

SLR(1) parsing / simple LR (1) parsing

Algoritma SLR(1) Parsing :

1. Jika state S berisi $A \rightarrow a.X\beta$, dimana X adalah terminal dan X adalah token berikutnya dalam string masukkan, maka shift ke stack, dan state baru di ke stack $A \rightarrow aX.\beta$
2. Jika state S adalah $A \rightarrow \gamma$. Dan token berikutnya adalah follow (A), maka berikutnya reduksi $A \rightarrow \gamma$. A oleh $S' \rightarrow S$, dimana S adalah state awal, maka aturan diterima apabila input berikutnya adalah $\$$. Dalam beberapa kasus, state baru dihitung sebagai berikut : Hapus γ dan semua state yang berhubungan dari stack parsing. Kemudian, buat state yang dimulai dari γ dimana $B \rightarrow a.A.\beta$, push A ke dalam stack dan $B \rightarrow aA.\beta$

Lanjutan Algoritma SLR (1)

3. Jika input berikutnya bukan kedua kasus diatas, maka beri penjelasan error. Algoritma SLR(1) di gunakan apabila :

1. Untuk semua $A \rightarrow a.X\beta$ dalam S dengan X adalah terminal, tidak ada $B \rightarrow \gamma$ pada S dengan X adalah follow (B)
2. Untuk $A \rightarrow a$ dan $B \rightarrow \beta$ pada S , maka follow (A) \cap follow (B) adalah empty

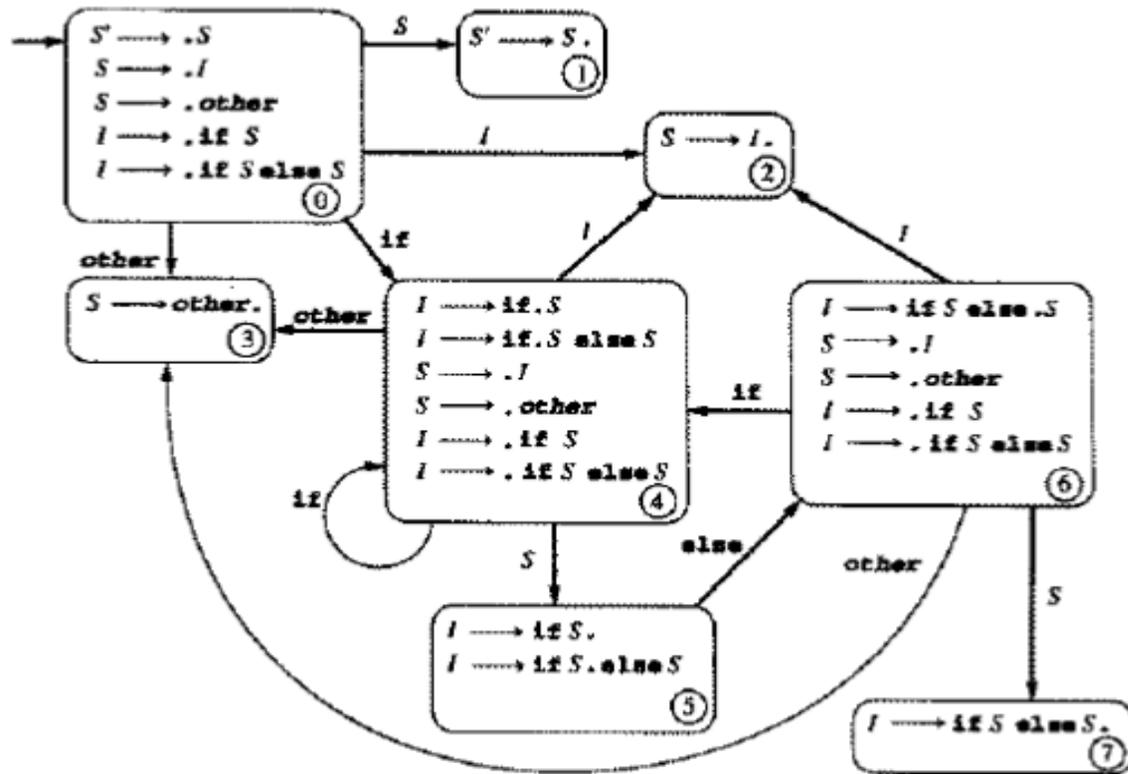
Contoh:

$S \rightarrow I \mid \text{other}$

$I \rightarrow \text{if } S \mid \text{if } S \text{ else } S$

Terlihat bahwa:

$\text{FOLLOW}(S) = \text{FOLLOW}(I) = \{\$, \text{else}\}$



➤ Membuat nomor untuk skema tabel dimana :

- (1) $S \rightarrow I$
- (2) $S \rightarrow \text{other}$
- (3) $I \rightarrow \text{if } S$
- (4) $I \rightarrow \text{if } S \text{ else } S$

Sehingga tabel SLR (1) Parsing :

State	Input				Goto	
	if	else	other	\$	S	I
0	s4		s3		1	2
1				accept		
2		r1		r1		
3		r2		r2		
4	s4		s3		5	2
5		s6		r3		
6	s4		s3		7	2
7		r4		r4		