

I KOLOVIJUM IZ PROGRAMSKIH PREVODILACA

1. Tip enum definisan je sledećom gramatikum:

EnumType → enum ID { ConstantList };
ConstantList → ConstantList , ConstantDefinition | ConstantDefinition
ConstantDefinition → ID | ID = CONST

Transformisati navedenu gramatiku u LL(1) gramatiku, dokazati da tako transformisana gramatika jeste tipa LL(1) i kreirati odgovarajuću sintaksnu tabelu. Korišćenjem kreirane tabele proveriti da li je sledeća definicija enum tipa korektno zapisana:

```
enum ID {  
    ID,  
    ID = CONST  
};
```

2. Kreirati LR sintaksnu tabelu za gramatiku datu u prvom zadatku i krišćenjem te tabele proveriti da li je primer dat u prvom zadatku korektno napisan.

I KOLOVIJUM IZ PROGRAMSKIH PREVODILACA

1. Tip enum definisan je sledećom gramatikum:

EnumType → enum ID { ConstantList };
ConstantList → ConstantList , ConstantDefinition | ConstantDefinition
ConstantDefinition → ID | ID = CONST

Transformisati navedenu gramatiku u LL(1) gramatiku, dokazati da tako transformisana gramatika jeste tipa LL(1) i kreirati odgovarajuću sintaksnu tabelu. Korišćenjem kreirane tabele proveriti da li je sledeća definicija enum tipa korektno zapisana:

```
enum ID {  
    ID,  
    ID = CONST  
};
```

2. Kreirati LR sintaksnu tabelu za gramatiku datu u prvom zadatku i krišćenjem te tabele proveriti da li je primer dat u prvom zadatku korektno napisan.

I KOLOVIJUM IZ PROGRAMSKIH PREVODILACA

1. Tip enum definisan je sledećom gramatikum:

EnumType → enum ID { ConstantList };
ConstantList → ConstantList , ConstantDefinition | ConstantDefinition
ConstantDefinition → ID | ID = CONST

Transformisati navedenu gramatiku u LL(1) gramatiku, dokazati da tako transformisana gramatika jeste tipa LL(1) i kreirati odgovarajuću sintaksnu tabelu. Korišćenjem kreirane tabele proveriti da li je sledeća definicija enum tipa korektno zapisana:

```
enum ID {  
    ID,  
    ID = CONST  
};
```

3. Kreirati LR sintaksnu tabelu za gramatiku datu u prvom zadatku i krišćenjem te tabele proveriti da li je primer dat u prvom zadatku korektno napisan.

1) $ET \rightarrow \text{enum ID } \{ CL \};$

$CL \rightarrow CL, CD \mid CD$

$CD \rightarrow ID \mid ID = \text{CONST}$

$$X = X \alpha / B \quad X = BX' \\ X' = \alpha X' / E$$

$$X \rightarrow Y \alpha / YB - X \rightarrow Y X'$$

$$X' \rightarrow \alpha / B$$

$\underbrace{CL \rightarrow CL, CD \mid CD}_{\downarrow} \\ \downarrow CL \rightarrow CD \quad CL'$

$CL' \rightarrow , CD \quad CL' \mid E$

$CD \rightarrow ID$

$\underbrace{CD \rightarrow ID = \text{CONST}}$

$* CD \rightarrow ID \quad CD'$

$CD' \rightarrow = \text{CONST} \mid E$



1) $ET \rightarrow \text{enum ID } \{ CL \};$

2) $CL \rightarrow CD \quad CL'$

3) $CL' \rightarrow , CD \quad CL'$

4) $CL' \rightarrow E$

5) $CD \rightarrow ID \quad CD'$

6) $CD' \rightarrow = \text{CONST}$

7) $CD' \rightarrow E$

3) $\text{FIRST}(, CD \quad CL') = \{ , \}$

4) $\text{FIRST}(E) = \{ E \}$

$$\nearrow n = \emptyset$$

$\text{Follow}(CL') = \{ , \}$

$\Rightarrow \text{FIRST } CL(1)$

6) ~~FIRST~~ $\text{FIRST}(= \text{CONST}) = \{ = \}$

7) $\text{FIRST}(E) = \{ E \}$

$\text{Follow}(CD') = \{ , \}$

$$\nearrow n = \emptyset$$



1) FIRST(enum ID { CL };) = { enum } - ET

2) FIRST(ID) = { ID } - CL

3) FIRST(, ID CL') = { , } \

4) FIRST(ϵ) = { ϵ } ✓ CL'

