

COS 301: Programming Languages

# Syntax Analysis

# Parsing

- Syntactic analysis = *parsing*
- Goal of parser:
  - Find all syntax errors  $\Rightarrow$  diagnostic message
  - If no syntax errors  $\Rightarrow$  parse tree

# Types of parsers

# Types of parsers

- *Top-down parsers:*

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB_y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xABy$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:



# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xABy$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xABy \Rightarrow xwBy \Rightarrow xwzy$$

- *Bottom-up parsers:*

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xABy$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xABy \Rightarrow xwBy \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree
  - Reverse order of a *rightmost* derivation

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree
  - Reverse order of a *rightmost* derivation
  - Ex:  $xwzy$ , same grammar, derivation:

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree
  - Reverse order of a *rightmost* derivation
  - Ex:  $xwzy$ , same grammar, derivation:

$$xwzy \Rightarrow xAzy \Rightarrow xAB y \Rightarrow S$$

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree
  - Reverse order of a *rightmost* derivation
  - Ex:  $xwzy$ , same grammar, derivation:

$$xwzy \Rightarrow xAzy \Rightarrow xAB y \Rightarrow S$$

# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree
  - Reverse order of a *rightmost* derivation
  - Ex:  $xwzy$ , same grammar, derivation:

$$xwzy \Rightarrow xAzy \Rightarrow xAB y \Rightarrow S$$



# Types of parsers

- *Top-down parsers:*
  - Start at root to produce parse tree
  - Order: *leftmost* derivation (preorder traversal)
  - Ex:  $xwzy$ , grammar:  $S \rightarrow xAB y$ ,  $A \rightarrow w$ ,  $B \rightarrow z$

derivation:

$$S \Rightarrow xAB y \Rightarrow xwB y \Rightarrow xwzy$$

- *Bottom-up parsers:*
  - Start at leaves to produce tree
  - Reverse order of a *rightmost* derivation
  - Ex:  $xwzy$ , same grammar, derivation:

$$xwzy \Rightarrow xAzy \Rightarrow xAB y \Rightarrow S$$

# Computational complexity

- Brute-force approach:
  - Try every possible rule (exhaustive search)
  - *Exponential* in length of the program
  - So don't do that!
- Better:
  - Several algorithms  $\Rightarrow \mathcal{O}(n^3)$
  - Still too expensive for commercial compilers
- Can reduce the generality of the language to be parsed  $\Rightarrow$  linear  $\mathcal{O}(n)$

# Top-Down Parsing

# Top-down parsing

- Given  $xA\alpha$ , where
  - $x$  is a string of terminal symbols
  - $A$  is the leftmost nonterminal symbol
  - $\alpha$  is a string of terminals and non-terminals
- Goal: find next sentential form in leftmost derivation
- Thus: choose rule such that  $A$  is the LHS; e.g.,
  - rules:  $A \Rightarrow bB$ ,  $A \Rightarrow cBb$ ,  $A \Rightarrow a$
  - then we choose between:
    - $xA\alpha \Rightarrow xbB\alpha$ ,  $xA\alpha \Rightarrow xcBb\alpha$ , and  $xA\alpha \Rightarrow xa\alpha$

# Top-down parsing

- Which to choose for  $xA\alpha$ ?
  - choices:  $A \Rightarrow xbB\alpha$ ,  $A \Rightarrow xcBb\alpha$ , and  $A \Rightarrow xa\alpha$
  - Look at the input
  - If the next token after  $x$  is  $a$ ,  $b$ , or  $c$ , then it's clear
- What if some rule's RHS has nonterminals first?
  - E.g.,  $A \rightarrow Bb$ , choice includes  $xBb\alpha$
  - Which to choose?
- Much harder in this case – depends on what  $B$  expands to, etc.

# Most common top-down parsers

- Two most common: *recursive-descent* and *table-driven*
- Work on a *subset* of CF grammars
- Algorithms are called **LL**
  - First L = left-to-right scan of input
  - Second L = leftmost derivation

# Recursive-descent parser

- Simple algorithm to understand
- **Mutually-recursive set of procedures**
  - One for each (EBNF) production of language
  - I.e., one for each nonterminal
- Top-down parsing
- To begin, call procedure for start nonterminal
- Call procedure for leftmost nonterminal, etc.
- Procedure tries  $\Rightarrow$  parse tree rooted at itself, matches input
- Use scanner (lexer, tokenizer)  $\Rightarrow$  next token as needed

# RD parser: example



# RD parser: example

- Grammar:

Given:  $A+B*C$

# RD parser: example

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$

Given: **A+B\*C**

# RD parser: example

- Grammar:

`<expr> → <term> { (+ | -) <term> }`

`<term> → <factor> { (* | /) <factor> }`

Given: **A+B\*C**

# RD parser: example

- Grammar:

`<expr> → <term> { (+ | -) <term> }`

`<term> → <factor> { (* | /) <factor> }`

`<factor> → <id> | int_const | (<expr>)`

Given: **A+B\*C**

# RD parser: example

- Grammar:

`<expr> → <term> { (+ | -) <term> }`

`<term> → <factor> { (* | /) <factor> }`

`<factor> → <id> | int_const | (<expr>)`

Given: **A+B\*C**

→Call `<expr>`

# RD parser: example

- Grammar:

`<expr> → <term> { (+ | -) <term> }`

`<term> → <factor> { (* | /) <factor> }`

`<factor> → <id> | int_const | (<expr>)`

Given: **A+B\*C**

→Call <expr>

→Call <term>

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$   
    → Call  $\langle \text{term} \rangle$   
        → Call  $\langle \text{factor} \rangle$

# RD parser: example

Given: **A+B\*C**

- Grammar:

`<expr> → <term> { (+ | -) <term> }`

`<term> → <factor> { (* | /) <factor> }`

`<factor> → <id> | int_const | (<expr>)`

→ Call `<expr>`

→ Call `<term>`

→ Call `<factor>`

→ Call `<id>`: next token is an id (A), so success; next token not \* or /



# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

# RD parser: example

Given: **A+B\*C**

- Grammar:

**<expr>** → **<term>** { (+ | -) **<term>** }

**<term>** → **<factor>** { (\* | /) **<factor>** }

**<factor>** → **<id>** | **int\_const** | (**<expr>**)

→ Call **<expr>**

→ Call **<term>**

→ Call **<factor>**

→ Call **<id>**: next token is an id (A), so success; next token not \* or /

← Return **<id>** → A

← Return **<factor>** → **<id>** → A

# RD parser: example

Given: **A+B\*C**

- Grammar:

`<expr> → <term> { (+ | -) <term> }`

`<term> → <factor> { (* | /) <factor> }`

`<factor> → <id> | int_const | (<expr>)`

→ Call `<expr>`

→ Call `<term>`

→ Call `<factor>`

→ Call `<id>`: next token is an id (A), so success; next token not \* or /

← Return `<id>` → A

← Return `<factor>` → `<id>` → A

Next token is +, so...

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow B$



# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow B$

← Return  $\langle \text{id} \rangle \rightarrow B$

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow B$

← Return  $\langle \text{id} \rangle \rightarrow B$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow B$

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow B$

← Return  $\langle \text{id} \rangle \rightarrow B$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow B$

Next token is \*, so...

# RD parser: example

Given: **A+B\*C**

- Grammar:

**<expr> → <term> { (+ | -) <term> }**

**<term> → <factor> { (\* | /) <factor> }**

**<factor> → <id> | int\_const | (<expr>)**

→ Call <expr>

→ Call <term>

→ Call <factor>

→ Call <id>: next token is an id (A), so success; next token not \* or /

← Return <id> → A

← Return <factor> → <id> → A

Next token is +, so...

← Return <term> → <factor> → <id> → A

Next token is +, so...

→ Call <term>

→ Call <factor>

→ Call <id> ⇒ B

← Return <id> → B

← Return <factor> → <id> → B

Next token is \*, so...

→ Call <factor>

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow B$

← Return  $\langle \text{id} \rangle \rightarrow B$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow B$

Next token is \*, so...

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow C,$

# RD parser: example

Given: **A+B\*C**

- Grammar:  
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+ \mid -) \langle \text{term} \rangle \}$   
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (* \mid /) \langle \text{factor} \rangle \}$   
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \text{int\_const} \mid (\langle \text{expr} \rangle)$

→ Call  $\langle \text{expr} \rangle$

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle$ : next token is an id (A), so success; next token not \* or /

← Return  $\langle \text{id} \rangle \rightarrow A$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

← Return  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow A$

Next token is +, so...

→ Call  $\langle \text{term} \rangle$

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow B$

← Return  $\langle \text{id} \rangle \rightarrow B$

← Return  $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \rightarrow B$

Next token is \*, so...

→ Call  $\langle \text{factor} \rangle$

→ Call  $\langle \text{id} \rangle \Rightarrow C,$

← Return  $\langle \text{id} \rangle \rightarrow C$

# RD parser: example

Given: **A+B\*C**

- Grammar:

**<expr> → <term> { (+ | -) <term> }**

**<term> → <factor> { (\* | /) <factor> }**

**<factor> → <id> | int\_const | (<expr>)**

→ Call <expr>

→ Call <term>

→ Call <factor>

→ Call <id>: next token is an id (A), so success; next token not \* or /

← Return <id> → A

← Return <factor> → <id> → A

Next token is +, so...

← Return <term> → <factor> → <id> → A

Next token is +, so...

→ Call <term>

→ Call <factor>

→ Call <id> ⇒ B

← Return <id> → B

← Return <factor> → <id> → B

Next token is \*, so...

→ Call <factor>

→ Call <id> ⇒ C,

← Return <id> → C

← Return <factor> → <id> → C

# RD parser: example

Given: **A+B\*C**

- Grammar:

**<expr>** → **<term>** { (+ | -) **<term>** }

**<term>** → **<factor>** { (\* | /) **<factor>** }

**<factor>** → **<id>** | **int\_const** | (**<expr>**)

→ Call **<expr>**

→ Call **<term>**

→ Call **<factor>**

→ Call **<id>**: next token is an id (A), so success; next token not \* or /

← Return **<id>** → A

← Return **<factor>** → **<id>** → A

Next token is +, so...

← Return **<term>** → **<factor>** → **<id>** → A

Next token is +, so...

→ Call **<term>**

→ Call **<factor>**

→ Call **<id>** ⇒ B

← Return **<id>** → B

← Return **<factor>** → **<id>** → B

Next token is \*, so...

→ Call **<factor>**

→ Call **<id>** ⇒ C,

← Return **<id>** → C

← Return **<factor>** → **<id>** → C

← Return **<term>** → (**<factor>** → **<id>** → B) \* (**<factor>** → **<id>** → C)



# RD parser: example

Given: **A+B\*C**

- Grammar:

**<expr> → <term> { (+ | -) <term> }**

**<term> → <factor> { (\* | /) <factor> }**

**<factor> → <id> | int\_const | (<expr>)**

→ Call <expr>

→ Call <term>

→ Call <factor>

→ Call <id>: next token is an id (A), so success; next token not \* or /

← Return <id> → A

← Return <factor> → <id> → A

Next token is +, so...

← Return <term> → <factor> → <id> → A

Next token is +, so...

→ Call <term>

→ Call <factor>

→ Call <id> ⇒ B

← Return <id> → B

← Return <factor> → <id> → B

Next token is \*, so...

→ Call <factor>

→ Call <id> ⇒ C,

← Return <id> → C

← Return <factor> → <id> → C

← Return <term> → (<factor> → <id> → B) \* (<factor> → <id> → C)

← Return <expr> → (<term> → <factor> → <id> → A) + (<factor> → <id> → B) \* <factor> → <id> → C)

# RD parser: <expr>

```
void expr() {
    /* parses <expr> => <term> { (+|-) <term> } */
    printf("enter <expr>\n");
    term();
    while (nextToken == ADD_OP ||
           nextToken == SUB_OP) {
        lex();
        term();
    }
    printf("exit <expr>\n");
}
```

/\* Q: Where does nextToken come from?

A: each function leaves the next unconsumed token in nextToken  
each function assumes on entry that it is available in  
nextToken \*/

# RD parser: <term>

```
void term() {
    /* parses <term> => <factor> { (*|/) <factor> } */

    printf("enter <term>\n");
    factor();
    while (nextToken == MULT_OP ||
           nextToken == DIV_OP) {
        lex();
        factor();
    }
    printf("exit <term>\n");
}
```

# RD parser: $\langle \text{factor} \rangle$

- Has to choose among several alternatives:  
 $\langle \text{factor} \rangle \rightarrow \langle \text{id} \rangle \mid \langle \text{int\_const} \rangle \mid (\langle \text{expr} \rangle)$
- May be able to detect a syntax error here:
  - E.g.,  $A + B * / C$
  - Fail in  $\langle \text{factor} \rangle \Rightarrow$  “/” isn’t any allowed symbol

# RD parser: <factor>

```
void factor() {
/* parses <factor> => <id> | int_constant | ( expr ) */

printf("enter <factor>\n");
if (nextToken == IDENT || nextToken == INT_LIT)
    lex();
else {
    if (nextToken == LEFT_PAREN) {
        lex();
        expr(); /* recursion! */
        if (nextToken == RIGHT_PAREN)
            lex();
        else
            error();
    } else
        error();
}
printf("exit <factor>\n");
}
```

# RD parser: Example

- Given (sum + 47)/total

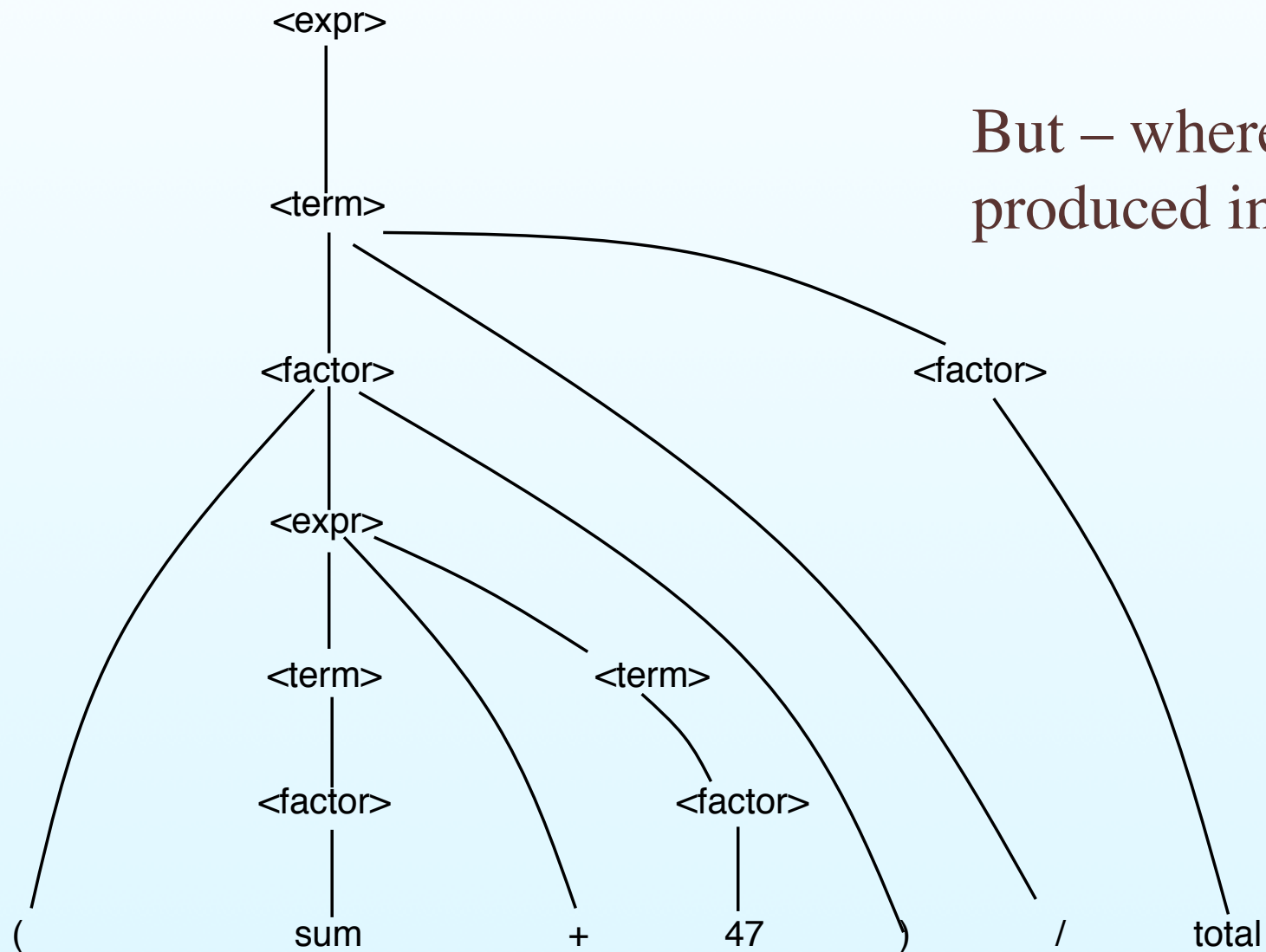
```
Next token is: 25 lexeme is (  
Enter <expr>  
Enter <term>  
Enter <factor>  
Next token is: 11 lexeme is sum  
Enter <expr>  
Enter <term>  
Enter <factor>  
Next token is: 21 lexeme is +  
Exit <factor>  
Exit <term>  
Next token is: 10 lexeme is 47  
Enter <term>  
Enter <factor>
```

# RD parser: Example

```
Next token is: 26 lexeme is )
Exit <factor>
Exit <term>
Exit <expr>
Next token is: 24 lexeme is /
Exit <factor>
Next token is: 11 lexeme is total
Enter <factor>
Next token is: -1 lexeme is EOF
Exit <factor>
Exit <term>
Exit <expr>
```

# Parse tree

- For (sum + 47)/total



But – where is this produced in the functions?



# Another example: if statement

```
<ifStmt> → if ( <boolExpr> ) <stmt> [else <stmt> ]
```

- Program for `<ifStmt>` nonterminal:
  - Check if current token is `if`
  - Call `lex()`, check if current token is `(`
  - Call `lex()`, then call `<boolExpr>`
  - Check if current token is `)`
  - Call `lex()` and call `<stmt>`
  - Check if current token = `else`
    - if so, call `lex()`, then call `<stmt>`

# <ifStmt>

```
void ifstmt(){
    if (nextToken != IF_CODE)
        error();
    else {
        lex();
        if (nextToken != LEFT_PAREN)
            error();
        else {
            lex(); /* error in text; this was omitted */
            boolexpr();
            if (nextToken != RIGHT_PAREN)
                error();
            else {
                lex(); /* error in text; this was omitted */
                stmt();

                if (nextToken == ELSE_CODE){
                    lex();
                    stmt();
                } /* end if (nextToken == ELSE_CODE) */
            } /* end if (nextToken != RIGHT_PAREN) */
        } /* end if (nextToken != LEFT_PAREN) */
    } /* end if (nextToken != IF_CODE) */
} /* end IF_STMT */
```

# LL grammars

- Recursive-descent parsers: parse left-to-right, leftmost derivations (LL)

- **Classes of grammars limited**

- Can't parse grammars with rules of form:

$$A \rightarrow A + B$$

- Direct left-recursion  $\Rightarrow$  infinite recursion!

