

# LR Parsing Table - An Example XML Document

Daniela da Cruz      Pedro Rangel Henriques

{danieladacruz,prh}@di.uminho.pt  
Departamento de Informática, CCTC  
Universidade do Minho

In this example we intend to develop a simple processor for annotated documents.  
The following context free grammar is a possible solution to express formally the structural (syntactic) requirements that should be observed by a "well-formed" XML document.

```
p1: DocXML --> MarcaAbr Conteudo MarcaFec
p2: MarcaAbr --> "<" EleXML ">"
p3: MarcaFec --> "<" "/" id ">"
p4: EleXML --> id Atribs
p5: Atribs --> &
p6: Atribs --> Atribs id "=" str
p7: Conteudo --> &
p8: Conteudo --> Conteudo Componente
p9: Componente--> pcddata
p10: Componente--> DocXML
```

Note: Initial Symbol is DocXML; Terminal symbols are written in lowercase (pseudo-terminals), or uppercase (reserved-words), or in inverted commas (like ', ' or '(' or ...); null string is denoted by & and character \$ represents EOF (end of input file).

## Resolution

To obtain the Bottom-up decision tables, ACTION and GOTO, we must build a finite deterministic automata (FDA) LR(0).

Beginning in state 0 with item

$$[S' \rightarrow \bullet \text{DocXML } '\$']$$

and following a systematic schema to build this automata, we obtain a FDA with 20 states, like the one shown in figure 1.