Follow(CL') = { , }

5) FIRST(ID CL') = { ID } - CD

6) FIRST(= const) = { = } ✓

7) FIRST(ϵ) = { ϵ } ✓ CD'

Follow(CD') = { , , } ✓

ID | { | } | ; | , | = | const | enum | #
ET (1)

CL (1,2)

CL' (1,4) (1,3)

CD (1,5)

CD' (1,7) (1,7) (1,6)

Follow

ID pop

{ pop

} pop

; pop

, pop

pop

pop

pop

const

enum

#

acc

enum ID {

 ID,
 ID
 ID = CONST

 ID
 ID
 ID

y,
++RAOMI STEK

ET

; } CL { ID enum

; } CL { ID

; } CL {

; } CL' CD

; } CL' CD' ID

; } CL' CD'

; } CL'

; } CL' CD ,

; } CL' CD

; } CL' CD' ID

; } CL' CD'

; } CL' CONST =

; } CL' CONST

; } CL'

; }

;

#

KOREKCIJA ZAPIS

ULAZNI SIMBOL

ACCYJA

enum

1

enum

pop

ID

pop

}

pop

ID

2

ID

5

ID

pop

,

7

,

3

,

pop

ID

5

ID

pop

=

6

=

pop

CONST

pop

y

4

y

pop

;