- Can also have indirect recursion:

$$A \rightarrow Ba$$

$$B \rightarrow Ab$$

- Also, need to be able to choose correct RHS based on next input token – not always possible

# Eliminating left-recursion

- For every nonterminal  $A$ :

- Group the rules for  $A$  as:

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

- Then replace these rules with:

$$A \rightarrow \beta_1A' \mid \beta_2A' \mid \dots \mid \beta_nA'$$

$$A' \rightarrow \alpha_1A' \mid \alpha_2A' \mid \dots \mid \alpha_mA' \mid \varepsilon$$

- Left-recursive grammar  $\Rightarrow$  right-recursive grammar

# Example

# Example

- Grammar:

$$\begin{aligned} \langle e \rangle &\rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle \\ \langle t \rangle &\rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle \\ \langle f \rangle &\rightarrow (\langle e \rangle) \mid \langle \text{id} \rangle \end{aligned}$$

# Example

- Grammar:

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$

$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$

$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle \text{id} \rangle$$

- For  $\langle e \rangle$ ,  $\alpha_1 = + \langle t \rangle$ ,  $\beta_1 = \langle t \rangle$ , so:

$$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$$

$$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \varepsilon$$

# Example

- Grammar:

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$

$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$

$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle \text{id} \rangle$$

- For  $\langle e \rangle$ ,  $\alpha_1 = + \langle t \rangle$ ,  $\beta_1 = \langle t \rangle$ , so:

$$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$$

$$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \varepsilon$$

- $\langle t \rangle$ :  $\alpha_1 = * \langle f \rangle$ ,  $\beta_1 = \langle f \rangle$ , so:

$$\langle t \rangle \rightarrow \langle f \rangle \langle t' \rangle$$

$$\langle t' \rangle \rightarrow * \langle f \rangle \langle t' \rangle \mid \varepsilon$$



# Example

- Grammar:

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$

$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$

$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle \text{id} \rangle$$

- For  $\langle e \rangle$ ,  $\alpha_1 = + \langle t \rangle$ ,  $\beta_1 = \langle t \rangle$ , so:

$$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$$

$$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \varepsilon$$

- $\langle t \rangle$ :  $\alpha_1 = * \langle f \rangle$ ,  $\beta_1 = \langle f \rangle$ , so:

$$\langle t \rangle \rightarrow \langle f \rangle \langle t' \rangle$$

$$\langle t' \rangle \rightarrow * \langle f \rangle \langle t' \rangle \mid \varepsilon$$

- **Note:** no left-associativity of operators enforced – would have to do this in code generation

# Example

- Grammar:

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$

$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$

$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle \text{id} \rangle$$

- For  $\langle e \rangle$ ,  $\alpha_1 = + \langle t \rangle$ ,  $\beta_1 = \langle t \rangle$ , so:

$$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$$

$$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \varepsilon$$

- $\langle t \rangle$ :  $\alpha_1 = * \langle f \rangle$ ,  $\beta_1 = \langle f \rangle$ , so:

$$\langle t \rangle \rightarrow \langle f \rangle \langle t' \rangle$$

$$\langle t' \rangle \rightarrow * \langle f \rangle \langle t' \rangle \mid \varepsilon$$

- **Note:** no left-associativity of operators enforced – would have to do this in code generation

# Example

- Grammar:

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$

$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$

$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle \text{id} \rangle$$

- For  $\langle e \rangle$ ,  $\alpha_1 = + \langle t \rangle$ ,  $\beta_1 = \langle t \rangle$ , so:

$$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$$

$$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \varepsilon$$

- $\langle t \rangle$ :  $\alpha_1 = * \langle f \rangle$ ,  $\beta_1 = \langle f \rangle$ , so:

$$\langle t \rangle \rightarrow \langle f \rangle \langle t' \rangle$$

$$\langle t' \rangle \rightarrow * \langle f \rangle \langle t' \rangle \mid \varepsilon$$

- **Note:** no left-associativity of operators enforced – would have to do this in code generation

# Example

## Grammar:

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

$\Rightarrow$

$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$

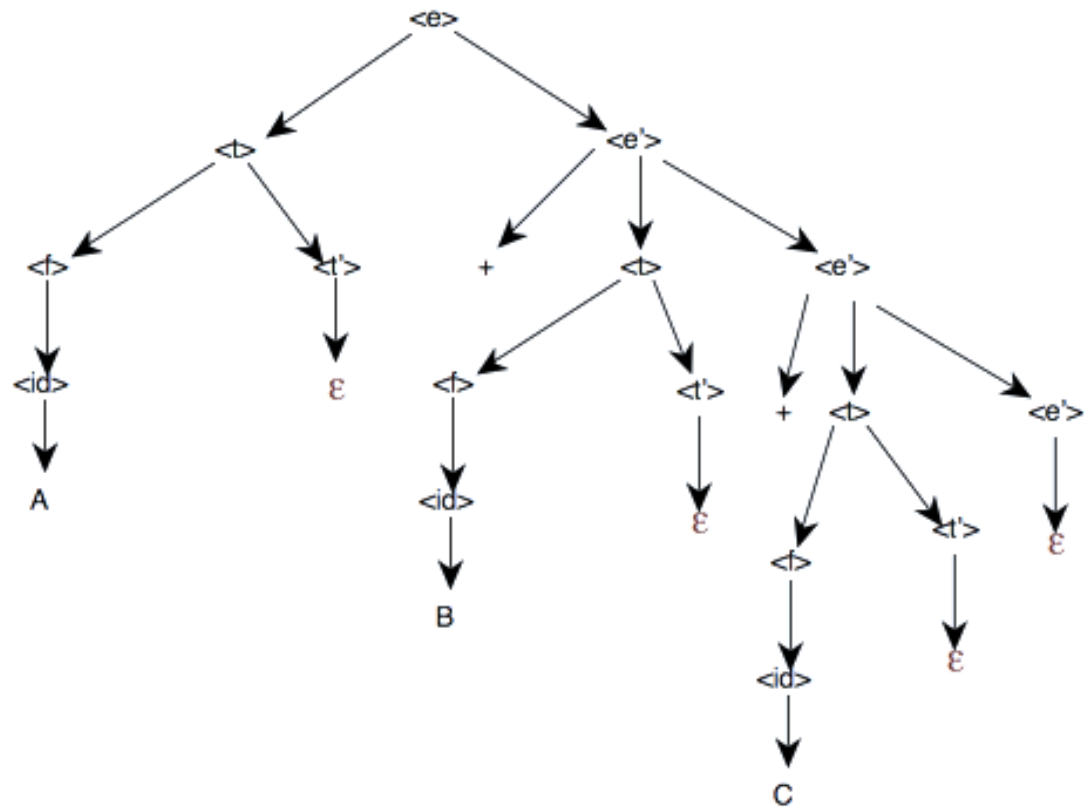
$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \epsilon$

$\langle t \rangle \rightarrow \langle f \rangle \langle t' \rangle$

$\langle t' \rangle \rightarrow * \langle f \rangle \langle t' \rangle \mid \epsilon$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Parse:  $A + B * C$



# Example

## Grammar:

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

$\Rightarrow$

$\langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$

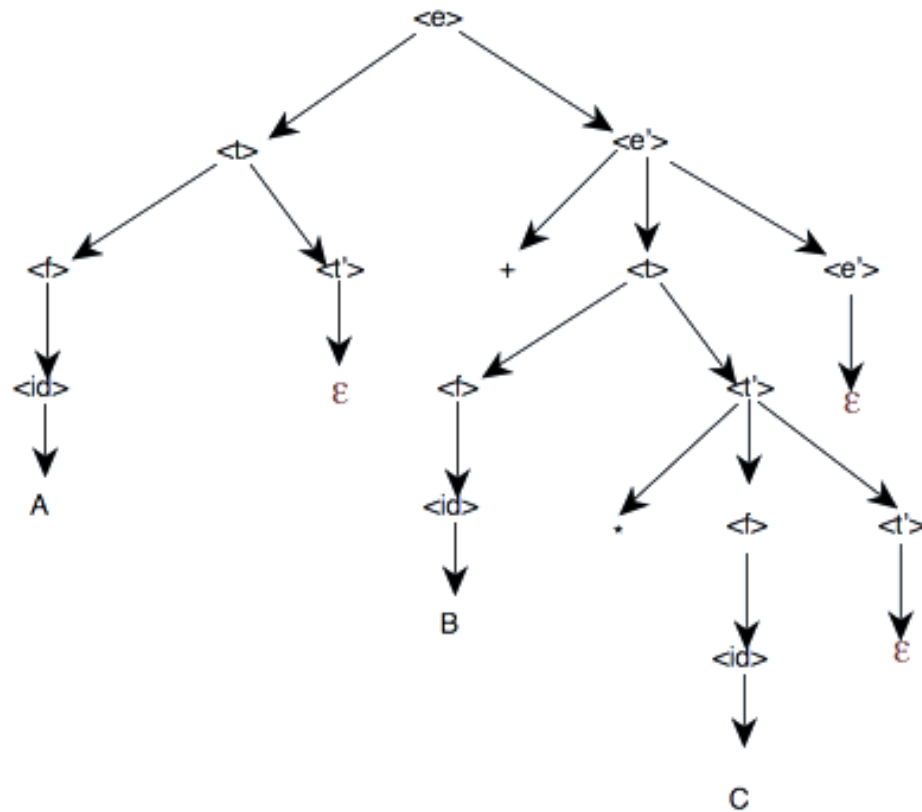
$\langle e' \rangle \rightarrow + \langle t \rangle \langle e' \rangle \mid \epsilon$

$\langle t \rangle \rightarrow \langle f \rangle \langle t' \rangle$

$\langle t' \rangle \rightarrow * \langle f \rangle \langle t' \rangle \mid \epsilon$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Parse:  $A + B * C$



# Indirect recursion

- Indirect recursion also a problem for recursive-descent parsers

- E.g.,

$$A \rightarrow B c$$
$$B \rightarrow A d$$

- Algorithm exists to get rid of it, too
- May be able to write grammar to avoid both kinds of recursion

# Choosing RHS

- Lookahead
- Best case:
  - single token, no backtracking
  - called a **LL(1)** grammar (and algorithm)
- **Pairwise disjointness test**: checks if grammar is LL(1)

# Pairwise disjointedness test

- General idea:
  - If non-terminal has  $> 1$  RHS and...
  - ...the different RHSs ultimately produce same terminal...
  - ...then:
    - grammar is not LL(1) and...
    - ...can't be parsed w/ recursive-descent



# Pairwise disjointedness test

- Compute set  $\text{First}(r)$ , where  $r$  is a rule RHS

$$\text{First}(r) = \{t \mid r \Longrightarrow^* t \beta\}$$

where  $t$  is a terminal,  $\beta$  is rest of sentential form

- If non-terminal  $N$  has two rules,  $i$  and  $j$ 
  - Compute  $\text{First}(i)$ ,  $\text{First}(j)$
  - If  $\text{First}(i) \cap \text{First}(j) \neq \{\}$   $\Rightarrow$  grammar is not LL(1)

# Left factoring

- If not pairwise disjoint, sometimes can fix grammar

# Left factoring

- If not pairwise disjoint, sometimes can fix grammar
- *Left factoring* sometimes works:
  - Creates a new nonterminal
  - Old rule(s) now combined into one, which uses new nonterminal

# Left factoring

- If not pairwise disjoint, sometimes can fix grammar
- *Left factoring* sometimes works:
  - Creates a new nonterminal
  - Old rule(s) now combined into one, which uses new nonterminal
- E.g.:  $A \rightarrow a \mid aB$  – not pairwise disjoint

$$A \rightarrow a A'$$

$$A' \rightarrow \varepsilon \mid B$$

# Left factoring

- If not pairwise disjoint, sometimes can fix grammar
- *Left factoring* sometimes works:
  - Creates a new nonterminal
  - Old rule(s) now combined into one, which uses new nonterminal
- E.g.:  $A \rightarrow a \mid aB$  – not pairwise disjoint

$$A \rightarrow a A'$$

$$A' \rightarrow \varepsilon \mid B$$

# Left factoring

- If not pairwise disjoint, sometimes can fix grammar
- *Left factoring* sometimes works:

- Creates a new nonterminal
- Old rule(s) now combined into one, which uses new nonterminal

- E.g.:  $A \rightarrow a \mid aB$  – not pairwise disjoint

$$A \rightarrow a A'$$

$$A' \rightarrow \varepsilon \mid B$$

- Or use EBNF:  $A \rightarrow a [ B ]$

# Left factoring

- If not pairwise disjoint, sometimes can fix grammar
- *Left factoring* sometimes works:
  - Creates a new nonterminal
  - Old rule(s) now combined into one, which uses new nonterminal
- E.g.:  $A \rightarrow a \mid aB$  – not pairwise disjoint

$$A \rightarrow a A'$$

$$A' \rightarrow \varepsilon \mid B$$

$$\text{Ex: } \langle e \rangle \rightarrow \langle t \rangle \mid \langle t \rangle + \langle e \rangle \mid \langle t \rangle - \langle e \rangle$$

$$\Rightarrow \langle e \rangle \rightarrow \langle t \rangle \langle e' \rangle$$

$$\langle e' \rangle \rightarrow + \langle e \rangle \mid - \langle e \rangle$$

- Or use EBNF:  $A \rightarrow a [ B ]$

# Bottom-Up Parsing



# Bottom-up parsing

- Called **LR algorithms**
  - L = left-to-right scan
  - R = rightmost derivation
- Creates a parse tree starting with the leaves (the input)
- Creates tree in reverse order of rightmost derivation
- Sentential forms for this are called *right sentential forms*

# Basic process

- Start with the input sentence (a right-sentential form)
- Find a RHS in the sentential form
- Replace it with the LHS
- Repeat until only the start symbol left.
- Record of the *reductions*  $\Rightarrow$  the parse tree

# Example

# Example

- Grammar:

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$

# Example

- Grammar:

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \end{aligned}$$

# Example

- Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow id \mid ( E )$$

# Example

- Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow id \mid ( E )$$

- Parse:  $A + B + C$



# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\text{id} + \text{id} + \text{id}$
- $\text{id} + \text{id} + \text{id}$
- $F + \text{id} + \text{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{F} + \mathbf{id} + \mathbf{id}$
- $\mathbf{T} + \mathbf{id} + \mathbf{id}$

# Example

- Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow \text{id} \mid ( E )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\text{id} + \text{id} + \text{id}$
- $\text{id} + \text{id} + \text{id}$
- $F + \text{id} + \text{id}$
- $T + \text{id} + \text{id}$
- $E + \text{id} + \text{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{F} + \mathbf{id} + \mathbf{id}$
- $\mathbf{T} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F} + \mathbf{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{F} + \mathbf{id} + \mathbf{id}$
- $\mathbf{T} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F} + \mathbf{id}$
- $\mathbf{E} + \mathbf{T} + \mathbf{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{F} + \mathbf{id} + \mathbf{id}$
- $\mathbf{T} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F} + \mathbf{id}$
- $\mathbf{E} + \mathbf{T} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id}$



# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{F} + \mathbf{id} + \mathbf{id}$
- $\mathbf{T} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F} + \mathbf{id}$
- $\mathbf{E} + \mathbf{T} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id}$

# Example

- Grammar:

$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
  - $\mathbf{id} + \mathbf{id} + \mathbf{id}$
  - $\mathbf{F} + \mathbf{id} + \mathbf{id}$
  - $\mathbf{T} + \mathbf{id} + \mathbf{id}$
  - $\mathbf{E} + \mathbf{id} + \mathbf{id}$
  - $\mathbf{E} + \mathbf{F} + \mathbf{id}$
  - $\mathbf{E} + \mathbf{T} + \mathbf{id}$
  - $\mathbf{E} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F}$
  - $\mathbf{E} + \mathbf{T}$

# Example

- Grammar:

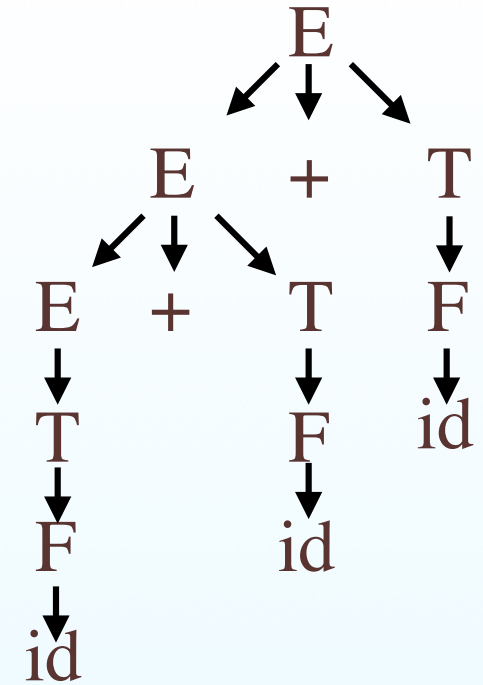
$$\mathbf{E} \rightarrow \mathbf{E} + \mathbf{T} \mid \mathbf{T}$$
$$\mathbf{T} \rightarrow \mathbf{T} * \mathbf{F} \mid \mathbf{F}$$
$$\mathbf{F} \rightarrow \mathbf{id} \mid ( \mathbf{E} )$$

- Parse:  $A + B + C$

- $A + B + C$  — assume lex can give  $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{id} + \mathbf{id} + \mathbf{id}$
- $\mathbf{F} + \mathbf{id} + \mathbf{id}$
- $\mathbf{T} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F} + \mathbf{id}$
- $\mathbf{E} + \mathbf{T} + \mathbf{id}$
- $\mathbf{E} + \mathbf{id}$
- $\mathbf{E} + \mathbf{F}$
- $\mathbf{E} + \mathbf{T}$
- $\mathbf{E}$

# Example

- Grammar:

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow \text{id} \mid ( E ) \end{aligned}$$


- Parse: A + B + C

- A + B + C — assume lex can give id + id + id
- id + id + id
- F + id + id
- T + id + id
- E + id + id
- E + F + id
- E + T + id
- E + id
- E + F
- E + T
- E

# Problem: Which RHS to choose?

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow \text{id} \mid ( E )$$

- If current sentential form is:  $E + T + \text{id}$
- Choose:
  - $E \rightarrow T$ ?
  - $E \rightarrow E + T$ ?
  - $T \rightarrow F$ ?

# Handles

- Problem: Which RHS to choose, if several?
- Find *handle* in current sentential form
  - Handle = *correct* pattern in RHS to replace s.t.:
    - $\Rightarrow$  new sentential form
    - & sentential form in right-derivation of current
- Only 1 handle/sentential form
- So...how to find handle?

