

# ToC

- 1 Syntax Analysis: the problem
- 2 Theoretical Background
- 3 Syntax Analysis: solutions**
  - Top-Down parsing
  - Bottom-Up Parsing**

# Bottom-up Parsing

## Bottom-up Parsing

The problem of Bottom-up parsing can be viewed as the problem of constructing a parse tree for an input string beginning at the leaves and working up towards the root.

In particular we will consider the problem of finding the rightmost derivation given an input string, through a series of reductions to reach the initial symbol

Let's consider the input string `id * id` using the simple grammar for expressions

# Bottom-up Parsing

## Bottom-up Parsing

The problem of Bottom-up parsing can be viewed as the problem of constructing a parse tree for an input string beginning at the leaves and working up towards the root.

In particular we will consider the problem of finding the rightmost derivation given an input string, through a series of reductions to reach the initial symbol

Let's consider the input string **id \* id** using the simple grammar for expressions

# Tools for Bottom-up Parsing

## Reductions

In a bottom-up parser at each step a **reduction** is applied. A certain string is reduced to the head of the production (non-terminal) applying the production in reverse. The **key decision** is when to reduce!

## Handle Pruning

A **handle** is a substring of a sentential form that matches the body of a production and whose reduction represents a step along the rightmost derivation of the sentential form in reverse.

Consider the grammar  $S \rightarrow 0S1|01$  and the two sentential forms 000111, 00S11

# Tools for Bottom-up Parsing

## Reductions

In a bottom-up parser at each step a **reduction** is applied. A certain string is reduced to the head of the production (non-terminal) applying the production in reverse. The **key decision** is when to reduce!

## Handle Pruning

A **handle** is a substring of a sentential form that matches the body of a production and whose reduction represents a step along the rightmost derivation of the sentential form in reverse.

Consider the grammar  $S \rightarrow 0S1|01$  and the two sentential forms 000111, 00S11

# Shift-reduce parsing

## Shift-reduce parsing

A **shift-reduce** parser is a particular kind of bottom-up parser in which a stack holds grammar symbols and an input buffer holds the rest of the string to be parsed. Four possible actions are possible:

- ▶ shift
- ▶ reduce
- ▶ accept
- ▶ error

## Conflicts

- ▶ shift/reduce
- ▶ reduce/reduce

# Shift-reduce parsing

## Shift-reduce parsing

A **shift-reduce** parser is a particular kind of bottom-up parser in which a stack holds grammar symbols and an input buffer holds the rest of the string to be parsed. Four possible actions are possible:

- ▶ shift
- ▶ reduce
- ▶ accept
- ▶ error

## Conflicts

- ▶ shift/reduce
- ▶ reduce/reduce

# Shift-reduce parsing

Consider the grammar  $S \rightarrow SS + \mid SS * \mid a$  and the following sentential forms:  
 $SSS + a * +$ ,  $SS + a * a +$ ,  $aaa * a + +$