktus: az egyik elem az egyik szabály szerinti, a másik egy másik szabály szerinti redukciót ír elő
- **Az $LR(0)$ tulajdonság biztosítja a táblázat konfliktusmentes kitöltését!**