# Finding the handle

- Right derivation of  $A + B * (C + D)$  given grammar:

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$
$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$
$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$$

# Finding the handle

Parse: A + B \* (C + D)

$$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$$
$$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$$
$$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$$



# Finding the handle

Parse: A + B \* (C + D)

<e>

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse: A + B \* (C + D)

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$



# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle t \rangle + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle t \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle f \rangle + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle t \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle f \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle id \rangle + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle t \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle f \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle id \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( C + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle t \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle f \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle id \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( C + D )$

$\langle e \rangle + \langle f \rangle * ( C + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle )$

$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle t \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle f \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( \langle id \rangle + D )$

$\langle e \rangle + \langle t \rangle * ( C + D )$

$\langle e \rangle + \langle f \rangle * ( C + D )$

$\langle e \rangle + \langle id \rangle * ( C + D )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B^* (C + D)$

$\langle e \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle + \langle t \rangle * \langle f \rangle$

$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$

$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$

$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$

$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$

$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$

$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$

$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$

$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$

$\langle e \rangle + \langle t \rangle * (C + D)$

$\langle e \rangle + \langle f \rangle * (C + D)$

$\langle e \rangle + \langle id \rangle * (C + D)$

$\langle e \rangle + B^* (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B^* (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B^* (C + D)$   
 $\langle t \rangle + B^* (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



# Finding the handle

Parse:  $A + B^* (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B^* (C + D)$   
 $\langle t \rangle + B^* (C + D)$   
 $\langle f \rangle + B^* (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B^* (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B^* (C + D)$   
 $\langle t \rangle + B^* (C + D)$   
 $\langle f \rangle + B^* (C + D)$   
 $\langle id \rangle + B^* (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 $\langle f \rangle + B * (C + D)$   
 $\langle id \rangle + B * (C + D)$   
 $A + B * (C + D)$

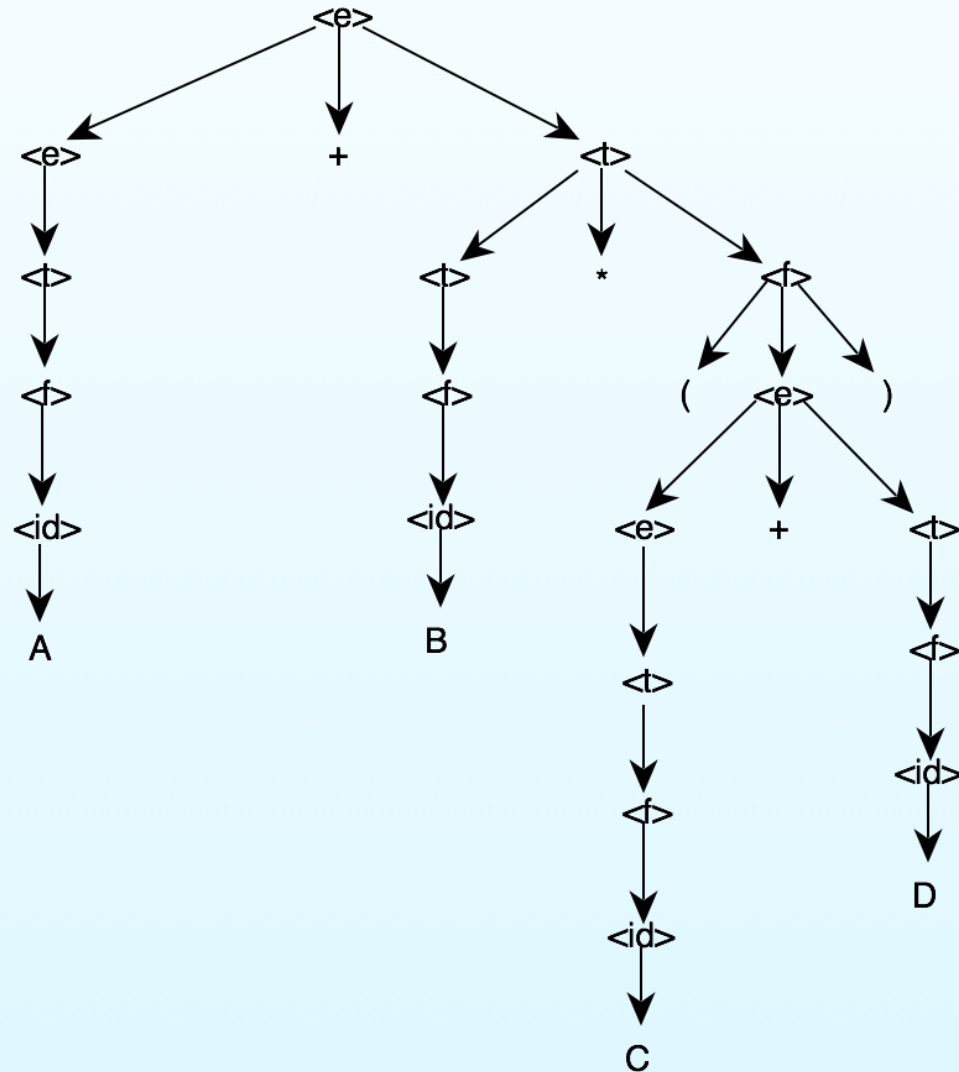
$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 $\langle f \rangle + B * (C + D)$   
 $\langle id \rangle + B * (C + D)$   
 $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

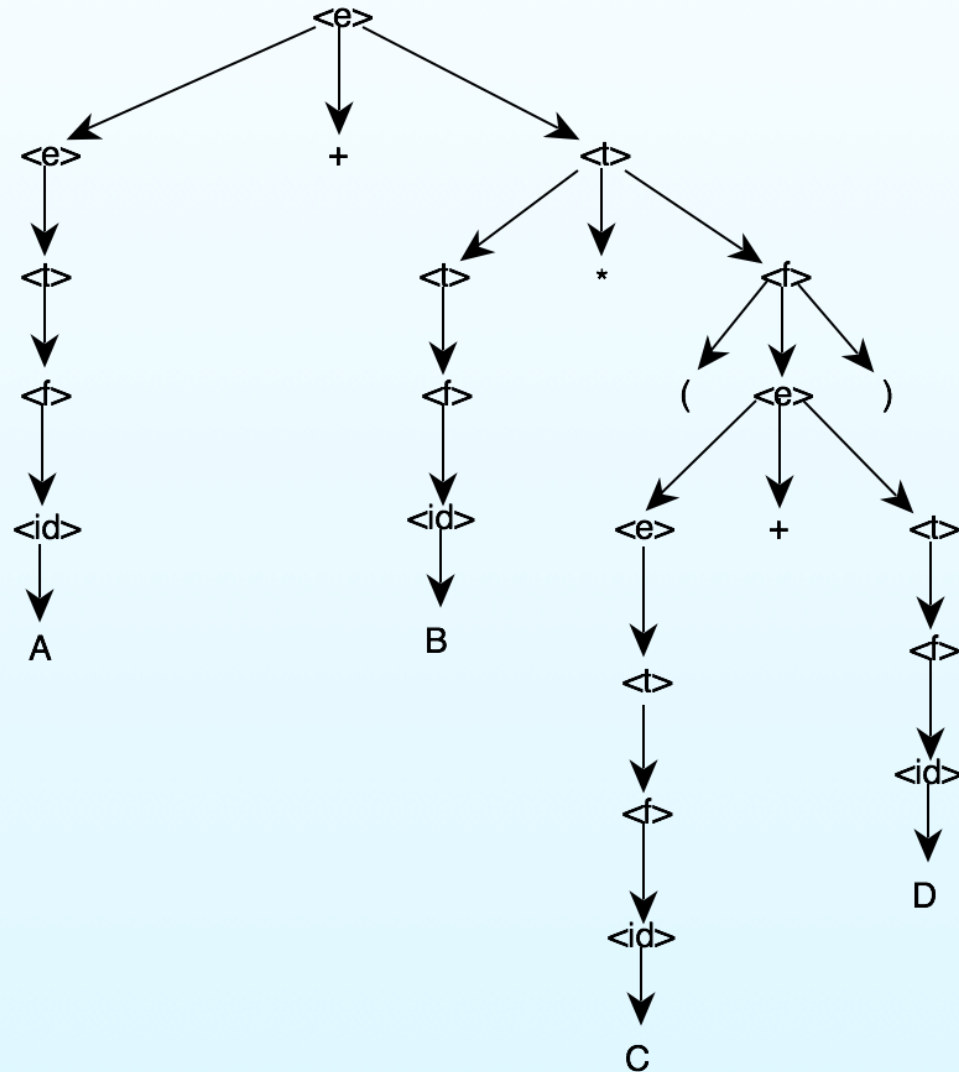


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 $\langle f \rangle + B * (C + D)$   
 $\langle id \rangle + B * (C + D)$   
 $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

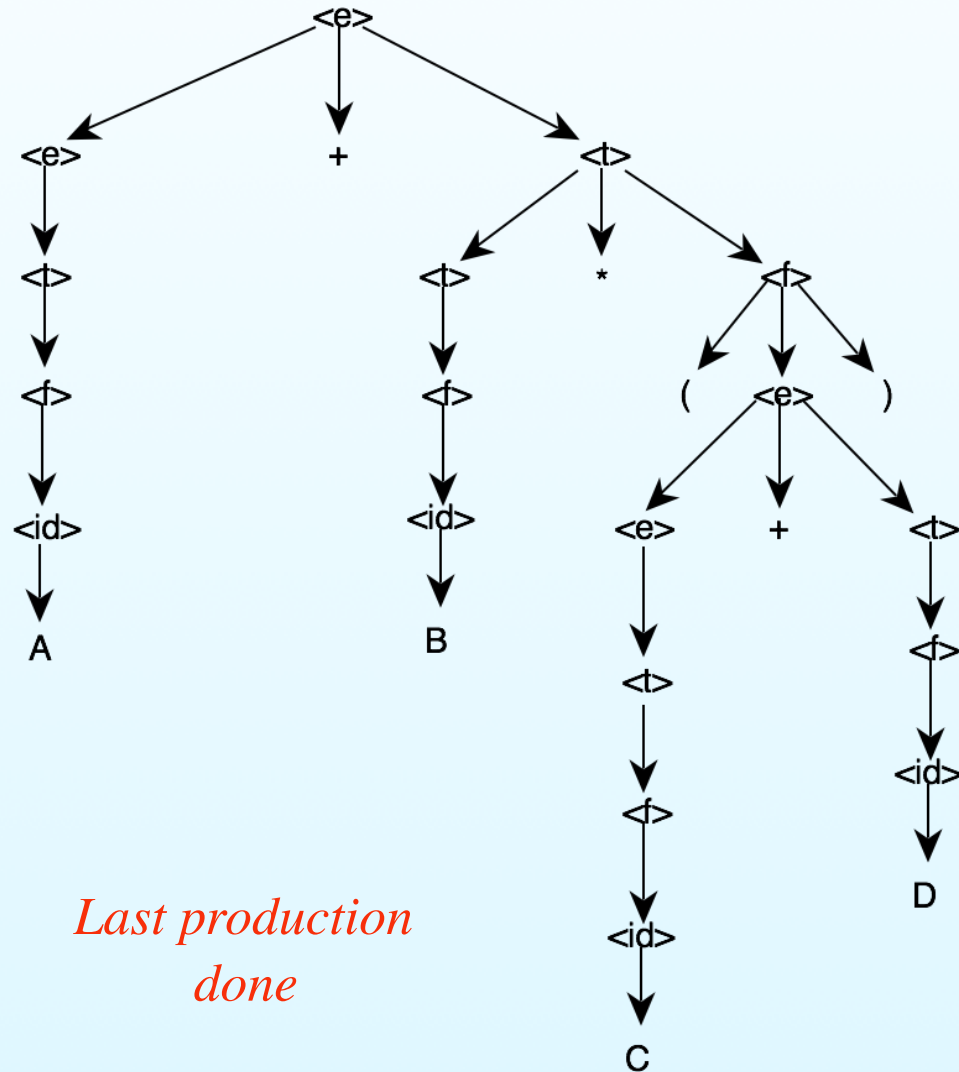


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 $\langle f \rangle + B * (C + D)$   
 $\langle id \rangle + B * (C + D)$   
 $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

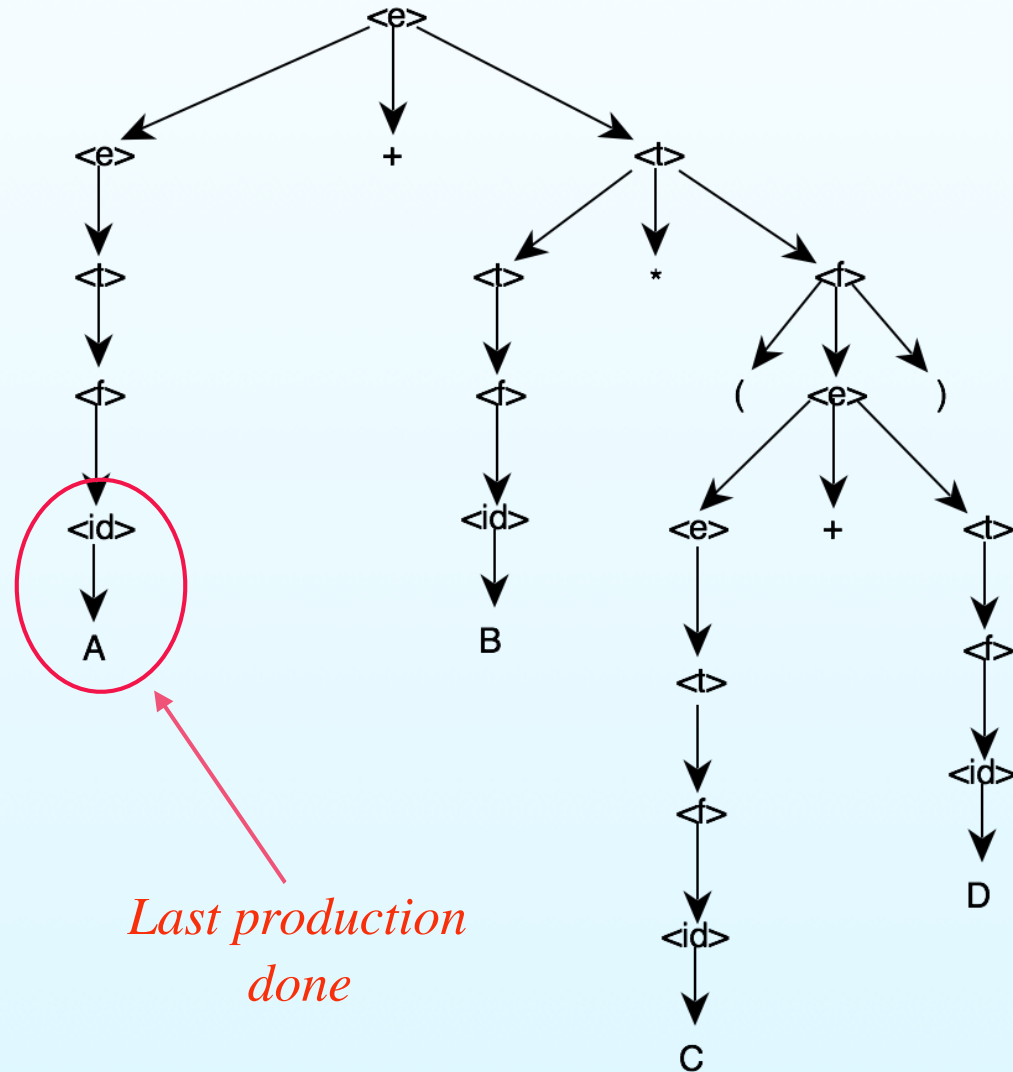


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 $\langle f \rangle + B * (C + D)$   
 $\langle id \rangle + B * (C + D)$   
 $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---



*Last production done*

# Finding the handle

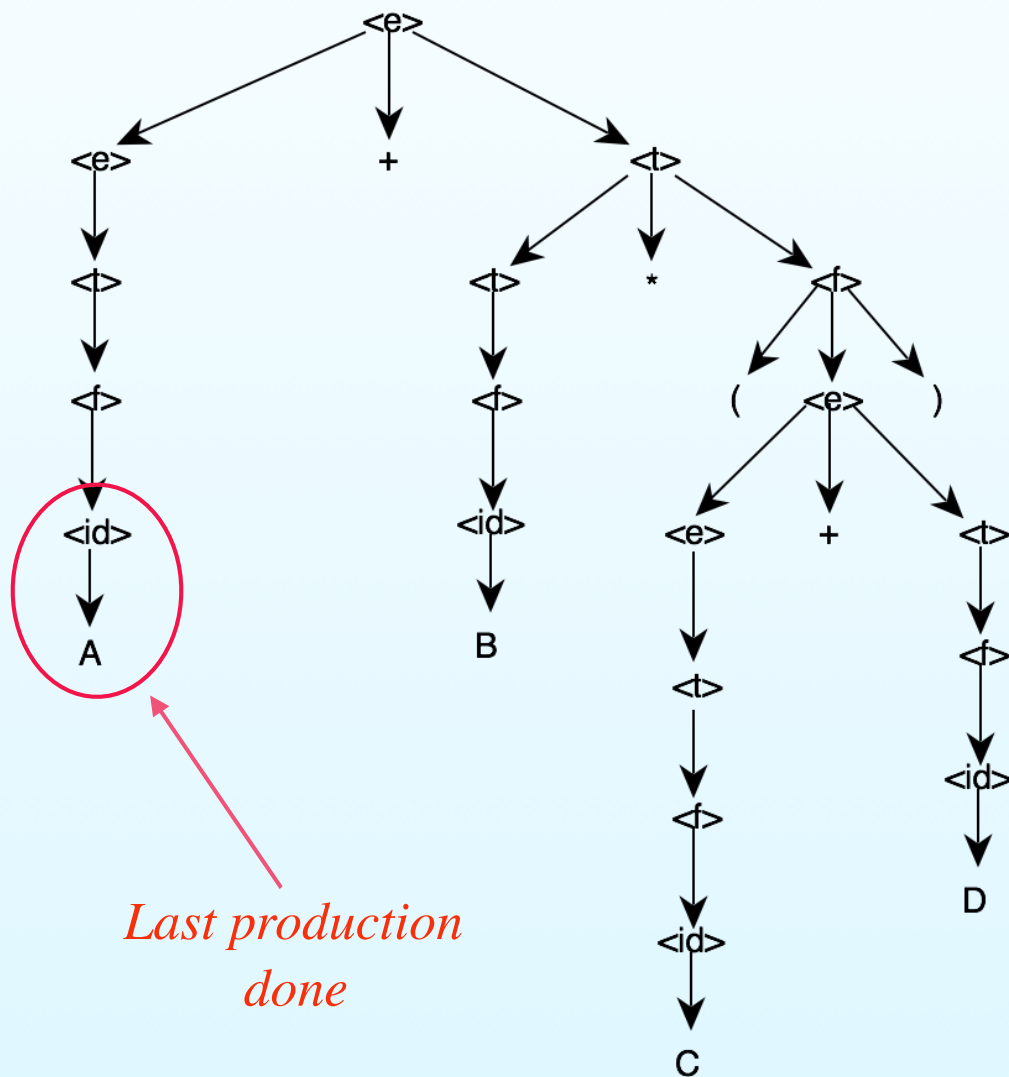
Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- $A + B * (C + D)$

*Handle*

*Last production done*



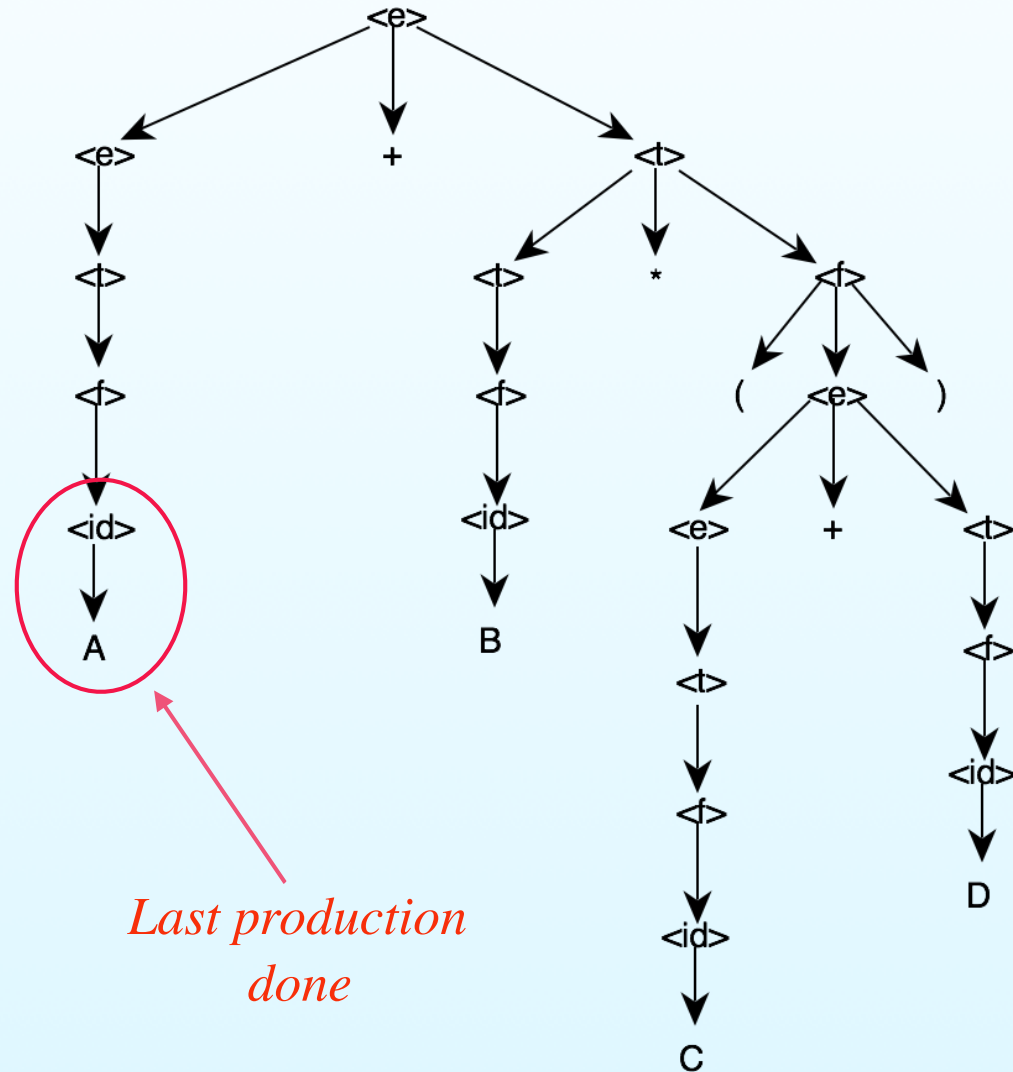


# Finding the handle

Parse:  $A + B^* (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B^* (C + D)$
- $\langle t \rangle + B^* (C + D)$
- $\langle f \rangle + B^* (C + D)$
- $\langle id \rangle + B^* (C + D)$
- $A + B^* (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