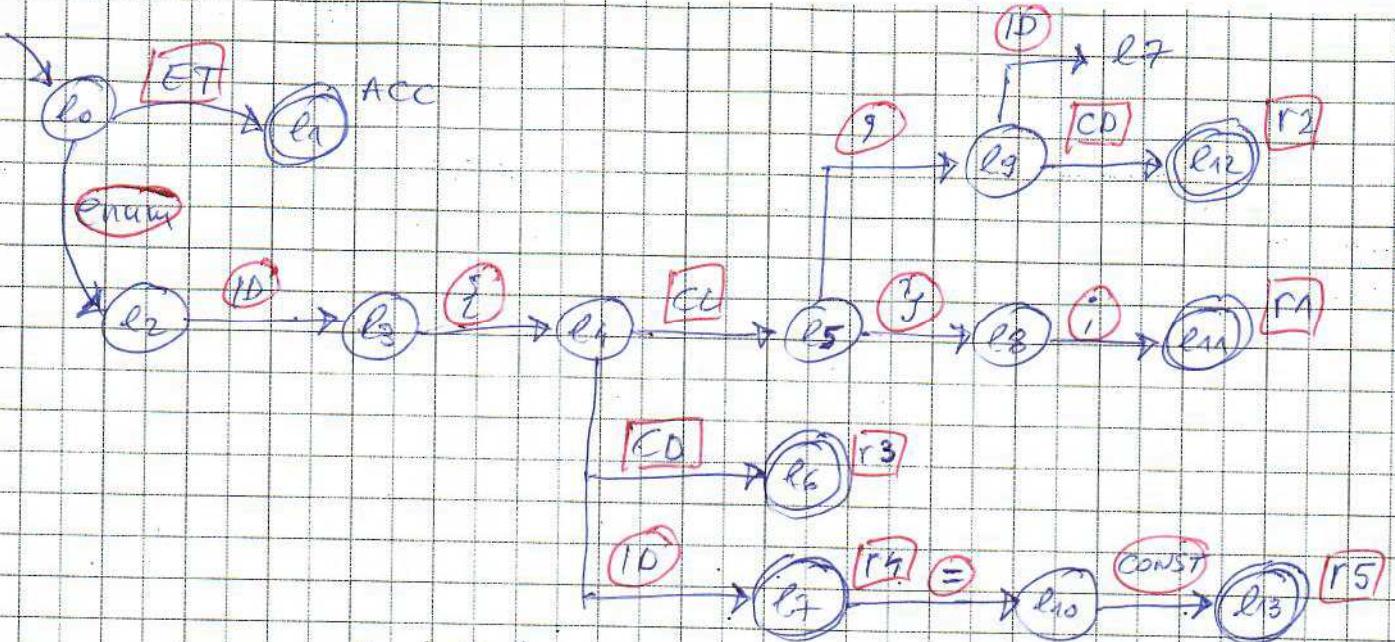
pop

#

[acc]

✓

- ②
 1) $ET \rightarrow \text{enum ID } \{ CL \}; \checkmark$ l6: goto (l4, CD)
 2) $CL \rightarrow CL, CD \checkmark$
 3) $CL \rightarrow CD \checkmark$
 4) $CD \rightarrow ID \checkmark$
 5) $CD \rightarrow ID = \text{CONST} \checkmark$
 $\overbrace{\quad\quad\quad}$
l0: $ET' \rightarrow ET$
 $ET \rightarrow . \text{enum ID } \{ CL \}; \checkmark$
 $\overbrace{\quad\quad\quad}$
l1: goto (l0, ET)
 $ET' \rightarrow ET. \text{ ACC}$
 $\overbrace{\quad\quad\quad}$
l2: goto (l0, enum)
 $ET \rightarrow \text{enum. ID } \{ CL \}; \checkmark$
 $\overbrace{\quad\quad\quad}$
l3: goto (l2, ID)
 $ET \rightarrow \text{enum ID. } \{ CL \}; \checkmark$
 $\overbrace{\quad\quad\quad}$
l4: goto (l3, ;)
 $ET \rightarrow \text{enum ID } \{ . CL \}; \checkmark$
 $CL \rightarrow . CL, CD \checkmark$
 $CL \rightarrow . CD \checkmark$
 $CD \rightarrow . ID \checkmark$
 $CD \rightarrow . ID = \text{CONST} \checkmark$
 $\overbrace{\quad\quad\quad}$
l5: goto (l4, CL)
 $ET \rightarrow \text{enum ID } \{ CL. \}; \checkmark$
 $CL \rightarrow CL., CD \checkmark$
- $CL \rightarrow CD.$ (r3)
- l7: goto (l4, ID)
 $CD \rightarrow ID.$ (r4)
- $CD \rightarrow ID. = \text{CONST} \checkmark$
l8: goto (l5, ;)
 $ET \rightarrow \text{enum ID } \{ CL \}. \checkmark$
- l9: goto (l5, ;)
 $CL \rightarrow CL. . CD \checkmark$
 $CD \rightarrow . ID \checkmark$
 $CD \rightarrow . ID = \text{CONST} \checkmark$
- l10: goto (l7, =)
 $CD \rightarrow ID = . \text{ CONST} \checkmark$
- l11: goto (l8, ;)
 $ET \rightarrow \text{enum ID } \{ CL \}. \checkmark$ (r1)
- l12: goto (l9, CD)
 $CL \rightarrow CL, CD.$ (r2)
- goto (l9, ID) = l7
- l13: goto (l10, CONST)
 $CD \rightarrow ID = \text{CONST}.$ (r5)



	enum	ID	{	}	:	;	,	=	CONST	#	ET	CC	CD
0		52								1			
1													acc
2			S3										
3				S4									
4					S7								
5						S8		S9					
6	r3						r3						
7	r4						r4		S10				
8								S11					
9			S7										12
10										S13			
11	r11												
12	r12					r2		r2					
13	r13						r5		r5				