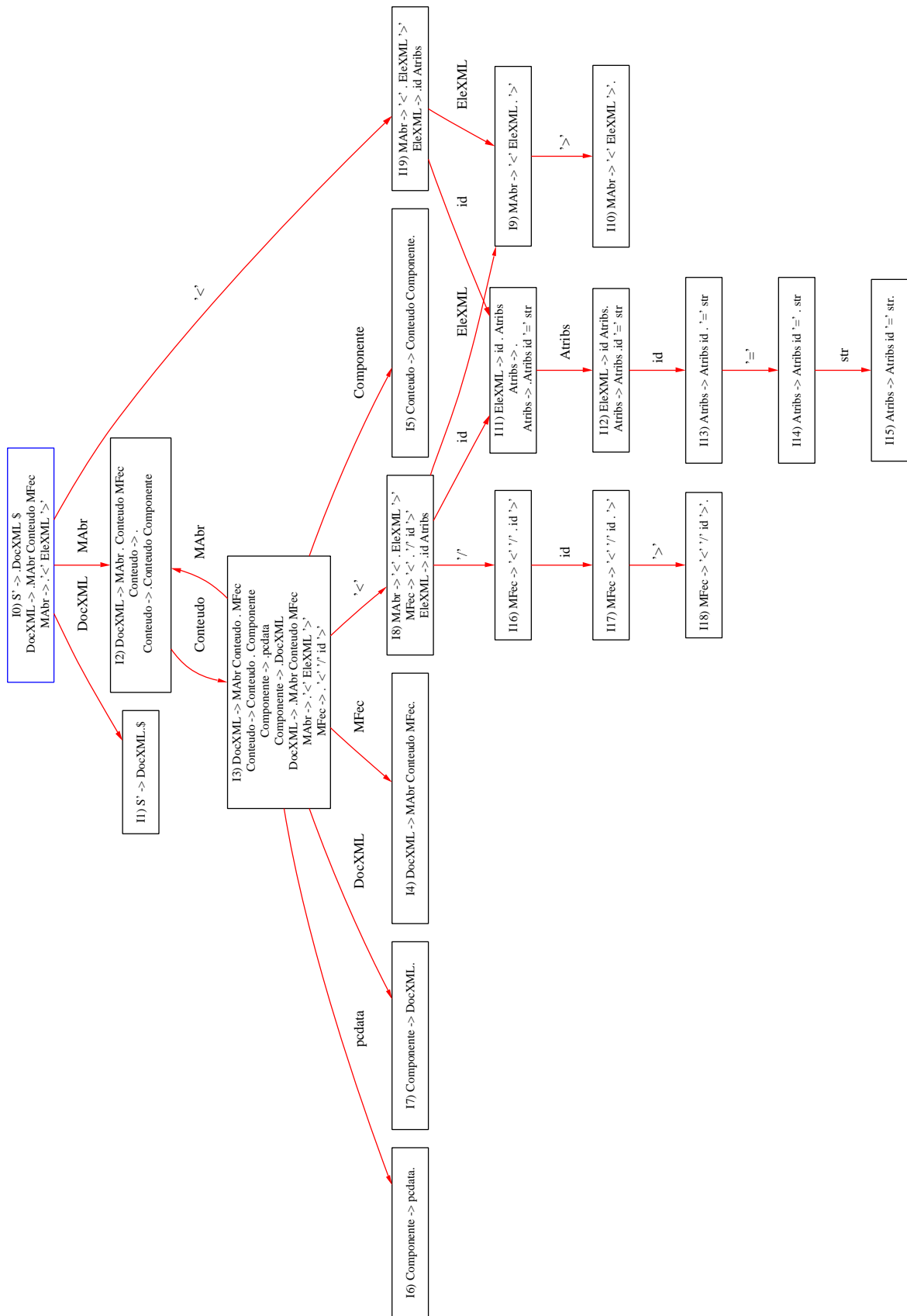


Figure 1: FDA from "XML" grammar.

From the transition function,  $\delta$ , associated with the automata LR(0) above, it is possible to derive bottom-up decision tables.

The tables ACTION and GOTO are shown in figures 2 and 3.

	'<'	'>'	'='	'/'	pcdata	id	str	\$
0	s19							
1								OK
2	#7	#7	#7	#7	#7	#7	#7	#7
3	s8				s6			
4	#1	#1	#1	#1	#1	#1	#1	#1
5	#8	#8	#8	#8	#8	#8	#8	#8
6	#9	#9	#9	#9	#9	#9	#9	#9
7	#10	#10	#10	#10	#10	#10	#10	#10
8				s16		s11		
9		s10						
10	#2	#2	#2	#2	#2	#2	#2	#2
11	#5	#5	#5	#5	#5	#5	#5	#5
12	#4	#4	#4	#4	#4	<b>#4/s13</b>	#4	#4
13			s14					
14							s15	
15	#6	#6	#6	#6	#6	#6	#6	#6
16						s17		
17		s18						
18	#3	#3	#3	#3	#3	#3	#3	#3
19						s11		

Figure 2: ACTION table LR(0)

	DocXML	MAbre	MFec	EleXML	Atribs	Conteudo	Componente
0	1	2					
1							
2						3	
3	7	2	4				5
4							
5							
6							
7							
8				9			
9							
10							
11					12		
12							
13							
14							
15							
16							
17							
18							
19				9			

Figure 3: GOTO table

However, looking to figures 1 and 2, we can conclude that the grammar **is not LR(0)** — there is a conflict *shift/reduce* at state 12 (reduce by production  $p_4$  and shift by terminal symbol  $id$ ). In this example it is important to remark that states 2 and 11 of automata LR(0) are tricky; booth contain a reduce item (by  $p_7$  on state 2, or by  $p_5$  on state 11), however there is no conflicts *shift/reduce* at these states because there is no transitions by terminal symbols (just one transition by a non-terminal symbol in each case, what does not imply a conflict). Reducing just on terminal symbols that belong to the *follow* of the left-hand side symbol of the production we want to reduce —SLR(1) strategy— we verify that:

- in state 12 we just reduce by terminal symbol ' $>$ ' ( $Follow(EleXML)$ ).

That result removes the previous *shift/reduce* conflict associated with terminal  $id$ . So, we can say that the grammar above is SLR(1).