*Last production done*

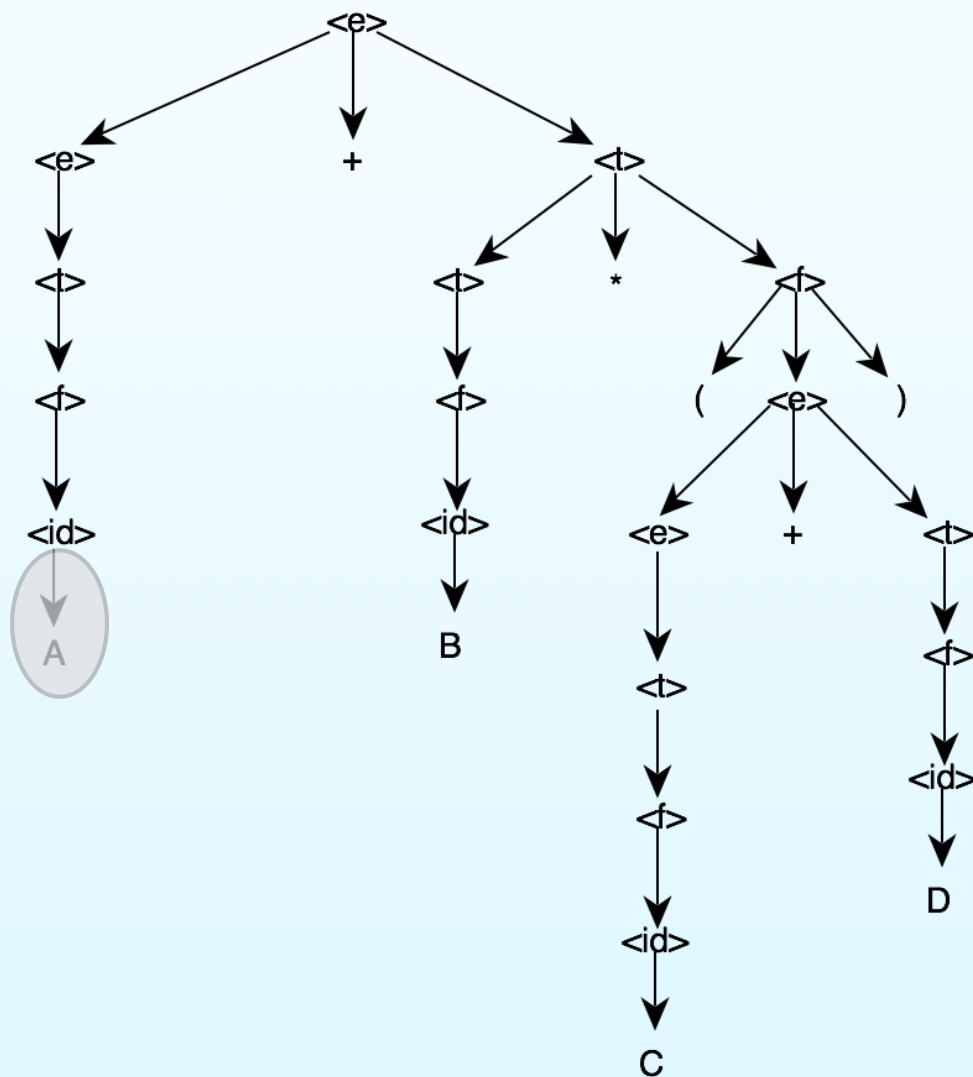
*Handle*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~

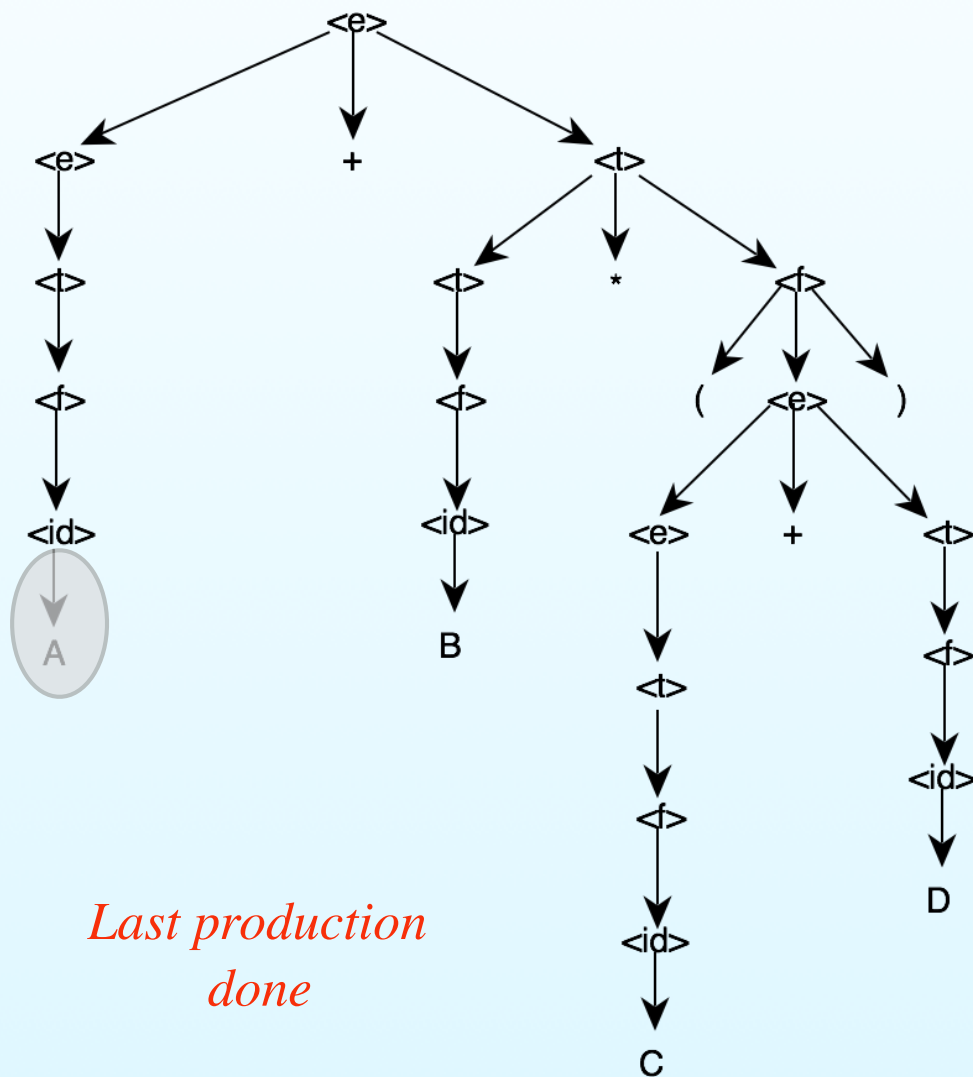


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~



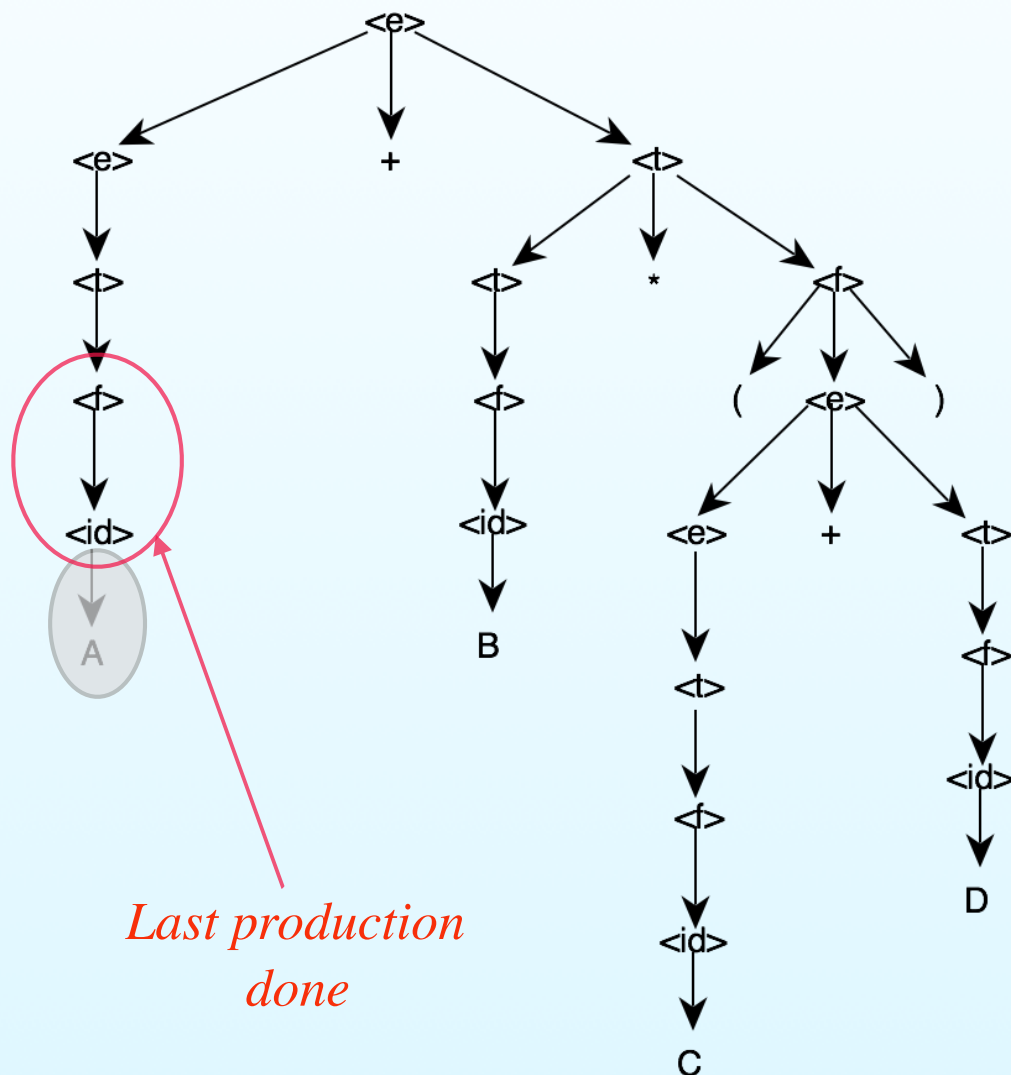
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~



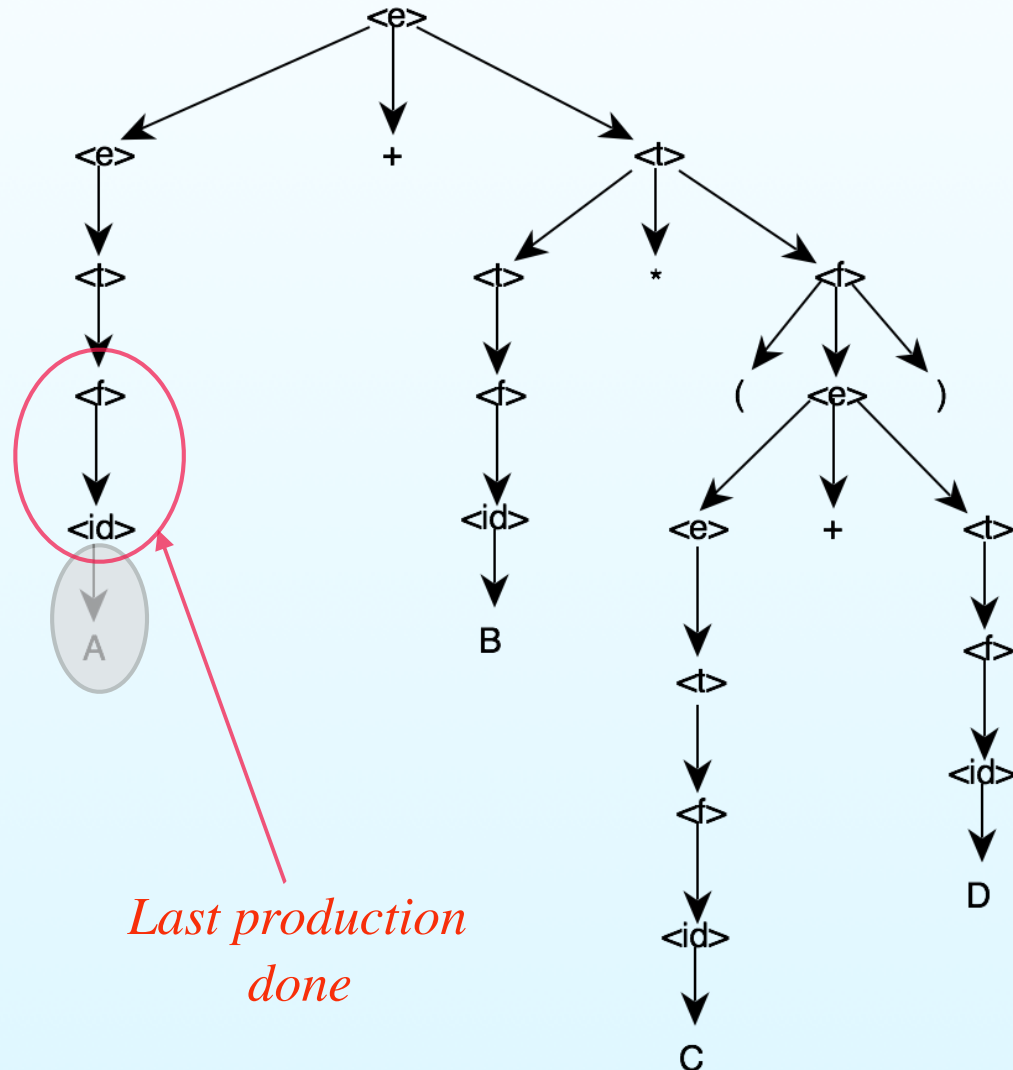
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

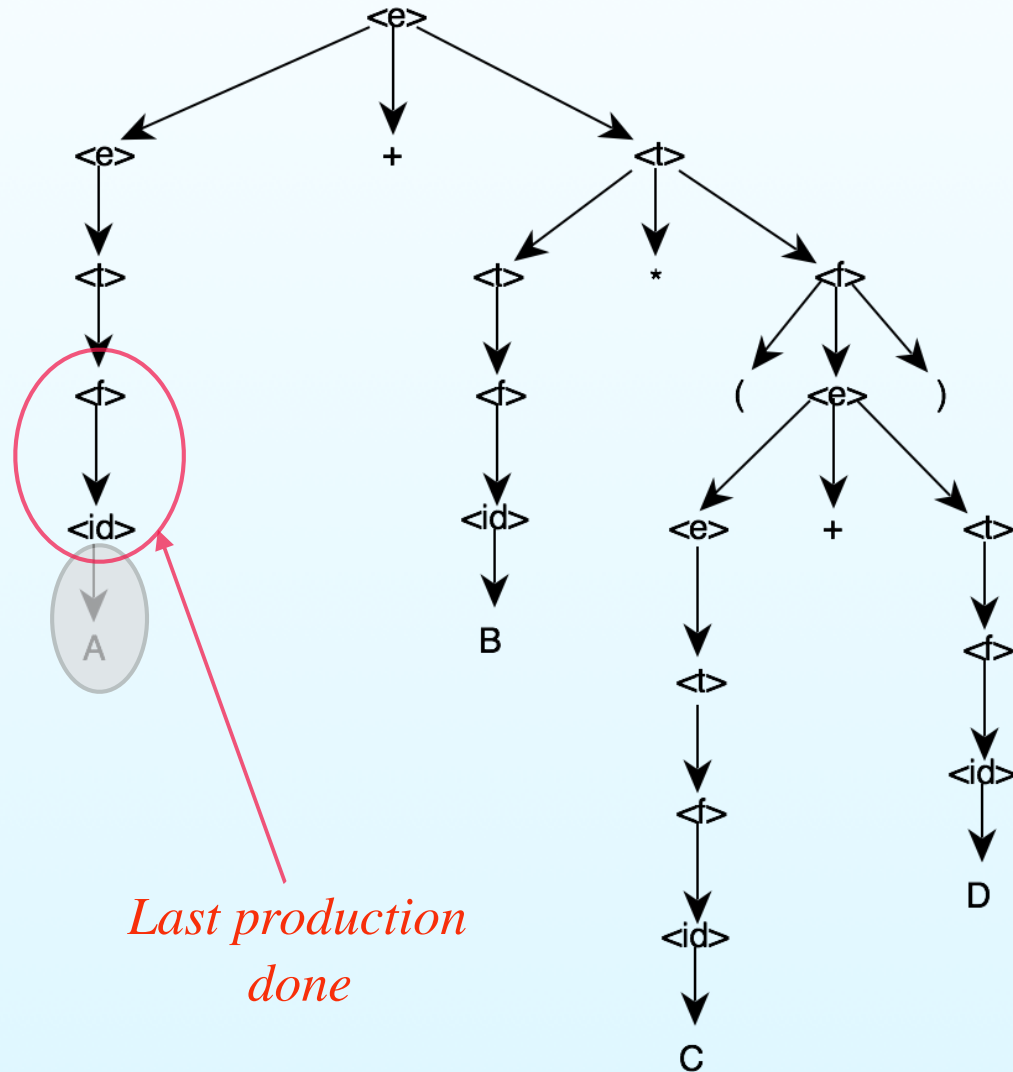


# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



*Last production done*

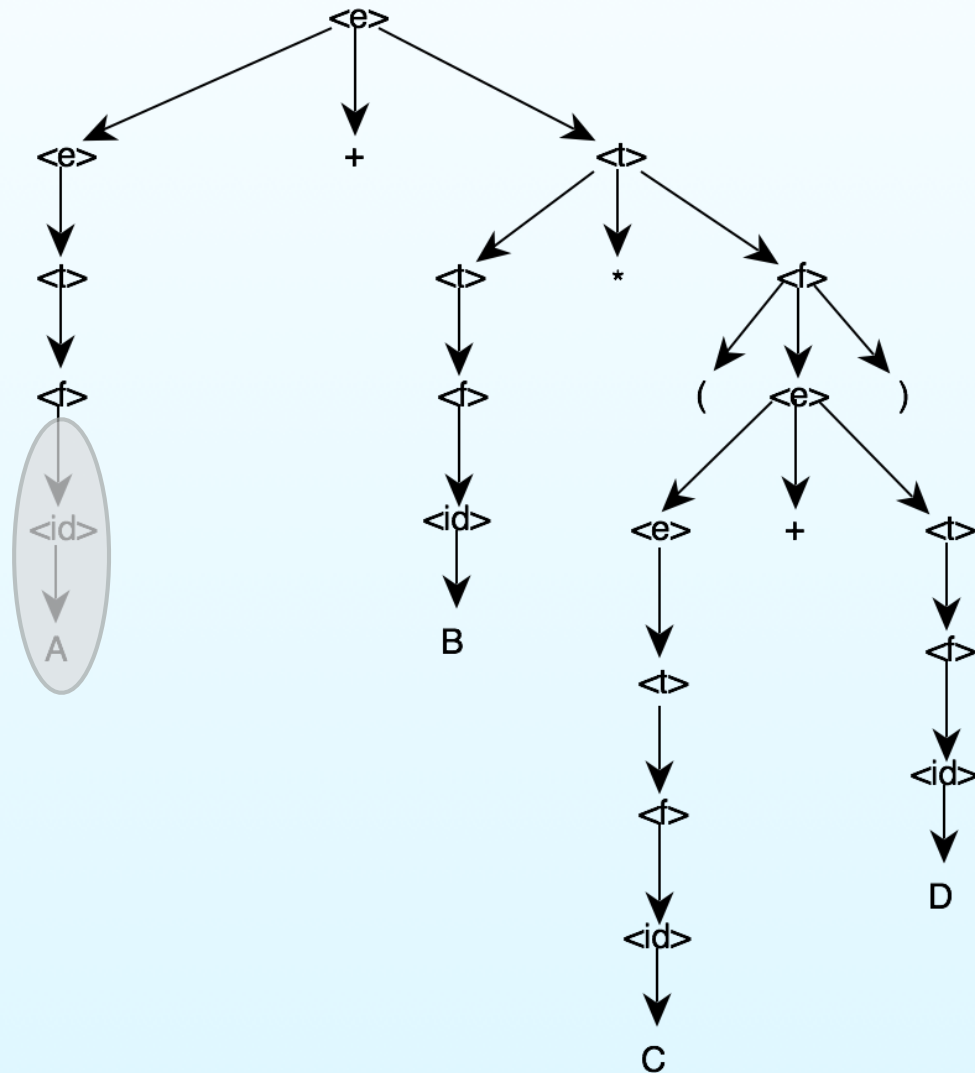
*Handle*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 $\langle f \rangle + B * (C + D)$   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