Follow(ET) = {#}

Follow(CC) = {, ,)}

Follow(CD) = {, ,)}

V

RAONI STEK

MISTAKI SIMBOL ACCUR

O		enum	S2
O enum 2		1D	S3
O enum 2 1D 3	{ 4	5	S4
O enum 2 1D 3 { 4	1D		S7
O enum 2 1D 3 { 4 } 10 7,	9		r4 $(CD \rightarrow 1D)$ $(4, CD) = 6$
O enum 2 1D 3 { 4 } CD 6,	,		r3 $(CL \rightarrow CD)$ $(4, CL) = 5$
O enum 2 1D 3 { 4 CL 5	,		S9
O enum 2 1D 3 { 4 CL 5, 9	1D		S7
O enum 2 1D 3 { 4 CL 5, 9 10 7.	=		S10
O enum 2 1D 3 { 4 CL 5, 9 1D 7 = 10	CONST		S13
O enum 2 1D 3 { 4 CL 5, 9 1D 7 = 10 CONST 13,	3		r5 $(CD \rightarrow 1D = CONST)$ $(9, CD) = 12$
O enum 2 1D 3 { 4 CL 5, 9 CD 12,	3		r2 $(CL \rightarrow CL, CD)$ $(4, CL) = 5$
O enum 2 1D 3 { 4 CL 5	3		S8
O enum 2 1D 3 { 4 CL 5 } 8	;		S11
O enum 2 1D 3 { 4 CL 5 } 8 ; 1n,	#		r1 $(ET \rightarrow enum 1D \& CL 3 ;$ $(0, ET) = 1$
O ET 1	#		acc
KOREKCIJA ZAPIS			✓

↓ ↓ ↓
Enum ID ↓
↑ ↑
ID , ↓ ↓
↓ ↓ ID = const
y : ↓
↓

I kolokvijum iz Programskih prevodilaca

- Gramatiku G zadatu sledećim skupom smena:

$XPath \rightarrow XPath / XPathStep | XPathStep$

$XPathStep \rightarrow Axis \ ID$

$Axis \rightarrow AxisType :: | \epsilon$

$AxisType \rightarrow parent | child$

transformisati u LL(1) gramatiku, dokazati da tako transformisana gramatika jeste tipa LL(1), kreirati LL(1) sintaksnu tabelu i proveriti da li je sledeći XPath sintaksno ispravan:

parent :: ID / ID

- Kreirati LR sintraksnu tabelu za gramatiku definisanu u prvom zadatku i korišćenjem te tabele proveriti da li je XPath naveden u prethodnom zadatku sintaksno ispravan.

I kolokvijum iz Programskih prevodilaca

- Gramatiku G zadatu sledećim skupom smena:

$XPath \rightarrow XPath / XPathStep | XPathStep$

$XPathStep \rightarrow Axis \ ID$

$Axis \rightarrow AxisType :: | \epsilon$

$AxisType \rightarrow parent | child$

transformisati u LL(1) gramatiku, dokazati da tako transformisana gramatika jeste tipa LL(1), kreirati LL(1) sintaksnu tabelu i proveriti da li je sledeći XPath sintaksno ispravan:

parent :: ID / ID

- Kreirati LR sintraksnu tabelu za gramatiku definisanu u prvom zadatku i korišćenjem te tabele proveriti da li je XPath naveden u prethodnom zadatku sintaksno ispravan.

I kolokvijum iz Programskih prevodilaca

- Gramatiku G zadatu sledećim skupom smena:

$XPath \rightarrow XPath / XPathStep | XPathStep$

$XPathStep \rightarrow Axis \ ID$

$Axis \rightarrow AxisType :: | \epsilon$

$AxisType \rightarrow parent | child$

transformisati u LL(1) gramatiku, dokazati da tako transformisana gramatika jeste tipa LL(1), kreirati LL(1) sintaksnu tabelu i proveriti da li je sledeći XPath sintaksno ispravan:

parent :: ID / ID

- Kreirati LR sintraksnu tabelu za gramatiku definisanu u prvom zadatku i korišćenjem te tabele proveriti da li je XPath naveden u prethodnom zadatku sintaksno ispravan.

