Fordítóprogramok 2/4. feladat megoldása

Kaposi Ambrus
http://akaposi.web.elte.hu
kaposi.ambrus@gmail.com

2012. március 2.

Feladat

Elemzőt szeretnénk készíteni egy nyelvhez, aminek egy példaprogramja:

```
if ?a1 then
  ?a1 := false
endif
while ?a3 do
  ?a3 := ?a2
  ?a2 := ?a1
done
```

A nyelvben minden változó `?` jellel kezdődik, deklarálni nem kell őket, mindegyik logikai típusú (`true`, `false`). A nyelvben van értékadás, elágazás és ciklus. Az értékadás baloldalán változó, jobboldalán változó, `true` vagy `false` állhat. A ciklusok és elágazások feltétele mindig egyetlen változó. Az egész program és a ciklusok, elágazások törzse is lehet üres.

(a) Határozd meg a lexikális elemeket és írj hozzájuk reguláris kifejezéseket!

(b) Írd fel a nyelvtan szabályait!

(c) Ellenőrizd, hogy LL(1)-e a nyelvtan!

(d) Készíts hozzá rekurzív leszállásos elemzőt!

Megoldás

(a) Határozd meg a lexikális elemeket és írj hozzájuk reguláris kifejezéseket!

Lexikális elemek: if, then, endif, while, do, done, true, false, $:=$, $\underbrace{\backslash?[\_0-9a-zA-Z]+}_{v}$

(b) Írd fel a nyelvtan szabályait!

$S \to \overset{1}{IS} \mid \overset{2}{WS} \mid \overset{3}{ES} \mid \overset{4}{\epsilon}$

$I \to \underline{if}\ v\ \overset{5}{\underline{then}}\ S\ \underline{endif}$

$W \to \underline{while}\ v\ \overset{6}{\underline{do}}\ S\ \underline{done}$

$E \to v\ \overset{7}{:=}\ A$

$A \to \overset{8}{v} \mid \overset{9}{\underline{true}} \mid \overset{10}{\underline{false}}$

(c) Ellenőrizd, hogy LL(1)-e a nyelvtan!

Környezetfüggetlen, nincs benne végtelen rekurzió.

Nem minden szabály-jobboldal kezdődik terminális szimbólummal, ezért nem lehet egyszerű LL(1).

A 4-es szabály ϵ-szabály, ezért nem ϵ-mentes LL(1).

Ahhoz, hogy általános LL(1)-nyelvtan legyen, teljesülnie kell, hogy minden $A \to \alpha_1 \mid \alpha_2$ szabály esetén a $FIRST_1(\alpha_1 FOLLOW_1(A)) \cap FIRST_1(\alpha_2 FOLLOW_1(A)) = \emptyset$ legyen (a szabály-jobboldalak $FIRST_1(...FOLLOW_1(...))$ halmazai páronként diszjunktak legyenek). Nézzük meg ezeket!

- S:
  $H_1 = FIRST_1(IS\ FOLLOW_1(S)) = FIRST_1(\underline{if}\ v\ \underline{then}\ S\ \underline{endif}\ S\ FOLLOW_1(S)) = \{\underline{if}\}$
  $H_2 = FIRST_1(WS\ FOLLOW_1(S)) = FIRST_1(\underline{while}\ v\ \underline{do}\ S\ \underline{done}\ S\ FOLLOW_1(S)) = \{\underline{while}\}$
  $H_3 = FIRST_1(ES\ FOLLOW_1(S)) = FIRST_1(v\ \underline{:=}\ A\ S\ FOLLOW_1(S)) = \{v\}$
  $H_4 = FIRST_1(FOLLOW_1(S)) = FIRST_1(\{\#\} \cup FOLLOW_1(S) \cup \{\underline{endif}\}FOLLOW_1(I) \cup \{\underline{done}\}FOLLOW_1(W)) = \{\#, \underline{endif}, \underline{done}\}$
  (Megjegyzés H_4 kiszámolásához: $FOLLOW_1(S)$-ben # mindig szerepel, hiszen S a kezdőszimbólum.)
- I:
  $H_5 = FIRST_1(\underline{if}\ v\ \underline{then}\ S\ \underline{endif}\ FOLLOW_1(I)) = \{\underline{if}\}$
- W:
  $H_6 = FIRST_1(\underline{while}\ v\ \underline{do}\ S\ \underline{done}\ FOLLOW_1(W)) = \{\underline{while}\}$
- E:
  $H_7 = FIRST_1(v\ \underline{:=}\ A\ FOLLOW_1(E)) = \{v\}$
- A:
  $H_8 = FIRST_1(v\ FOLLOW_1(A)) = \{v\}$
  $H_9 = FIRST_1(\underline{true}\ FOLLOW_1(A)) = \{\underline{true}\}$
  $H_{10} = FIRST_1(\underline{false}\ FOLLOW_1(A)) = \{\underline{false}\}$

Megállapítjuk, hogy:

- S: $H_1 \cap H_2 = \emptyset, H_1 \cap H_3 = \emptyset, H_1 \cap H_4 = \emptyset, H_2 \cap H_3 = \emptyset, H_2 \cap H_4 = \emptyset, H_3 \cap H_4 = \emptyset$
- A: $H_8 \cap H_9 = \emptyset, H_8 \cap H_{10} = \emptyset, H_9 \cap H_{10} = \emptyset$

Tehát páronként diszjunktak a megfelelő halmazok, ez a nyelv tehát LL(1)-elemezhető.

