

Script generated by TTT

Title: Petter: Compiler Construction (11.06.2020)
-30: LR(1) Parser Example

Date: Wed Jun 10 15:16:10 CEST 2020

Duration: 08:24 min

Pages: 5

Chapter 2:
LR(k)-Parser Design

LR(k)-Parser Design

```

S' → E
E → E + T
   | T
T → T * F
   | F
F → ( E )
   | int
    
```

Parser Actions

For each rule, specify user code to be executed in case of reduction actions.

LR(k)-Parser Design

```

S' ::= E           {}
   ;
E ::= E plus T    {}
   | T             {}
   ;
T ::= T times F   {}
   | F             {}
   ;
F ::= lbrac E rbrac {}
   | intconst     {}
   ;
    
```

Parser Actions

For each rule, specify user code to be executed in case of reduction actions.

- add code sections delimited with `{: ;}` to each variant

LR(k)-Parser Design

```
S' ::= E:e      { : RESULT = e; : }
    |
E ::= E:e plus T:t { : RESULT = e + t; : }
   | T:t          { : RESULT = t; : }
   |
T ::= T:t times F:f { : RESULT = t * f; : }
   | F:f           { : RESULT = f; : }
   |
F ::= lbrac E:e rbrac { : RESULT = e; : }
   | intconst:c      { : RESULT = c; : }
   |
```

Parser Actions

For each rule, specify user code to be executed in case of reduction actions.

- 1 add code sections delimited with { : : } to each variant
- 2 produce results by assigning values to **RESULT**
- 3 add labels to symbols to refer to former results

46/57

LR(k)-Parser Design

```
S' ::= E:e      { : RESULT = e; : }
    |
E ::= E:e plus T:t { : RESULT = e + t; : }
   | T:t          { : RESULT = t; : }
   |
T ::= T:t times F:f { : RESULT = t * f; : }
   | F:f           { : RESULT = f; : }
   |
F ::= lbrac E:e rbrac { : RESULT = e; : }
   | intconst:c      { : RESULT = c; : }
   |
```

Parser Actions

For each rule, specify user code to be executed in case of reduction actions.

- 1 add code sections delimited with { : : } to each variant
- 2 produce results by assigning values to **RESULT**
- 3 add labels to symbols to refer to former results

Implementation Idea: add data stack that

- pushes **RESULT** after each user action
- translates labeled symbols to offset from top of stack based on the position in the rhs

46/57