1. $\text{XPath} \rightarrow \text{XPath} / \text{XPathStep} \mid \text{XPathStep}$

$\text{XPathStep} \rightarrow \text{Axis ID}$

$\text{Axis} \rightarrow \text{AxisType} :: \mid \epsilon$

$\text{AxisType} \rightarrow \text{parent} \mid \text{child}$

$\text{XPath} \leftrightarrow \text{XP}$

$\text{XPathStep} \leftrightarrow \text{XPS}$

$\text{Axis} \leftrightarrow \text{A}$

$\text{AxisType} \leftrightarrow \text{AT}$

Transformacija:

1. $\text{XP} \rightarrow \text{XPS } \text{XP}' \quad \} \quad \begin{matrix} \text{jer po LL(1) gramatici ne moze} \\ \text{prvi simbol sa desne strane da} \\ \text{bude preduak simboli sa leve} \end{matrix}$
2. $\text{XP}' \rightarrow / \text{XPS } \text{XP}' \quad \}$
3. $\text{XP}' \rightarrow \epsilon$
4. $\text{A} \rightarrow \text{AT} ::$
5. $\text{A} \rightarrow \epsilon$
6. $\text{AT} \rightarrow \text{parent}$
7. $\text{AT} \rightarrow \text{child}$
8. $\text{XPS} \rightarrow \text{A ID}$

Proveravamo da li je LL(1) \rightarrow uporedujemo smene koje sa leve strane imaju isti simbol i proveravamo da li su prvi simboli sa desne strane preduaci

* [2. i 3.]

\rightarrow ukoliko jesu, treba izvrsiti levu faktorizaciju

$$\text{FIRST}(/ \text{XPS } \text{XP}') = \{ / \}$$

$$\text{FIRST}(\epsilon) = \epsilon$$

$$\text{FIRST}(/ \text{XPS } \text{XP}') \cap \text{FIRST}(\epsilon) = \emptyset$$

Pošto imamo preslikavanje u Σ , odredujemo

FOLLOW f-ju za simbol sa leve strane:

$$\text{FOLLOW}(XP') = \{\#\}$$

$$\text{FIRST}(\cdot / XPS XP') \cap \text{FOLLOW}(XP') = \emptyset$$

[4. i 5.]

$$\text{FIRST}(AT::) = \{\text{parent, child}\}$$

$$\text{FIRST}(E) = \{\epsilon\}$$

$$\text{FIRST}(AT::) \cap \text{FIRST}(\epsilon) = \emptyset$$

$$\text{FOLLOW}(A) = \{10\}$$

$$\text{FIRST}(AT::) \cap \text{FOLLOW}(A) = \emptyset$$

[6. i 7.]

$$\text{FIRST}(\text{parent}) = \{\text{parent}\}$$

$$\text{FIRST}(\text{child}) = \{\text{child}\}$$

$$\text{FIRST}(\text{parent}) \cap \text{FIRST}(\text{child}) = \emptyset$$



Kreisrunde Tabelle:

1. FIRST (XPS XP') = {parent, child, ID} XPS

2. FIRST (/XPS XP') = {/} XP'

3. FIRST (ϵ) = ϵ XP'

folgende FOLLOW (XP') = {#}

4. FIRST (AT ::) = {parent, child} A

5. FIRST (ϵ) = ϵ A

FOLLOW (A) = {ID}

6. FIRST (parent) \rightarrow {parent} AT

7. FIRST (child) = {child} AT

8. FIRST (A ID) = {parent, child, ID} XPS

	/	ID	:	parent	child	#
XP		(XPS XP', 1)		(XPS XP', 1)	(XPS XP', 1)	
XP'	/	(XPS XP', 2)				(E, 3)
XPS		(A ID, 8)		(A ID, 8)	(A ID, 8)	
A		(E, 5)		(AT ::, 4)	(AT ::, 4)	
AT				(parent, 4)	(child, 7)	
/	pop					
ID		pop				
::			pop			
parent				pop		
child					pop	
#						acc