Megcsináljuk az elemző táblázatot is:

	if	then	endif	while	do	done	true	false	:=	v	#
S	1 $S \to IS$		4 $S \to \epsilon$	2 $S \to WS$		4 $S \to \epsilon$				3 $S \to ES$	4 $S \to \epsilon$
I	5 $I \to \underline{if}\ v\ \underline{then}\ S\ \underline{endif}$										
W				6 $W \to \underline{while}\ v\ \underline{do}\ S\ \underline{done}$							
E										7 $E \to v := A$	
A							9 $A \to \underline{true}$	10 $A \to \underline{false}$		8 $A \to v$	
if	pop										
then		pop									
endif			pop								
while				pop							
do					pop						
done						pop					
true							pop				
false								pop			
:=									pop		
v										pop	
#											OK

Elemezzük az alábbi programot:

```
if ?a1 then ?a2 := false endif
```

A lexikális elemző az alábbit adja:

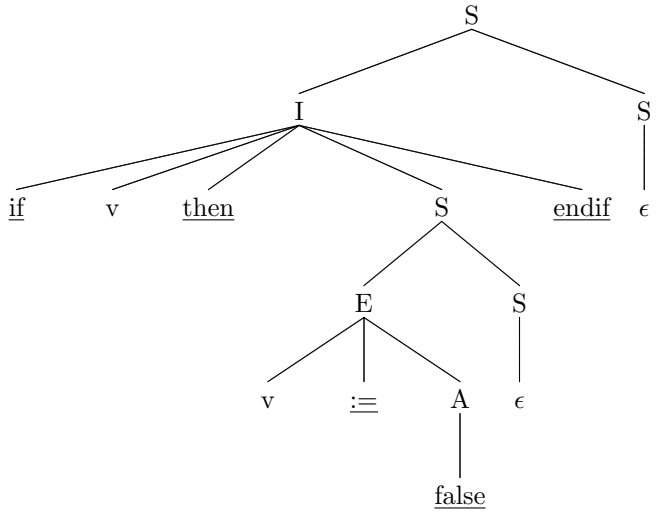
$\underline{if}\ v\ \underline{then}\ v\ \underline{:=}\ \underline{false}\ \underline{endif}$

Elemzés:

$(\underline{if}\ v\ \underline{then}\ v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ S\#,\ \epsilon) \overset{1}{\to}$
$(\underline{if}\ v\ \underline{then}\ v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ IS\#,\ 1) \overset{5}{\to}$
$(\underline{if}\ v\ \underline{then}\ v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ \underline{if}\ v\ \underline{then}\ S\ \underline{endif}\ S\#,\ 1\ 5) \overset{pop}{\to}$
$(v\ \underline{then}\ v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ v\ \underline{then}\ S\ \underline{endif}\ S\#,\ 1\ 5) \overset{pop}{\to}$

$(\underline{then}\ v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ \underline{then}\ S\ \underline{endif}\ S\#,\ 1\ 5) \overset{pop}{\rightarrow}$
$(v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ S\ \underline{endif}\ S\#,\ 1\ 5) \overset{3}{\rightarrow}$
$(v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ E\ S\ \underline{endif}\ S\#,\ 1\ 5\ 3) \overset{7}{\rightarrow}$
$(v\ \underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ v\ \underline{:=}\ A\ S\ \underline{endif}\ S\#,\ 1\ 5\ 3\ 7) \overset{pop}{\rightarrow}$
$(\underline{:=}\ \underline{false}\ \underline{endif}\ \#,\ \underline{:=}\ A\ S\ \underline{endif}\ S\#,\ 1\ 5\ 3\ 7) \overset{pop}{\rightarrow}$
$(\underline{false}\ \underline{endif}\ \#,\ A\ S\ \underline{endif}\ S\#,\ 1\ 5\ 3\ 7) \overset{10}{\rightarrow}$
$(\underline{false}\ \underline{endif}\ \#,\ \underline{false}\ S\ \underline{endif}\ S\#,\ 1\ 5\ 3\ 7\ 10) \overset{pop}{\rightarrow}$
$(\underline{endif}\ \#,\ S\ \underline{endif}\ S\#,\ 1\ 5\ 3\ 7\ 10) \overset{4}{\rightarrow}$
$(\underline{endif}\ \#,\ \underline{endif}\ S\#,\ 1\ 5\ 3\ 7\ 10\ 4) \overset{pop}{\rightarrow}$
$(\#,\ S\#,\ 1\ 5\ 3\ 7\ 10\ 4\ 4) \overset{4}{\rightarrow}$
$(\#,\ \#,\ 1\ 5\ 3\ 7\ 10\ 4\ 4) \rightarrow OK$

Szintaxisfa:



(d) Készíts hozzá rekurzív leszállásos elemzőt!

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

void S() {
  if (aktualis == 'if') {                                                    // 1
    I();
    S();
  } else if (aktualis == 'while') {                                          // 2
    W();
    S();
  } else if (aktualis == 'v') {                                              // 3
    E();
    S();
  } else if (aktualis == 'endif' || aktualis == 'done' || aktualis == '#') { // 4
    ;
  } else {
    hiba();
  }
}

void I() {
  if (aktualis == 'if') {           //  5
```

```
    elfogad('if');
    elfogad('v');
    elfogad('then');
    S();
    elfogad('endif');
  } else {
    hiba();
  }
}

void W() {
  if (aktualis == 'while') {        //  6
    elfogad('while');
    elfogad('v');
    elfogad('do');
    S();
    elfogad('done');
  } else {
    hiba();
  }
}

void E() {
  if (aktualis == 'v') {            //  7
    elfogad('v');
    elfogad(':=');
    A();
  } else {
    hiba();
  }
}

void A() {
  if if (aktualis == 'v') {         //  8
    elfogad('v');
  } else (aktualis == 'true') {     //  9
    elfogad('true');
  } else if (aktualis == 'false') { // 10
    elfogad('false');
  } else {
    hiba();
  }
}
```