Contents

1 Shift reduce parcing

1.1 Concept

Construct a DFA where each state is labelled by all possibilities given the input and reductions thus far. (Similar to how an NFA is turned into a DFA.). Whenever reduction is possible, if there **is only one** possible reduction, then it is always clear what to do.

1.2 LL, LR and LALR parsing

LL(k)

- Input scanned **L**eft to right
- **L**eft most derivation
- **k** symbols of lookahead

LR(k)

- Input scanned **L**eft to right
- **R**ightmost derivation in reverse
- **k** symbols of lookahead

LALR(k)

- **L**ook**A**head **LR** (simplified LR parsing)

1.3 Why

These methods handle a wide class of grammars of practical significance. In particular, handles left- and right-recursive grammars (but left rec. needs less stack). LALR is a good compromise between expressiveness and space cost of implementation. Consequently, many parser generator tools based on LALR.

Reference section

placeholder