Provera:

parent :: ID / FD

stek	ulaz	akcija
XP #	parent	smera 1
XPS XP' #	parent	smera 8
A ID XP' #	parent	smera 4
AT :: ID XP' #	parent	smera 6
parent :: ID XP' #	parent	pop
:: ID XP' #	::	pop
ID XP' #	ID	pop
XP' #	/	smera 2
/ XPS XP' #	/	pop
XPS XP' #	ID	smera 8
A ID XP' #	ID	smera 5

<u>stek</u>	<u>ulaz</u>	<u>akcija</u>	
ID	#	pop	
XP' #	#	smena 3	
#	#	<u>acc</u>	→ Izdati kod je
			sintaksno ispravan
			jer smo stigli
			do accept
			aticje!
			✓

$2. \quad XP \rightarrow XP/XPS \quad r_1$
 $XP \rightarrow XPS \quad r_2$
 $XPS \rightarrow A \text{ ID} \quad r_3$
 $A \rightarrow AT:: \quad r_4$
 $A \rightarrow \epsilon \quad r_5$
 $AT \rightarrow \text{parent} \quad r_6$
 $AT \rightarrow \text{child} \quad r_7$

(l0) $XP^1 \rightarrow .XP \quad \checkmark$ $A \rightarrow .AT :: \quad \checkmark$

$XP \rightarrow .XP/XPS \quad \checkmark$ $A \rightarrow .\epsilon \Leftrightarrow A \rightarrow \epsilon. \quad r_5$

$XP \rightarrow .XPS \quad \checkmark$ $AT \rightarrow .\text{parent}$
 $XPS \rightarrow .A \text{ ID} \quad \checkmark$ $AT \rightarrow .\text{child}$

(l1) = goto (l0, XP)

$XP^1 \rightarrow XP. \quad \boxed{\text{ace}}$

$XP \rightarrow XP./XPS \quad \checkmark$

(l2) = goto (l1, /)

$XP \rightarrow XP./.XPS \quad \checkmark$

$XPS \rightarrow .A \text{ ID} \quad \checkmark$

$A \rightarrow .AT :: \quad \checkmark$

$A \rightarrow .\epsilon \Leftrightarrow A \rightarrow \epsilon. \quad r_5$

$AT \rightarrow .\text{parent} \quad \checkmark$

$AT \rightarrow .\text{child} \quad \checkmark$

(l3) = goto (l2, XPS)

Xp → Xp / XPS. [r1]

(l4) = goto (l2, A)

XPS → A. ID

(l5) = goto (l4, ID)

XPS → A. ID.

[r3]

goto (lo, A) = l4

goto (lo, AT) = l6

goto (lo, parent) = l8

goto (lo, child) = l9

(l7) = goto (l6, ::)

A → AT ::

[r4]

(l8) = goto (l2, parent)

AT → parent. [r6]

(l9) = goto (l2, child)