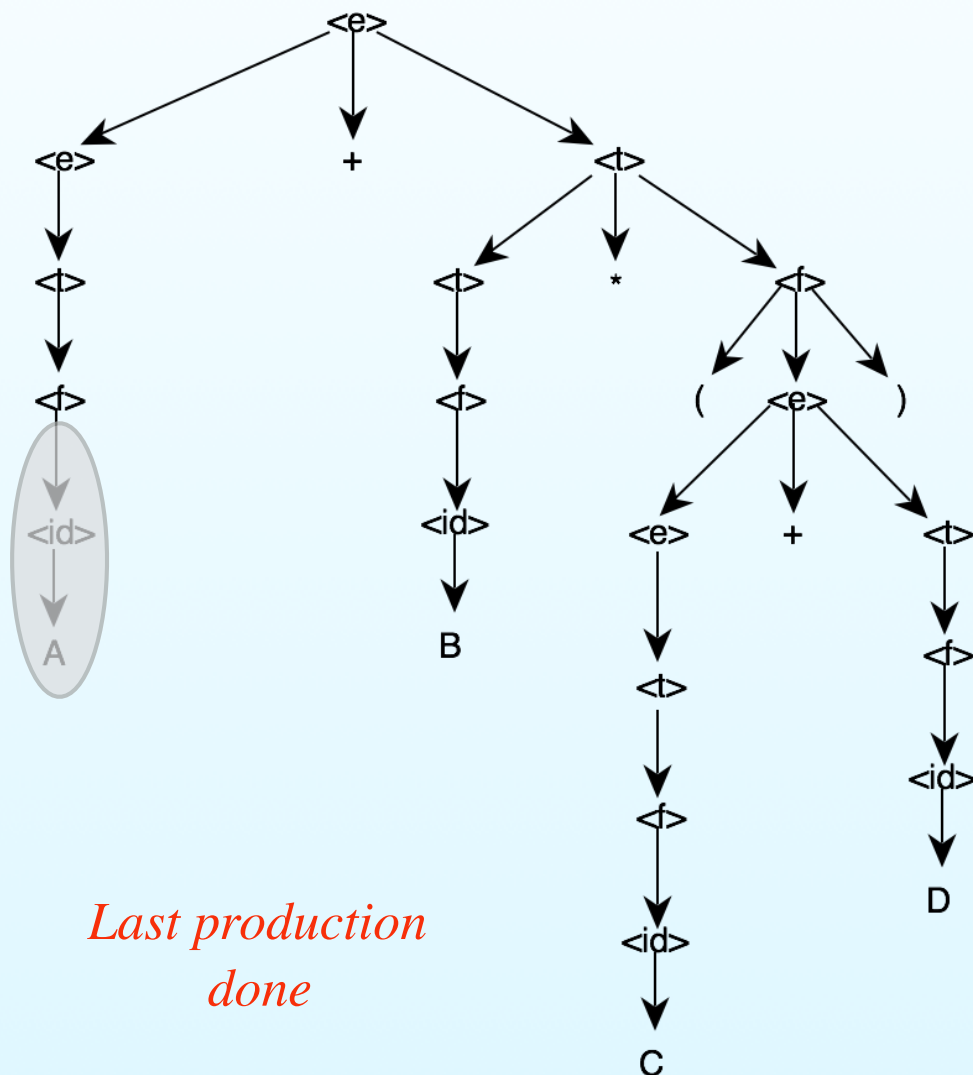


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Last production done*

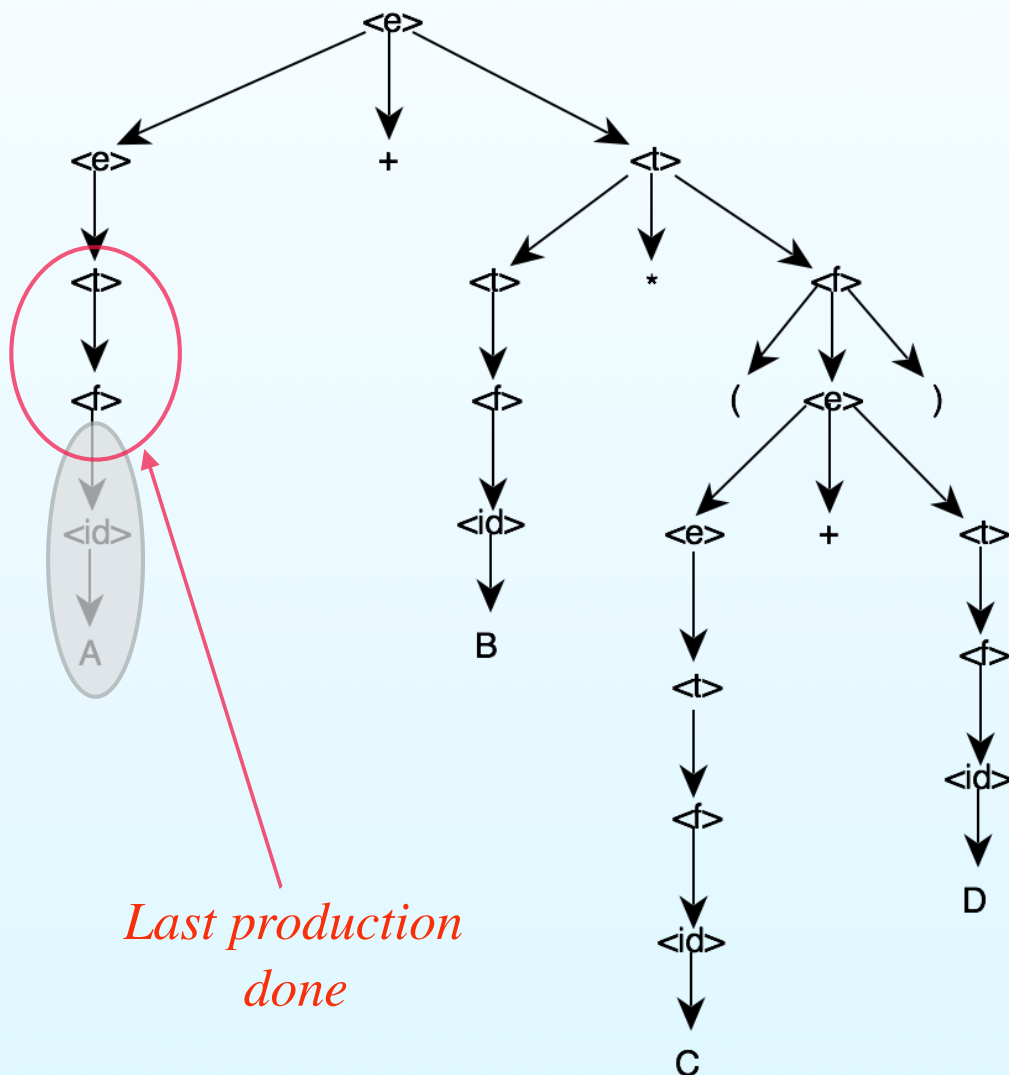


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Last production done*

# Finding the handle

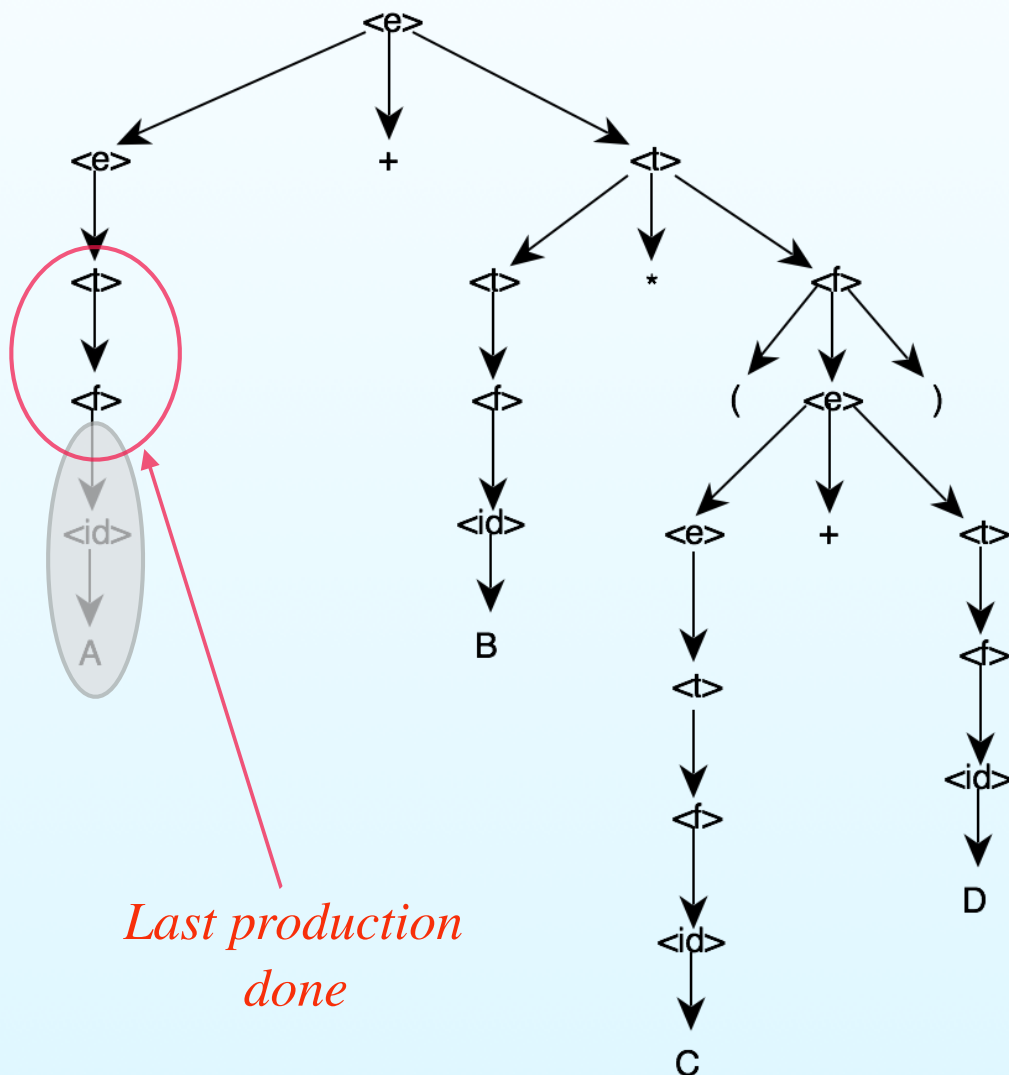
Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*

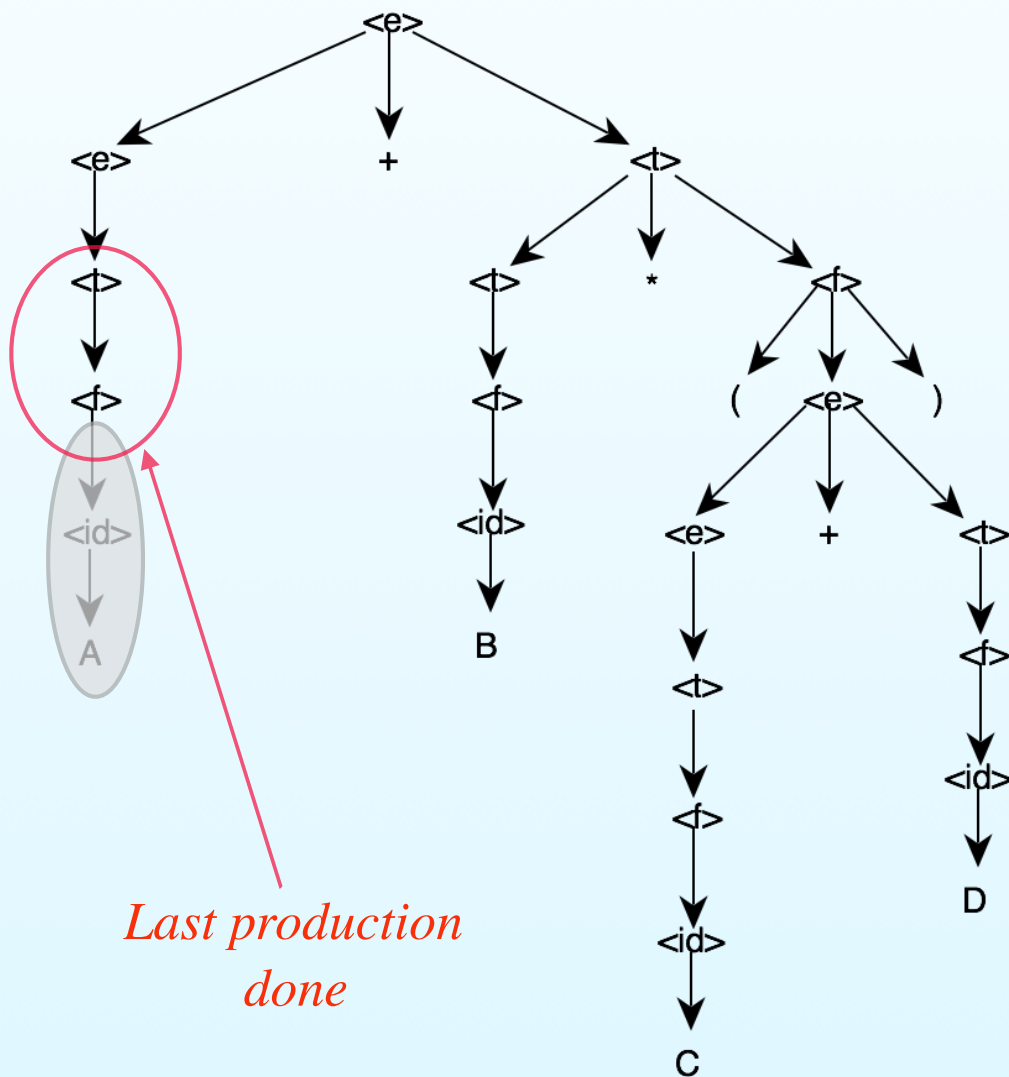


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Handle*

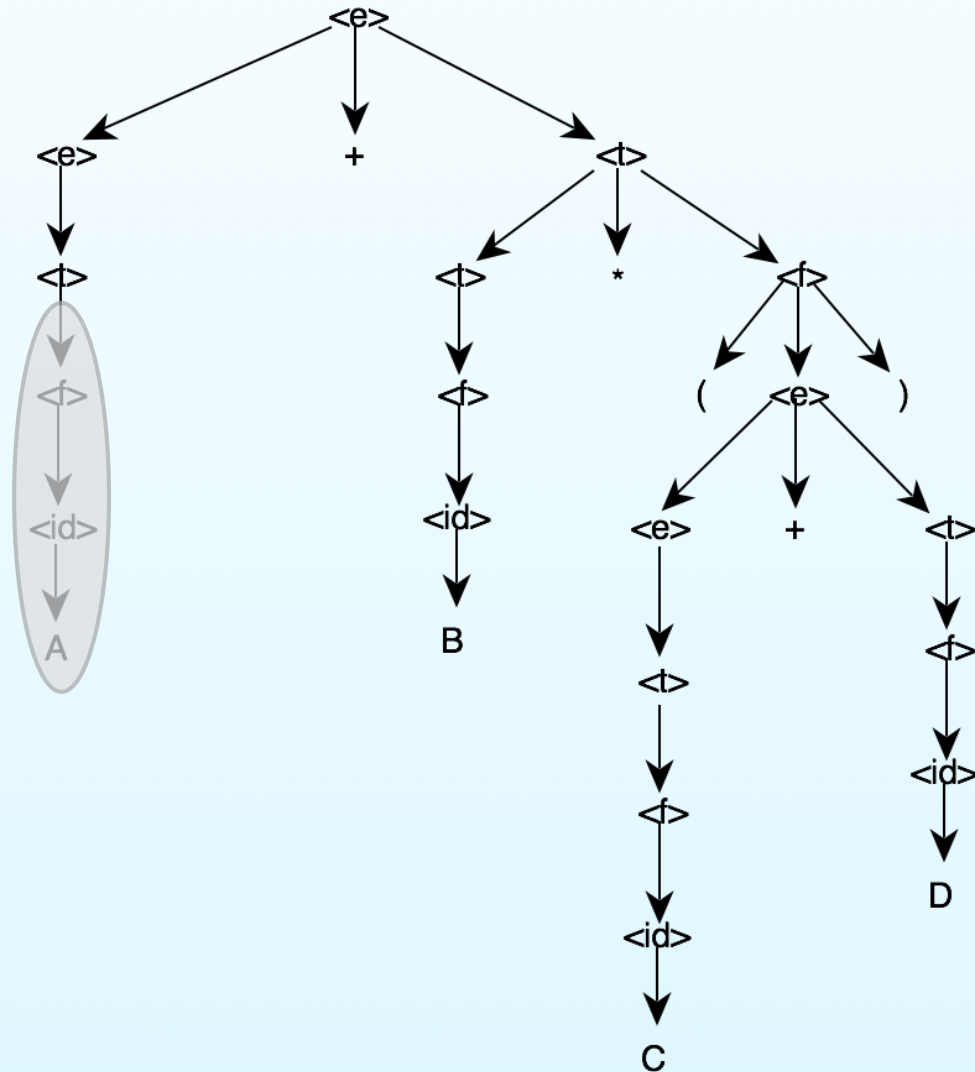
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 $\langle e \rangle + \langle f \rangle * (C + D)$   
 $\langle e \rangle + \langle id \rangle * (C + D)$   
 $\langle e \rangle + B * (C + D)$   
 $\langle t \rangle + B * (C + D)$   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

