

23 de Julho de 2021, Niterói
Instituto de Computação - UFF
Aluno: Thiago do Prado Silva
Matrícula: 117031024
Disciplina: Compiladores

SLR vs LALR

Analisando o caso de conflito shift/reduce:

No estado 2 da tabela de parsing gerada por SLR existe conflito, porque com o token '=' existem duas transições possíveis: shift (*regra 1 indo para o estado 6*) e reduce (*regra 5*). Esse problema não acontece com parsing LALR pois os itens nesse tipo de parsing são compostos pelo par produção e terminal ou \$.

Com isso, ao utilizar parsing LALR, no estado 2 o conflito já não existe tendo apenas uma transição para o token '=', o shift para o estado 6.

Como a ferramenta PLY reporta o conflito shift/reduce:

O arquivo parser.out gerado ao utilizar Parsing SLR informa no fim que existe um conflito shift/reduce no estado 2, resolvido como shift. Na descrição do segundo estado o PLY descreve as duas transições (linha 55 e linha 58 logo abaixo) sendo a primeira a transição realizada com shift pela regra 1 e a segunda transição com reduce pela regra 5.

```
123
124 WARNING:
125 WARNING: Conflicts:
126 WARNING:
127 WARNING: shift/reduce conflict for = in state 2 resolved as shift
128
```

```
48
49 state 2
50
51 ... (1) S → L . = R
52 ... (5) R → L .
53
54 ... ! shift/reduce conflict for = resolved as shift
55 ... = ..... shift and go to state 6
56 ... $end ..... reduce using rule 5 (R → L .)
57
58 ... ! = ..... [ reduce using rule 5 (R → L .) ]
59
```

Pode ser verificado abaixo como a ferramenta reporta a transição **sem** o conflito shift/reduce por utilizar Parsing LALR.

```
48
49 state 2
50
51 .....(1) S → L . = R
52 .....(5) R → L .
53
54 .....= .....shift and go to state 6
55 .....$end .....reduce using rule 5 (R → L .)
56
```