Table 4 is the ACTION table for SLR(1) strategy (the lookaheads, necessary to build this table, are shown after table 4):

	'<'	'>'	'='	'/'	pcdata	id	str	\$
0	s19							
1								OK
2	#7				#7			
3	s8				s6			
4	#1							
5	#8				#8			
6					#9	#9	#9	#9
7	#10							
8				s16		s11		
9		s10						
10	#2							
11		#5				#5		
12						#4		
13			s14					
14							s15	
15						#6		
16						s17		
17		s18						
18	#3							
19						s11		

Figure 4: ACTION table SLR(1)

## Lookaheads

- p1

$$\begin{aligned}
 \text{lookahead}(\text{DocXML} \rightarrow \text{MAbr Conteudo MFec}) &= \text{First}(\text{MAbr Conteudo MFec}) \\
 &= \text{First}(\text{MAbr}) \\
 &= \text{First}('<' EleXML' >') \\
 &= \text{First}('<') \\
 &= \{ '<' \}
 \end{aligned}$$

- p2

$$\begin{aligned}
 \text{lookahead}(\text{MAbr} \rightarrow '<' EleXML' >') &= \text{First}('<' EleXML' >') \\
 &= \text{First}('<') \\
 &= \{ '<' \}
 \end{aligned}$$

- p3

$$\text{lookahead}(\text{MFec} \rightarrow '<' '/'id' >') = \text{First}('<')$$

{'<'}

- p4

$$\begin{aligned} \text{lookahead}(\text{EleXML} \rightarrow \text{idAtribs}) &= \text{First}(\text{idAtribs}) \\ &= \text{First}(\text{id}) \\ &= \{\text{id}\} \end{aligned}$$

- p5

$$\begin{aligned} \text{lookahead}(\text{Atribs} \rightarrow \epsilon) &= \text{First}(\epsilon) \cup \text{Follow}(\text{Atribs}) \\ &= (\text{First}(\epsilon) \cup \text{Follow}(\text{EleXML})) \cup \text{First}(\text{id}' = \text{str}) \\ &= \text{First}('>') \cup \text{First}(\text{id}) \\ &= \{>', \text{id}\} \end{aligned}$$

- p6

$$\begin{aligned} \text{lookahead}(\text{Atribs} \rightarrow \text{Atribsid}' = \text{str}) &= \text{First}(\text{Atribsid}' = \text{str}) \\ &= (\text{First}(\epsilon) \cup \text{First}(\text{Atribsid}' = \text{str})) \cup \text{First}(\text{id}' = \text{str}) \\ &= (\text{First}(\text{Atribs}) \cup \text{First}(\text{id})) \cup \text{First}(\text{id}) \\ &= \{\text{id}\} \end{aligned}$$

- p7

$$\begin{aligned} \text{lookahead}(\text{Conteudo} \rightarrow \epsilon) &= \text{First}(\epsilon) \cup \text{Follow}(\text{Conteudo}) \\ &= \text{First}(\text{MFec}) \cup \text{First}(\text{Componente}) \\ &= \text{First}('<' '/' \text{id}' >') \cup (\text{First}(\text{DocXML}) \cup \text{First}(\text{pcdata})) \\ &= \text{First}('<') \cup \{\text{pcdata}\} \cup \{<'\} \\ &= \{<', \text{pcdata}\} \end{aligned}$$

- p8

$$\begin{aligned}
\text{lookahead}(\text{Conteudo} \rightarrow \dots) &= \text{First}(\text{Conteudo Componente}) \\
&= (\text{First}(\epsilon) \cup \text{First}(\text{Conteudo Componente})) \cup \text{First}(\text{Componente}) \\
&= (\text{First}(\text{Conteudo}) \cup \text{First}(\text{Componente})) \\
&\quad \cup (\text{First}(\text{DocXML}) \cup \text{First}(\text{pdata})) \\
&= \{ '<', \text{pdata} \}
\end{aligned}$$

- p9

$$\begin{aligned}
\text{lookahead}(\text{Componente} \rightarrow \text{pdata}) &= \text{First}(\text{pdata}) \\
&= \{ \text{pdata} \}
\end{aligned}$$

- p10

$$\begin{aligned}
\text{lookahead}(\text{Componente} \rightarrow \text{DocXML}) &= \text{First}(\text{DocXML}) \\
&= \text{First}(\text{MAbr Conteudo MFec}) \\
&= \text{First}(\text{MAbr}) \\
&= \text{First}('<' EleXML' >') \\
&= \text{First}('<') \\
&= \{ '<' \}
\end{aligned}$$