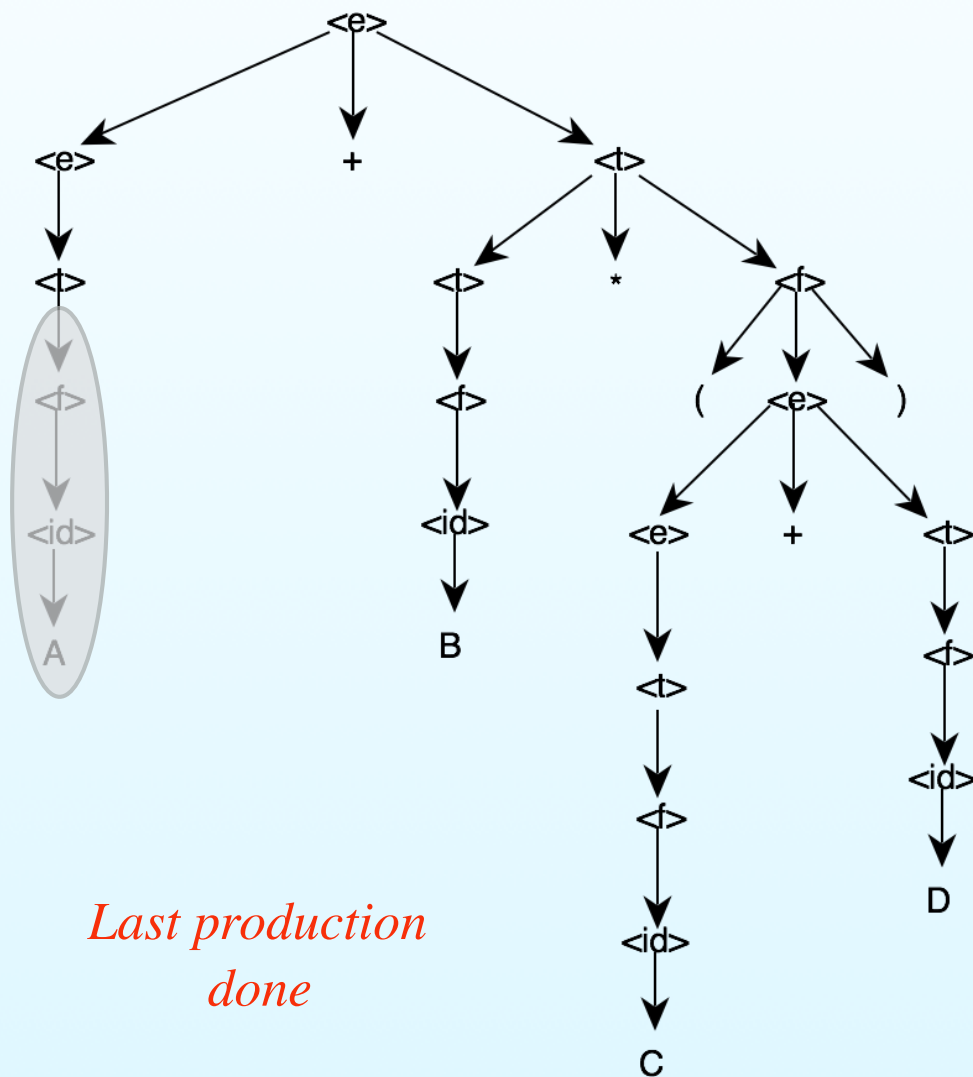


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



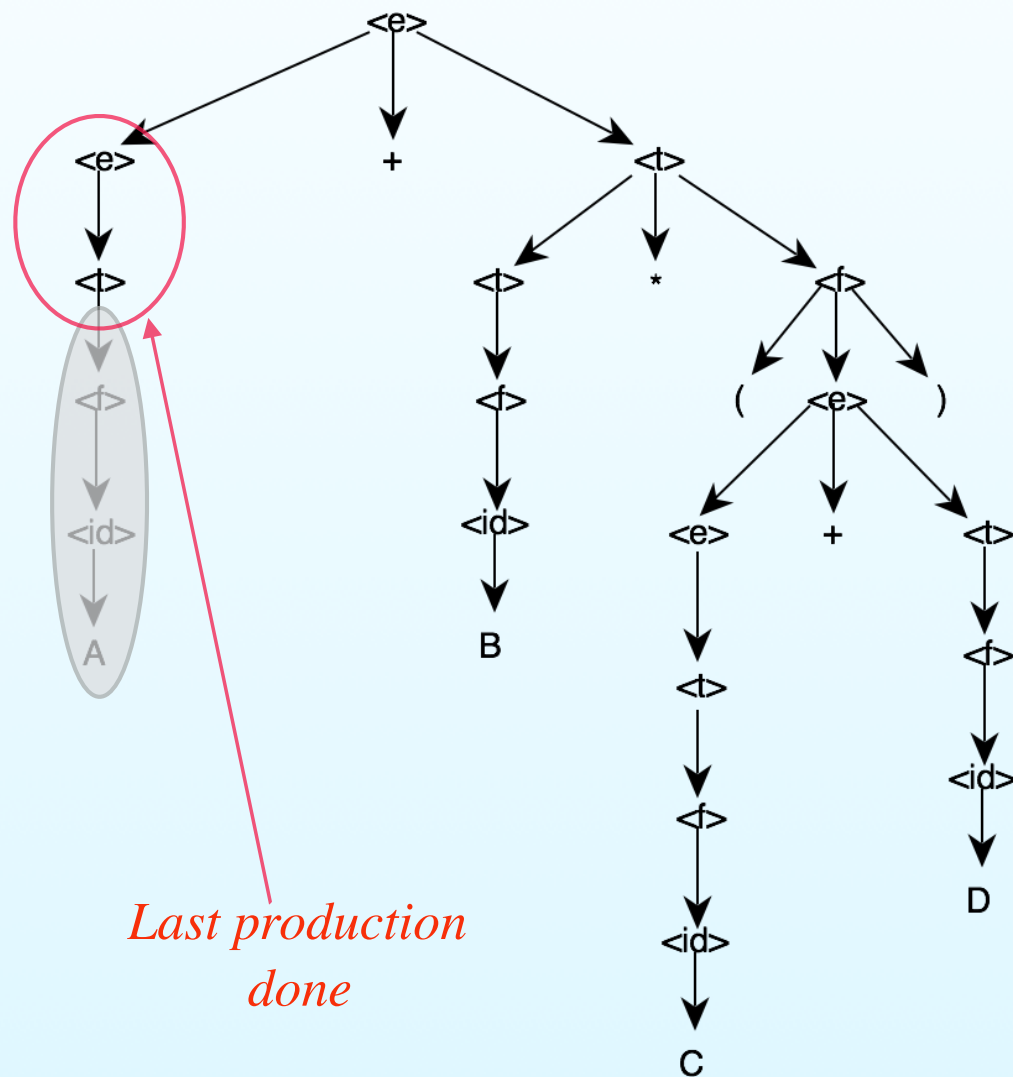
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



# Finding the handle

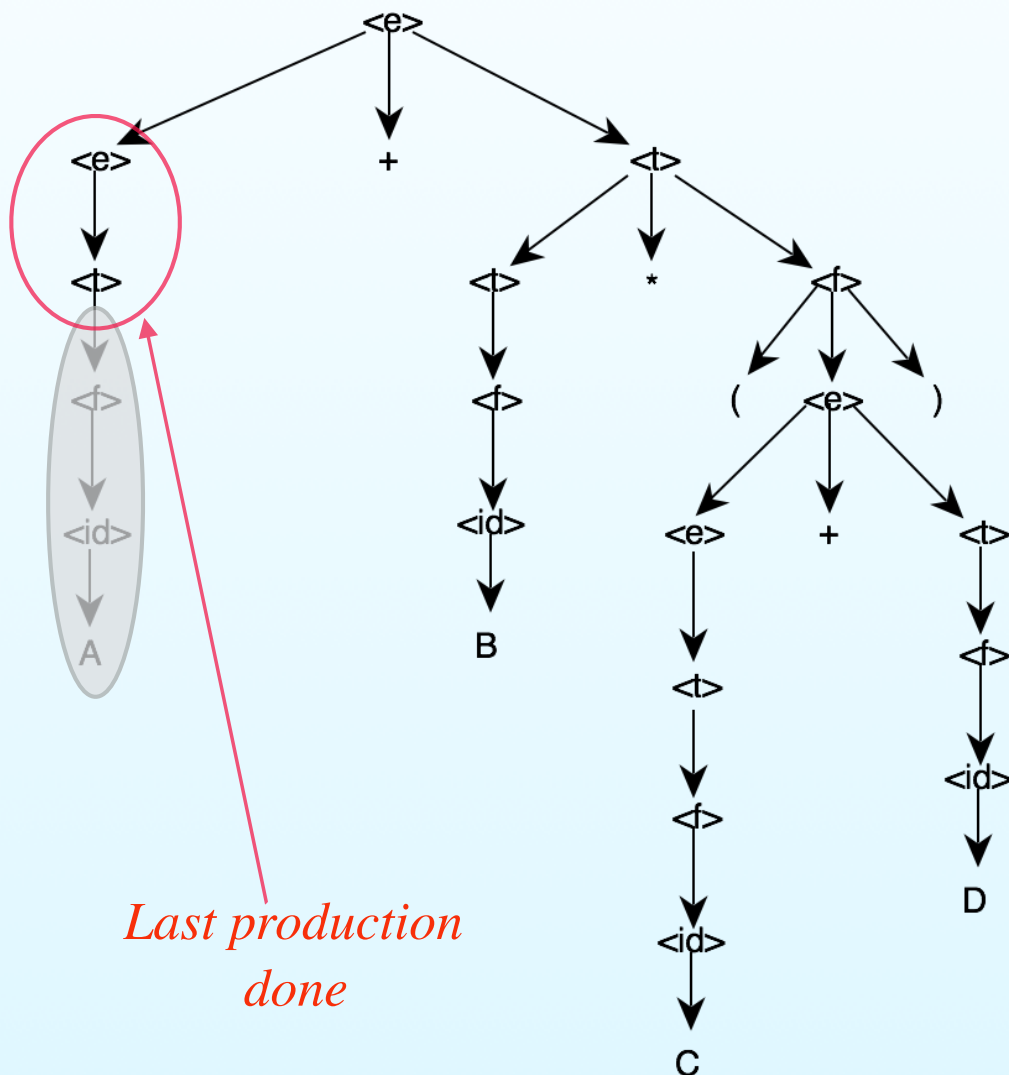
Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*



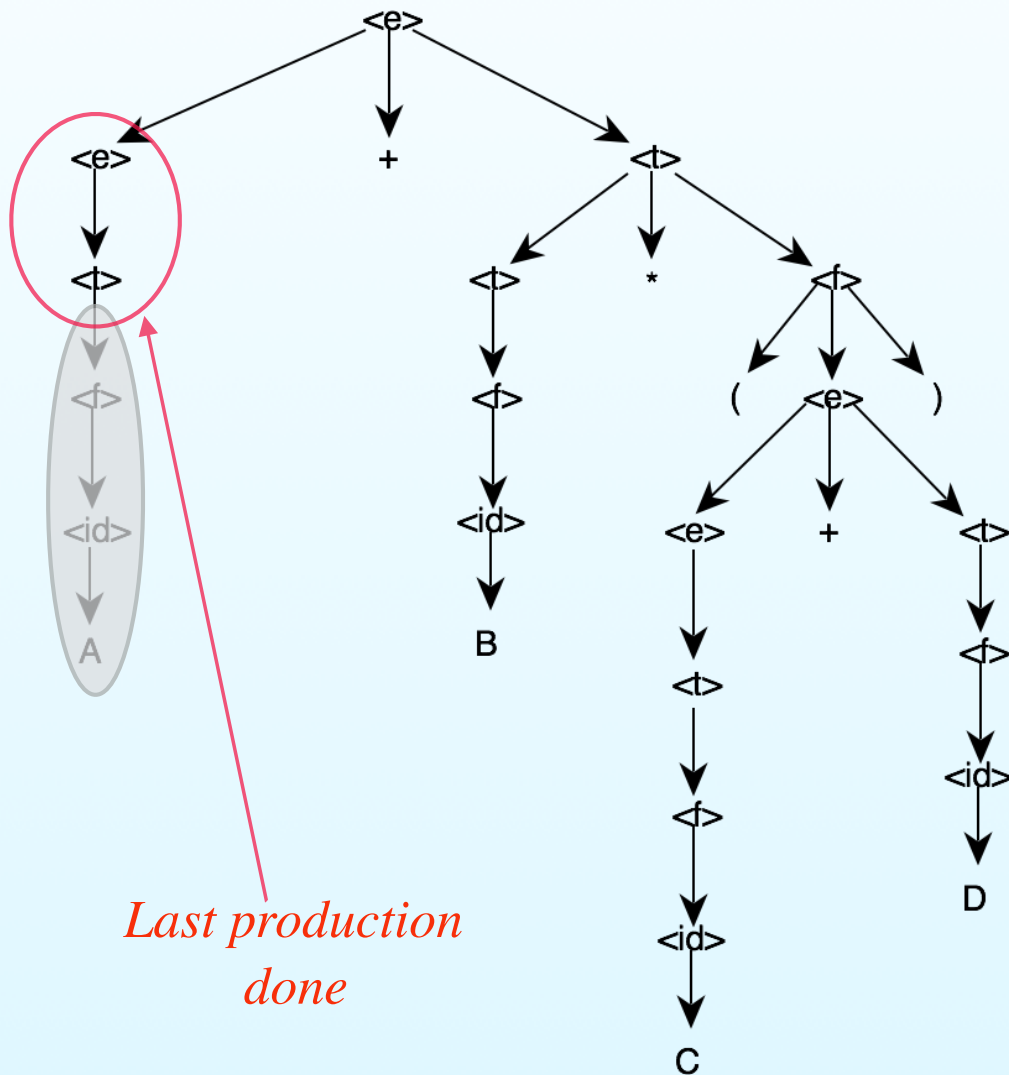
# Finding the handle

Parse: A + B \* (C + D)

- <e>
- <e> + <t>
- <e> + <t> \* <f>
- <e> + <t> \* (<e>)
- <e> + <t> \* (<e> + <t>)
- <e> + <t> \* (<e> + <f>)
- <e> + <t> \* (<e> + <id>)
- <e> + <t> \* (<e> + D)
- <e> + <t> \* (<t> + D)
- <e> + <t> \* (<f> + D)
- <e> + <t> \* (<id> + D)
- <e> + <t> \* (C + D)
- <e> + <f> \* (C + D)
- <e> + <id> \* (C + D)
- <e> + B \* (C + D)
- <t> + B \* (C + D)**
- ~~<f> + B \* (C + D)~~
- ~~<id> + B \* (C + D)~~
- ~~A + B \* (C + D)~~