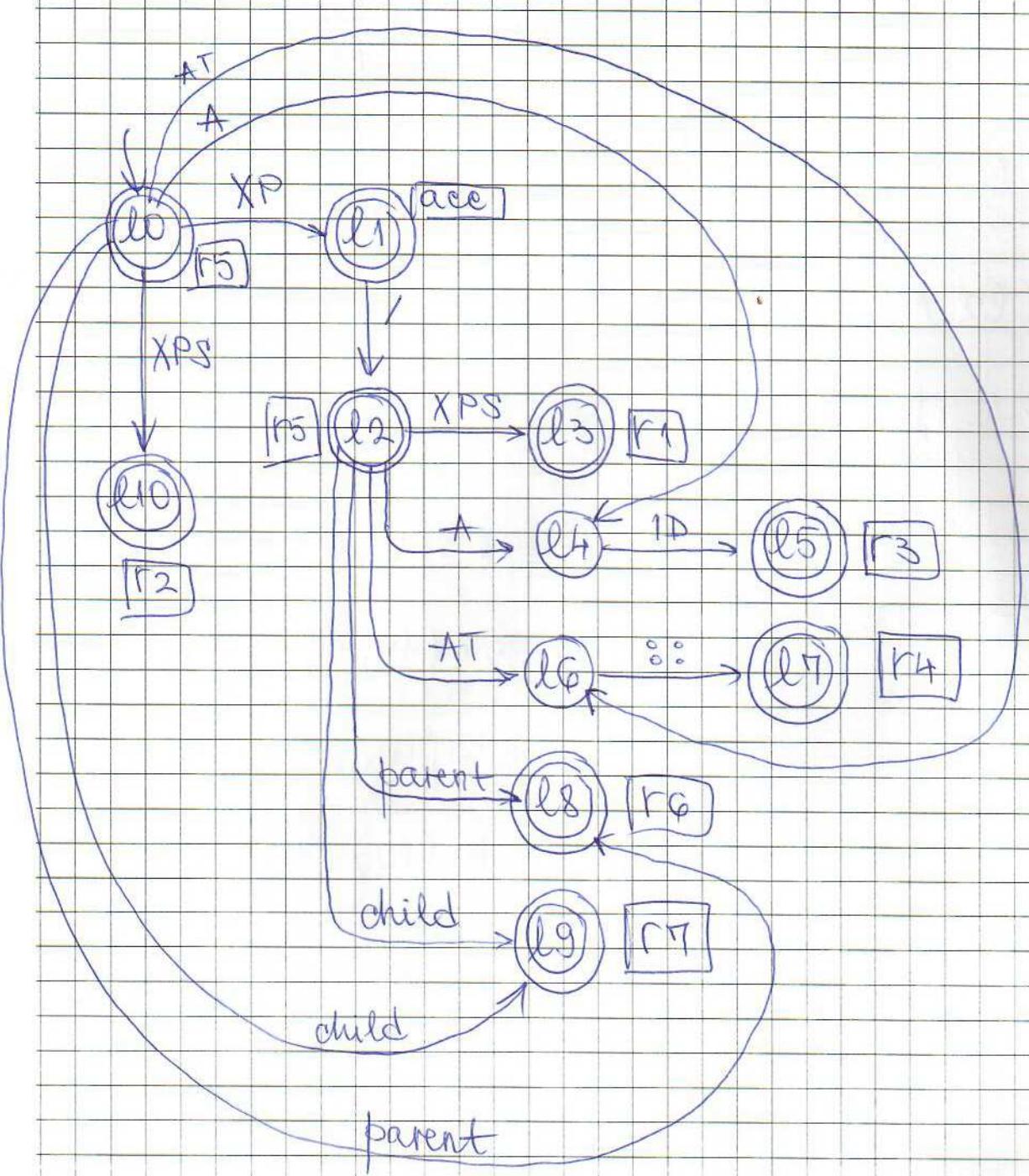
AT → child.

[r7]

(l10) = goto (lo, XPS)

Xp → XPS.

[r2]



$\text{FOLLOW}(\text{XP}) = \{\#, /\}$

$\text{FOLLOW}(\text{XPS}) = \{ /, \#\}$

$\text{FOLLOW}(A) = \{ \text{ID} \}$

$\text{FOLLOW}(\text{AT}) = \{ :: \}$

l	parent	child	ID	/	:::	/	/	Xp	Xps	A	AT
0	S8	sg	r5					1	10	4	6
1				S2		acc					
2	S8	sg	r5						3	4	6
3				r1		r1					
4			S5								
5				r3		r3					
6					S7						
7			r4								
8				r6							
9				r7							
10			r2		r2						

* Neopoznata polja su greske

parent :: ID / ID

stek	mala2	akcija
0	parent	S8
0parent8	::	r6
0AT6	::	S7
0AT6::7	ID	r4
0A4	ID	S5
0A4ID5	/	r3
0XPS-10-	/	r2
0XP1	/	S2
0XP1/2	ID	r5
0XP1/QA4	ID	S5

stek	rezultat	akcija
0XP1/2A4ID5	#	r3
0XP1/2XRS3	#	r1
0XP1	#	<u>ace</u> ↑

I primenom LR algoritma
dolezimo da je XPath
sintaksno ispravan