*Handle*

<e> → <e> + <t> | <t>  
 <t> → <t> \* <f> | <f>  
 <f> → (<e>) | <id>



*Last production done*

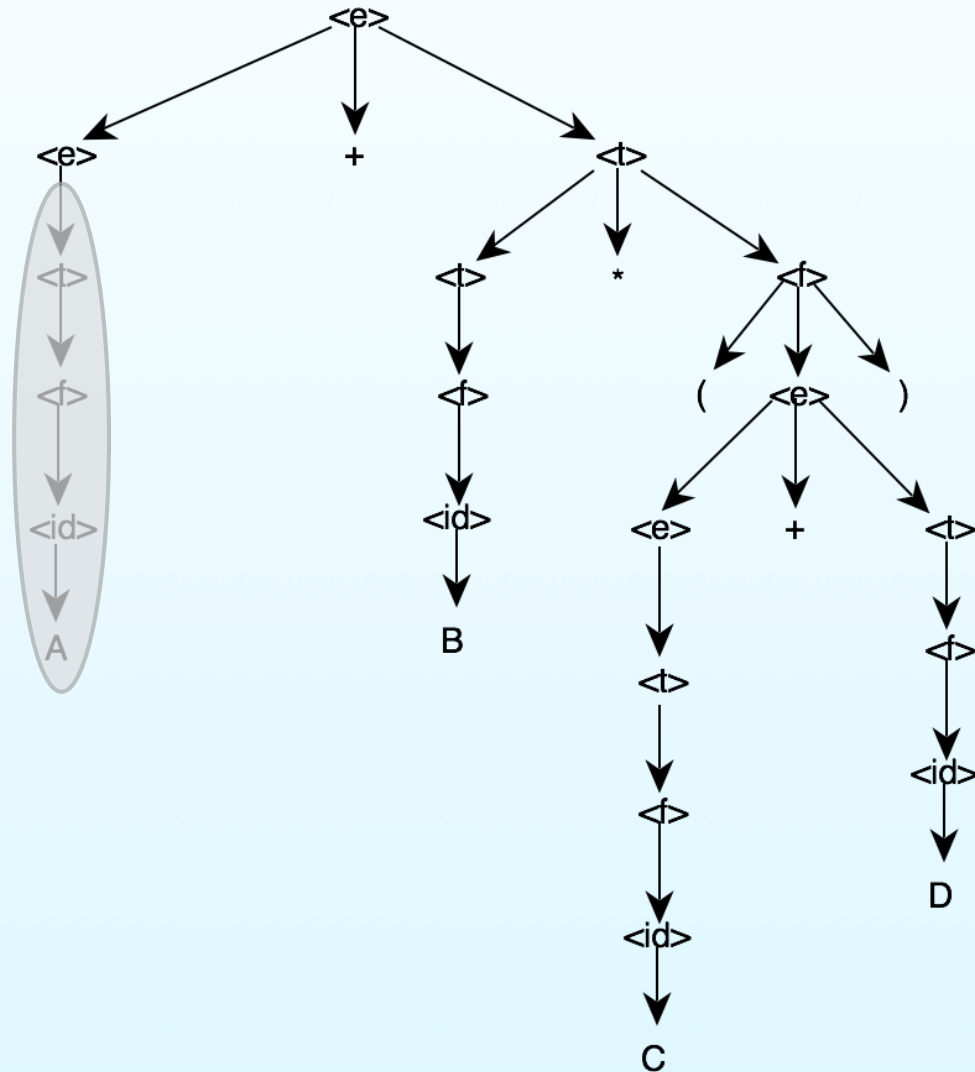


# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

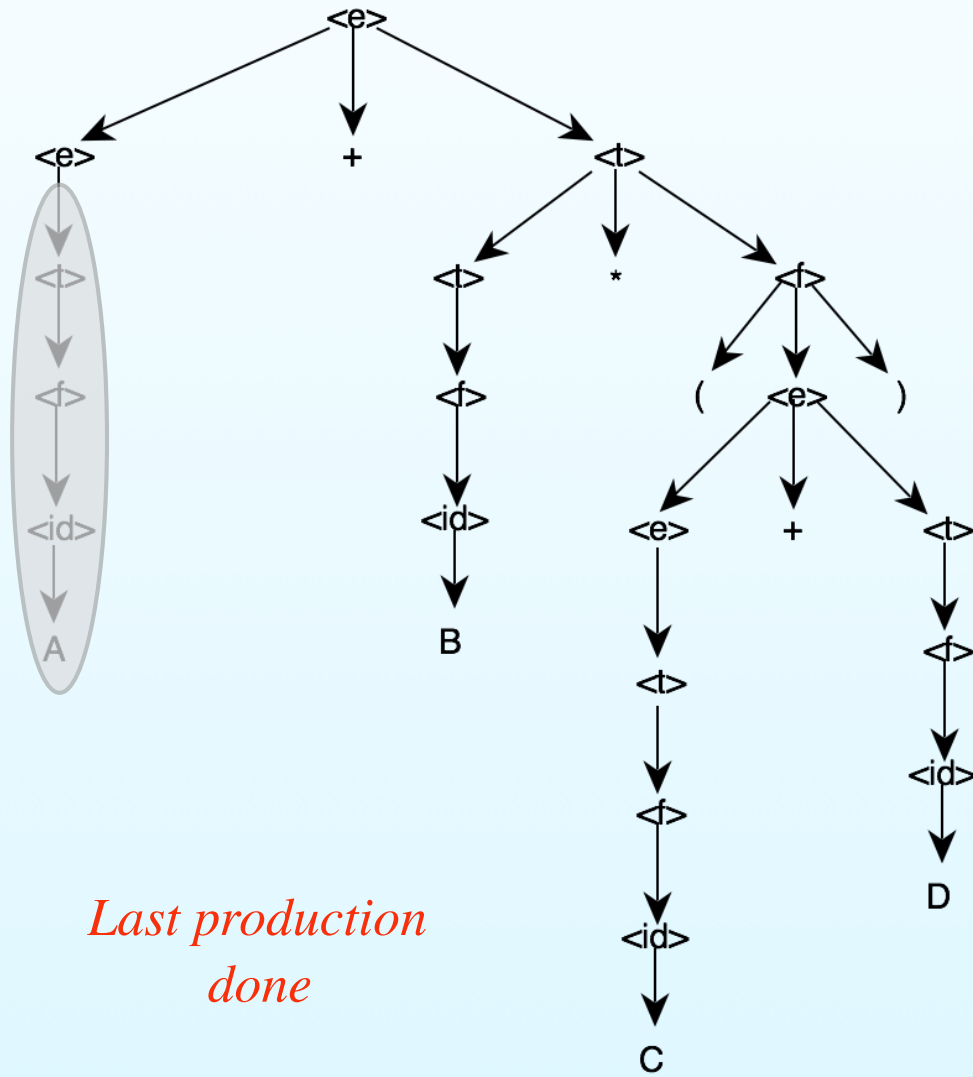


# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



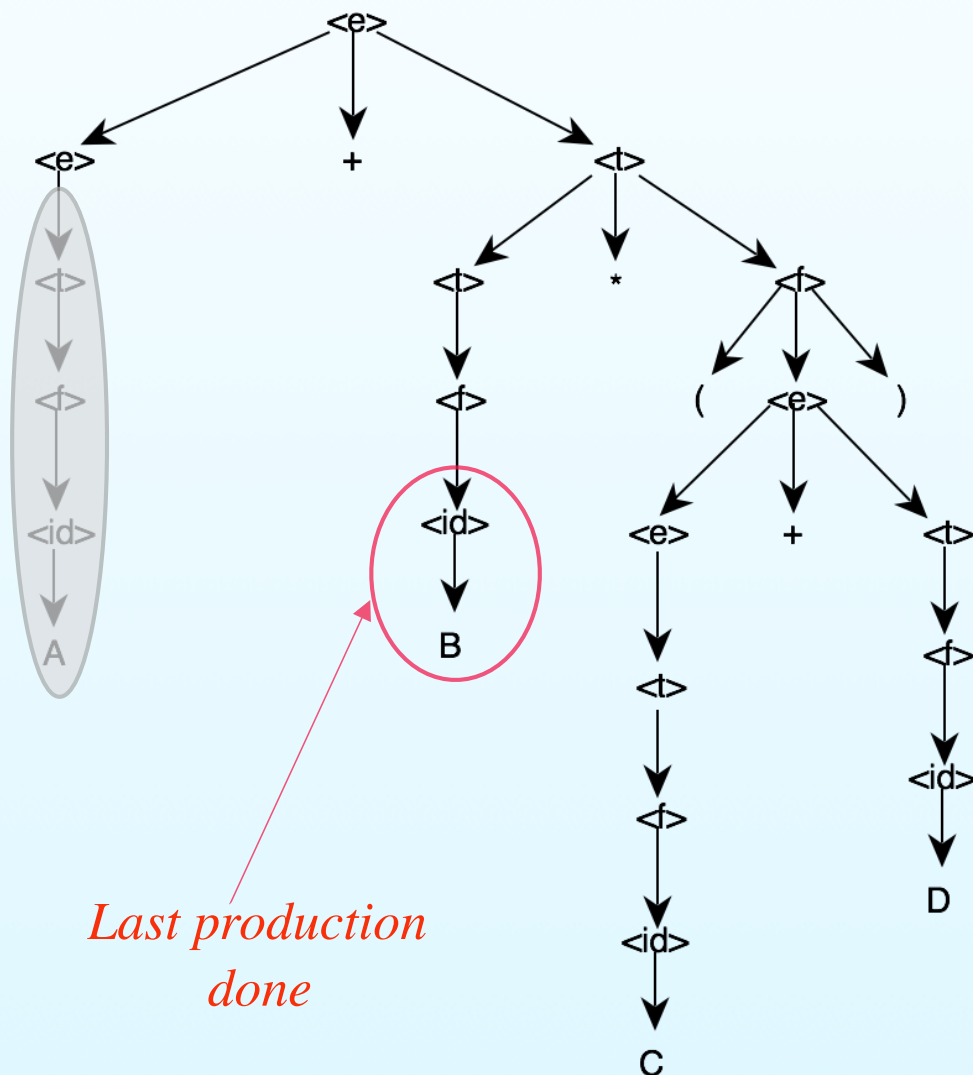
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Last production done*

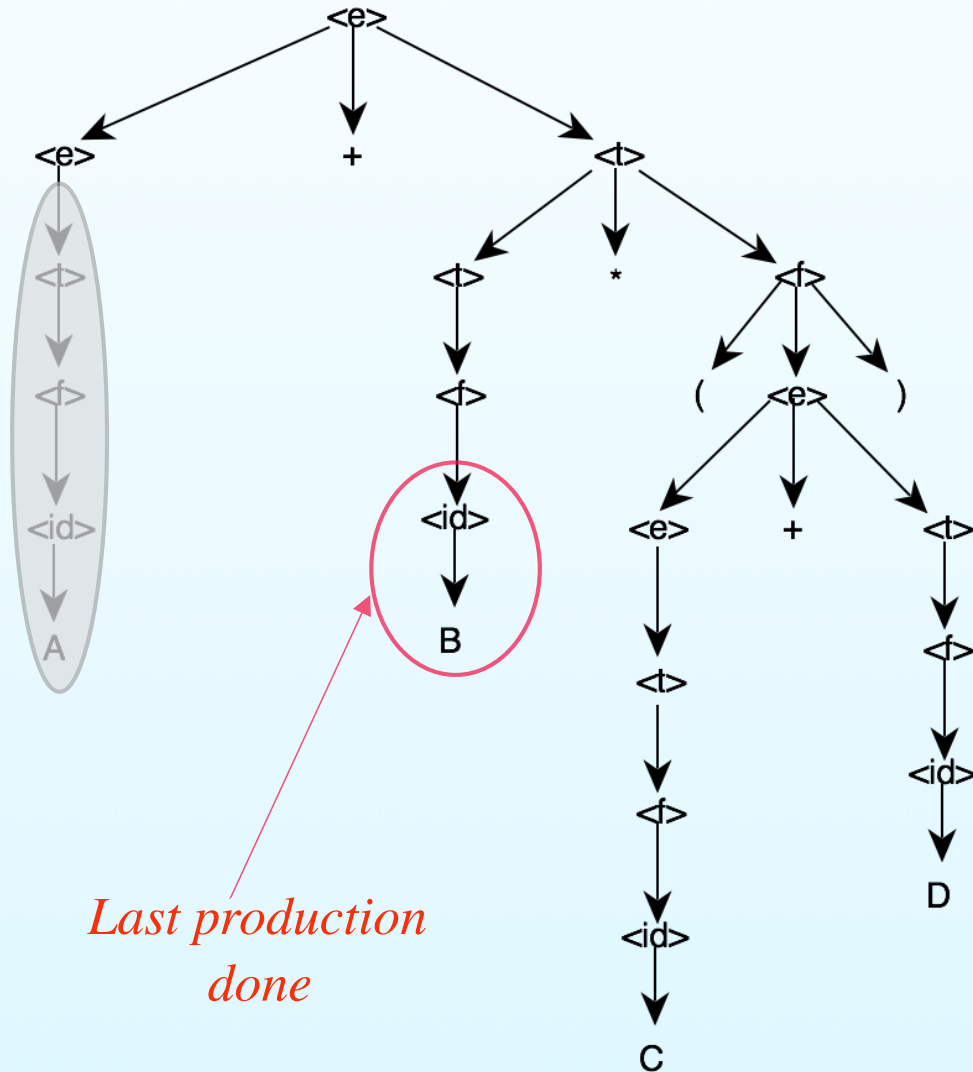
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



*Last production done*

# Finding the handle

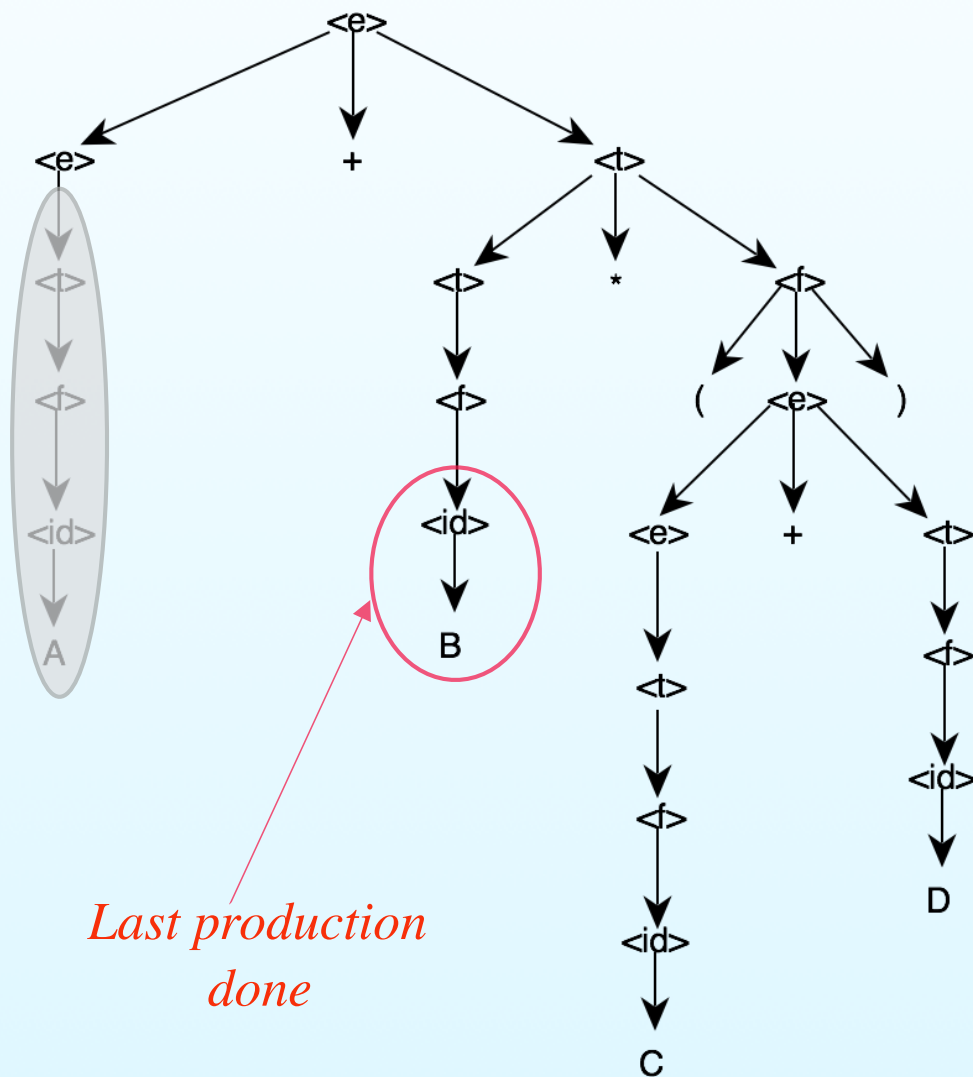
Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*

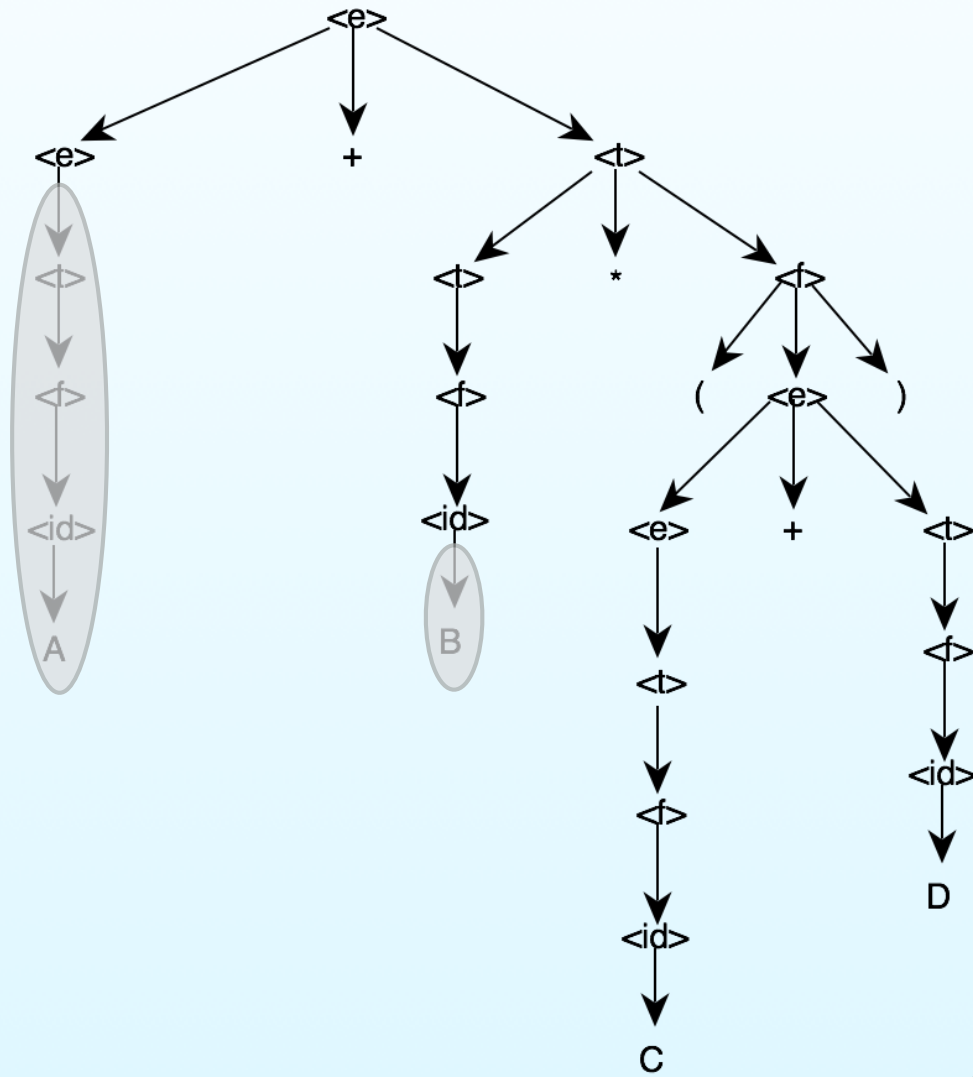


# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

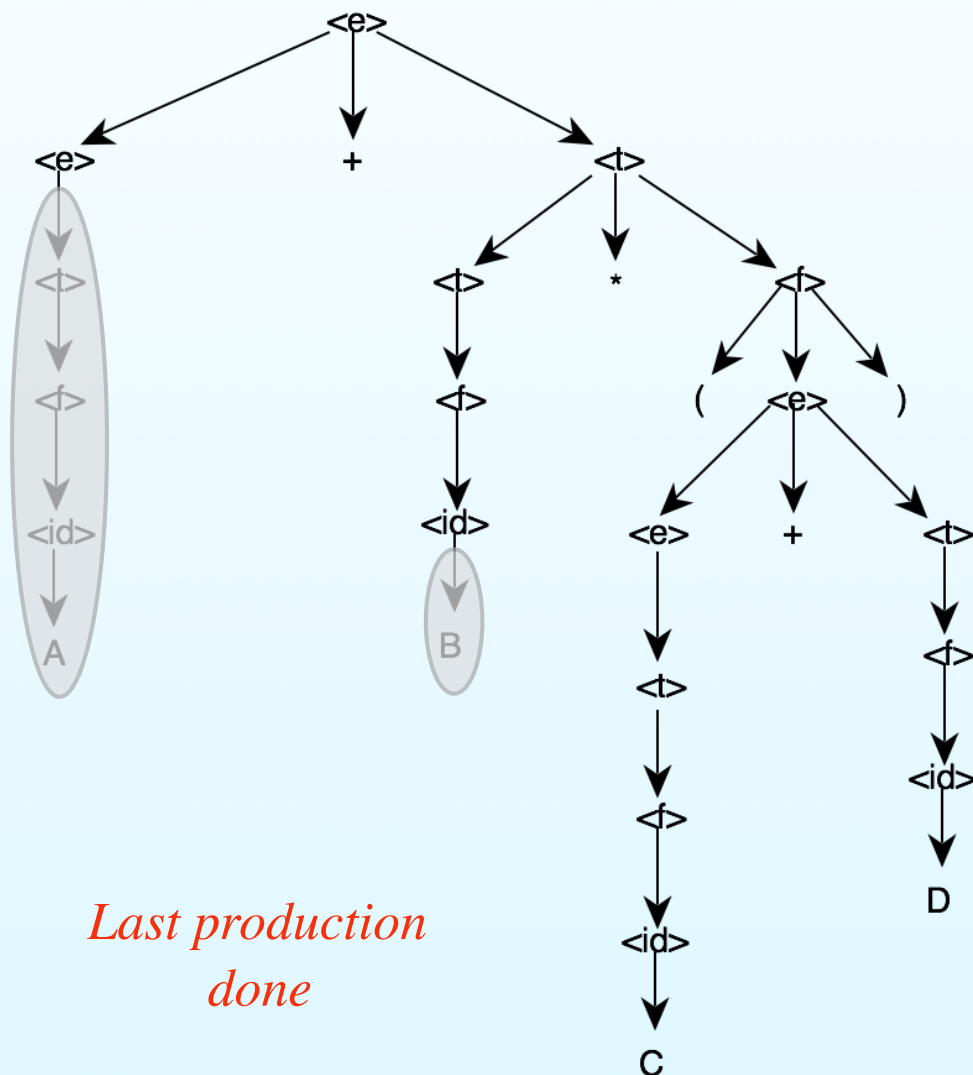


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



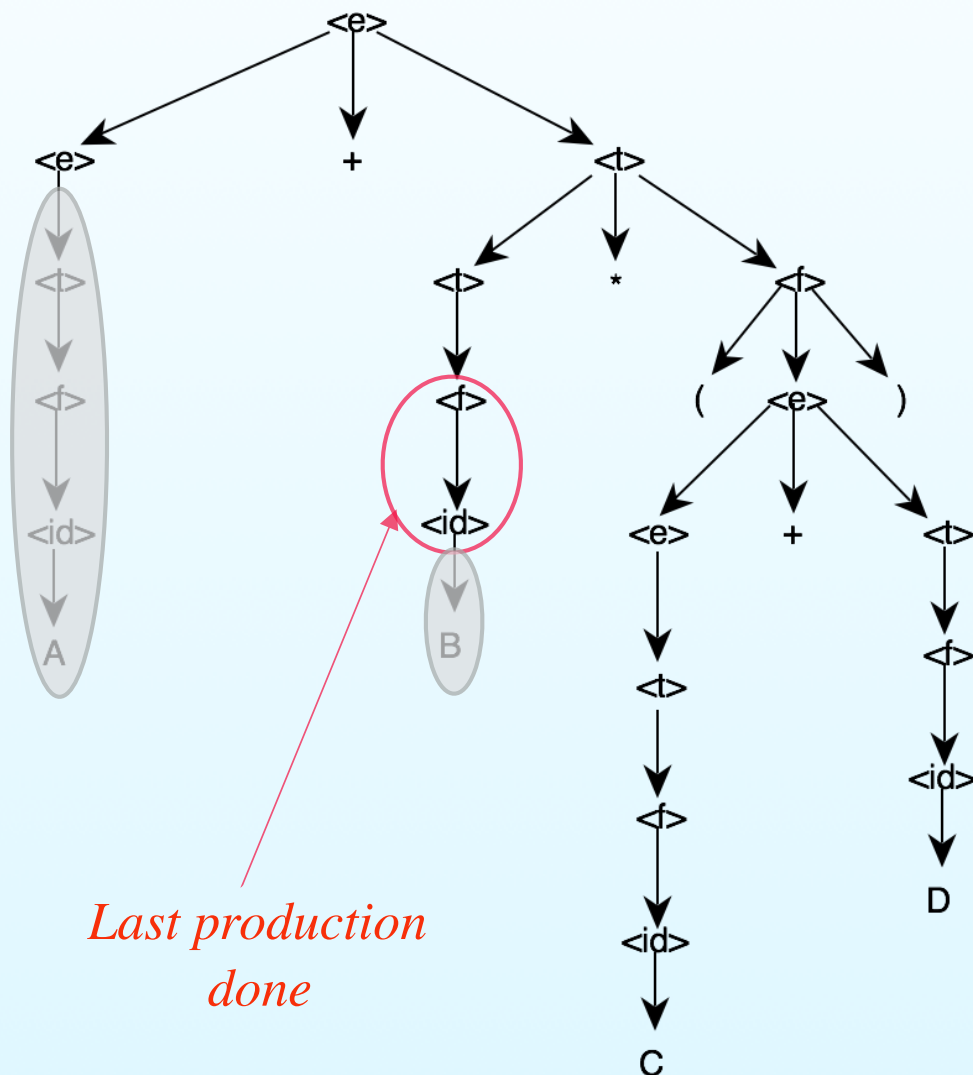
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Last production done*



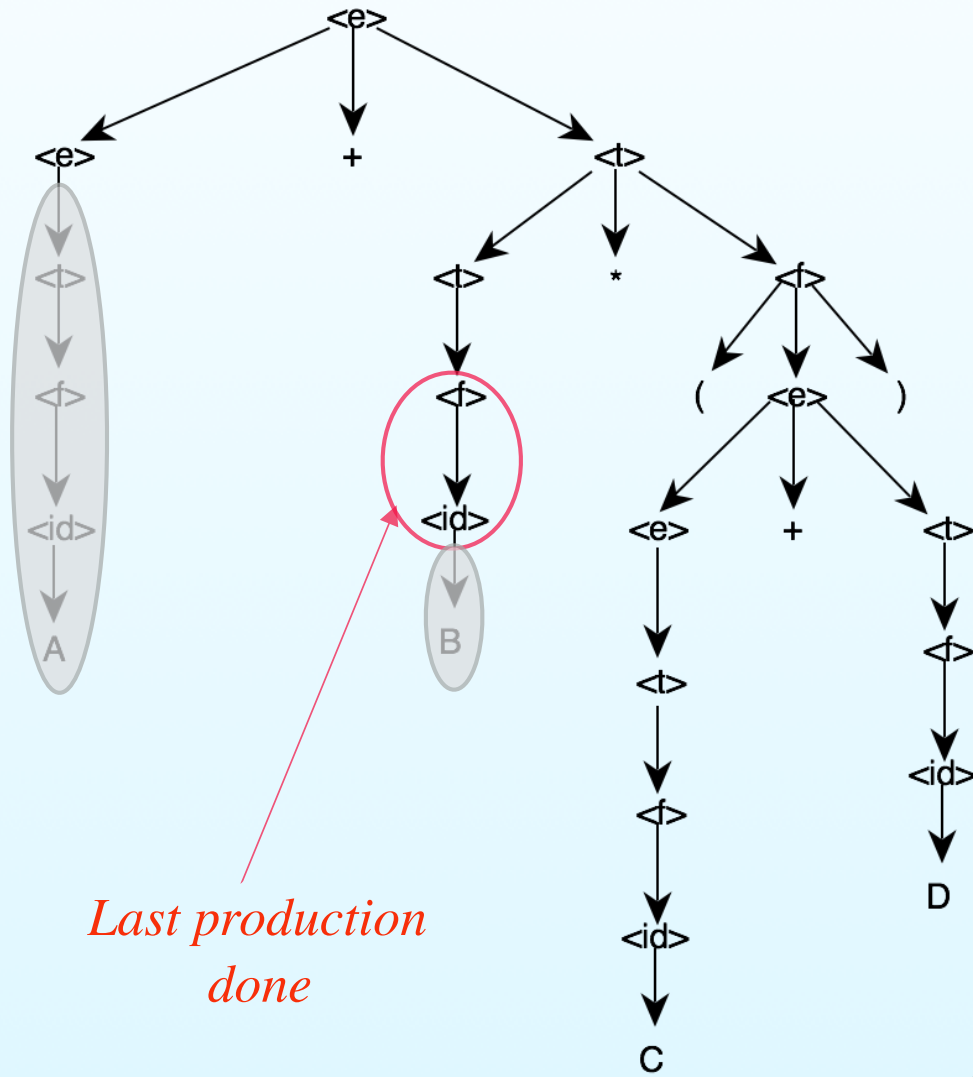
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



*Last production done*

# Finding the handle

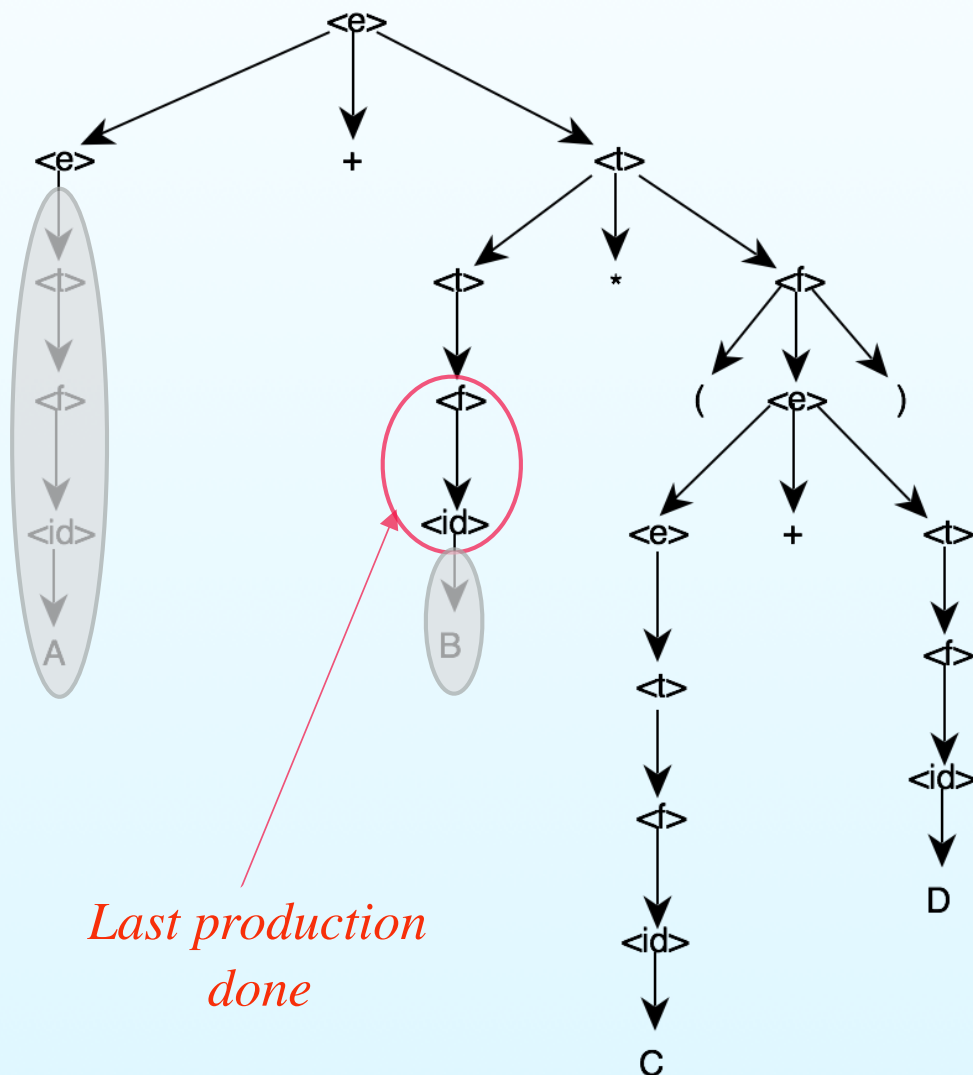
Parse: A + B \* (C + D)

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*

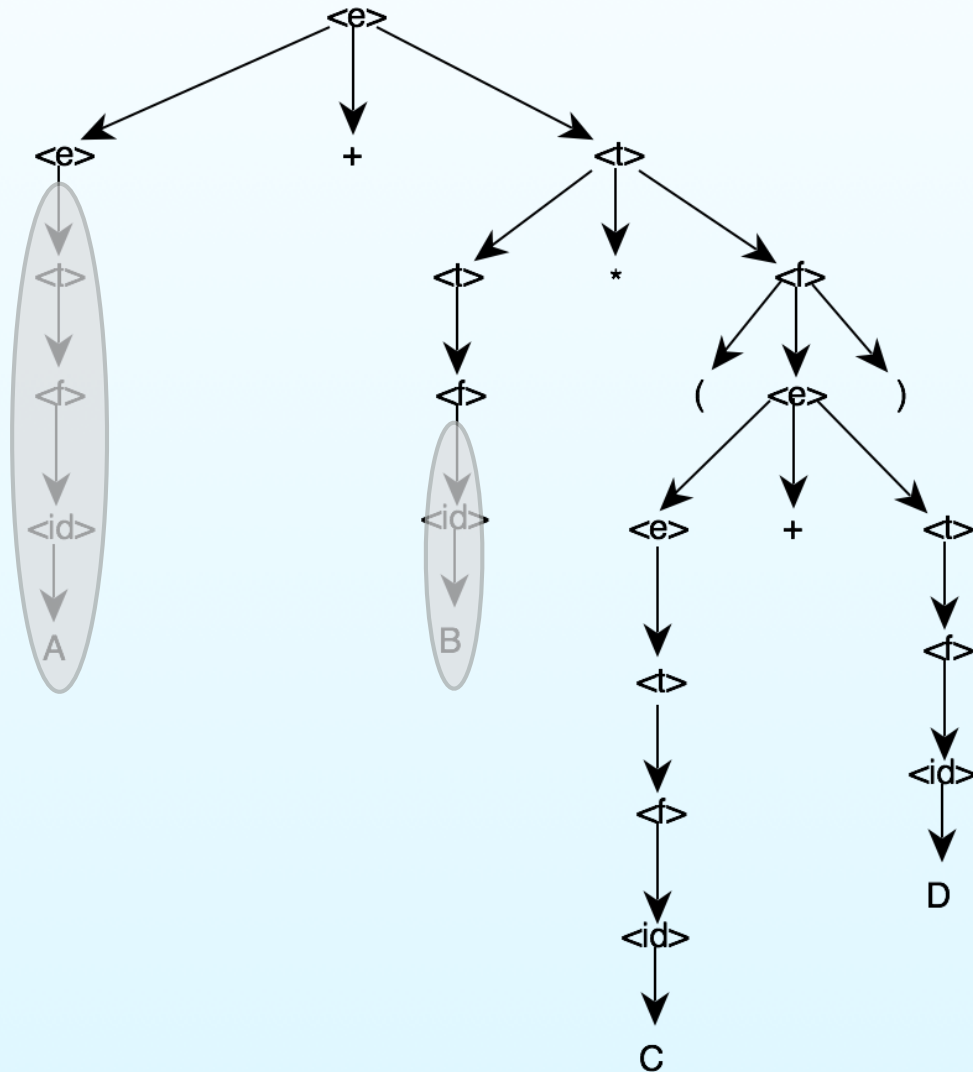


# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

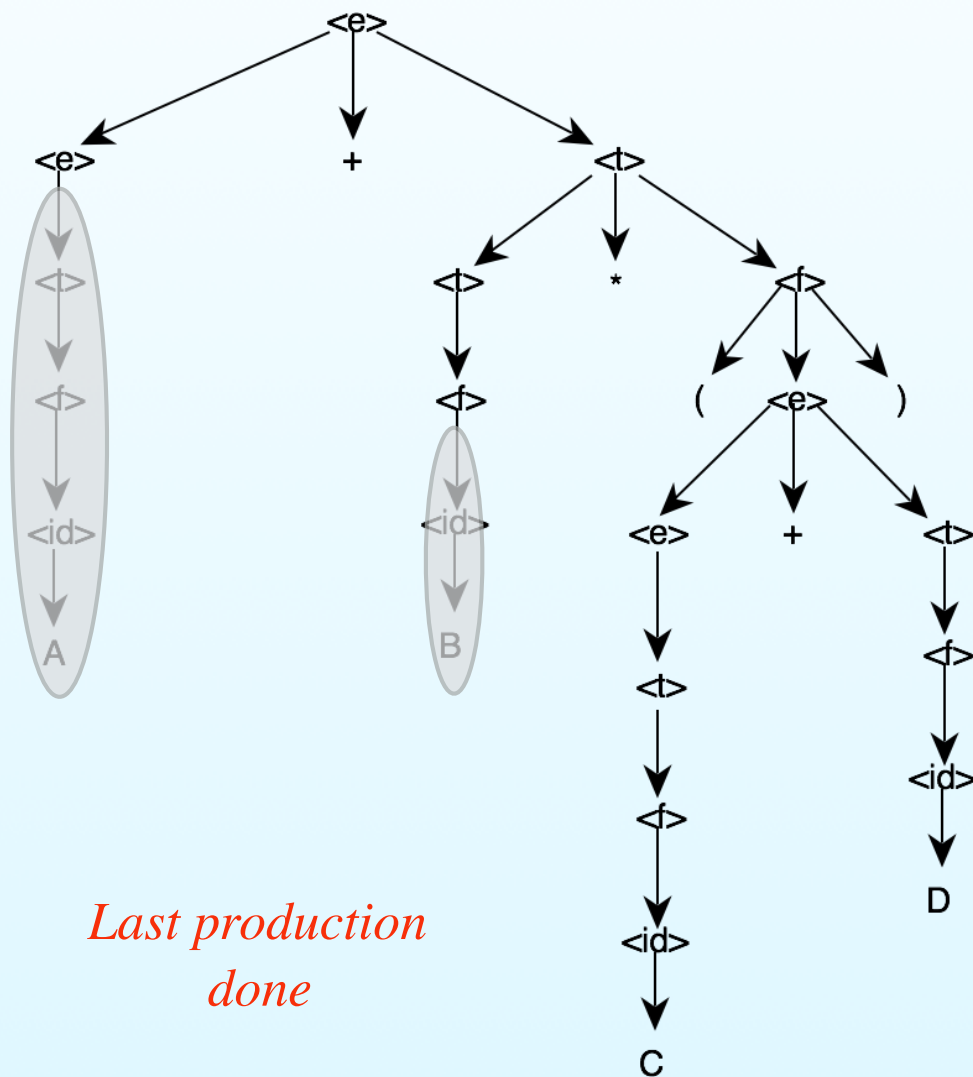


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



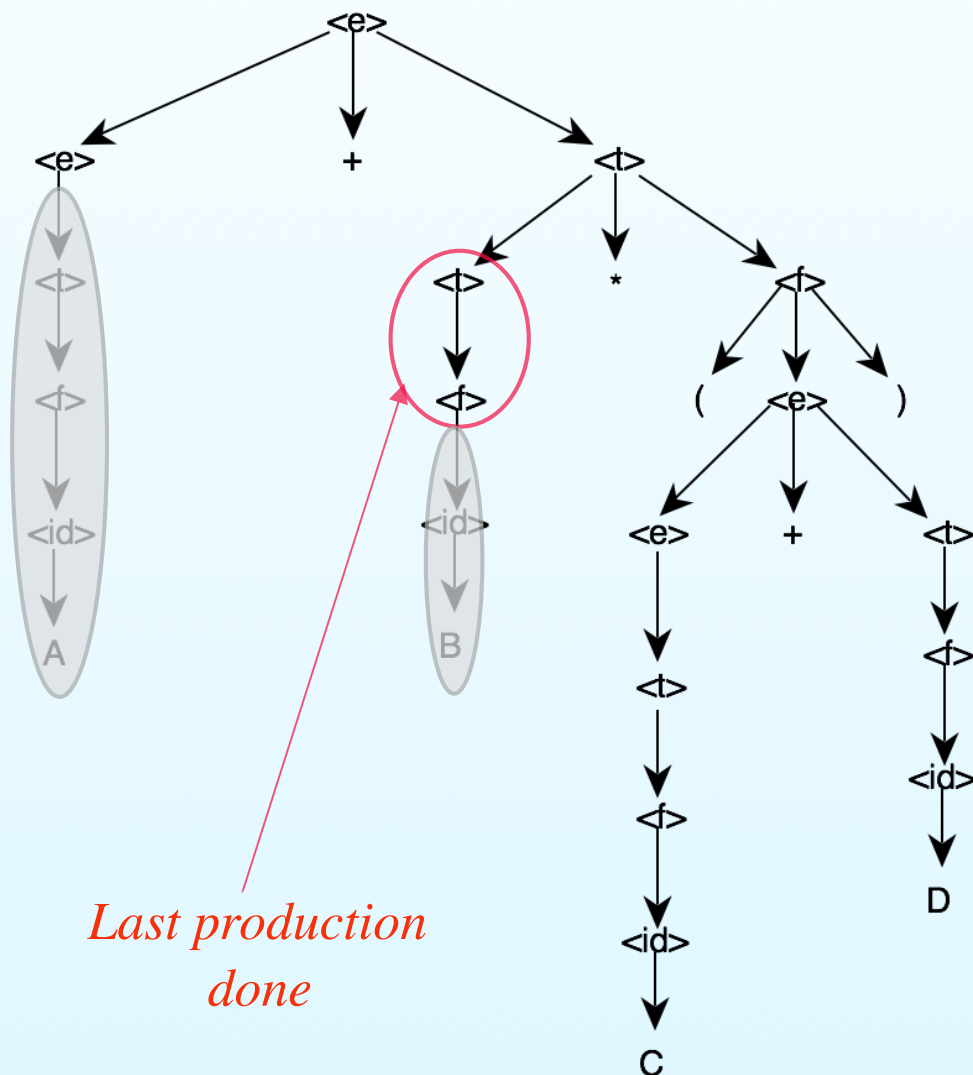
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Last production done*

# Finding the handle

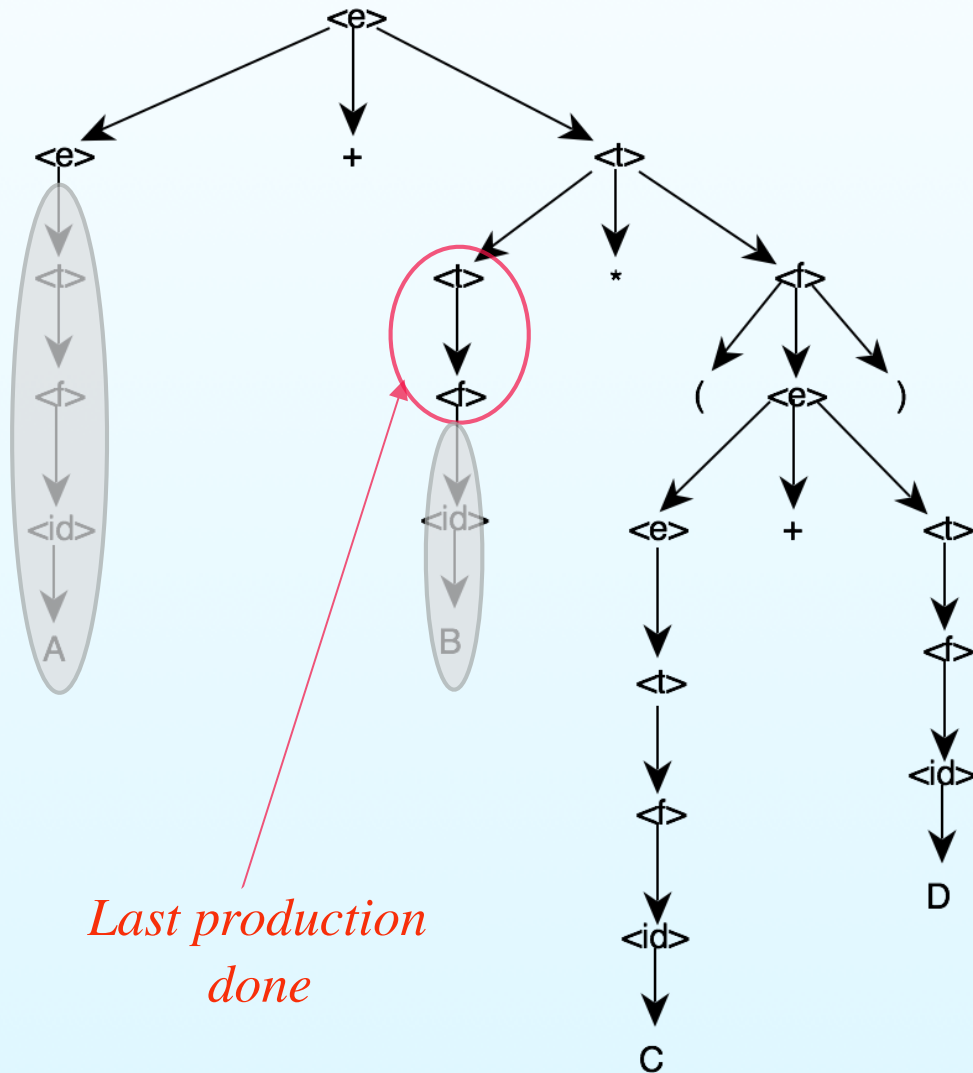
Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*



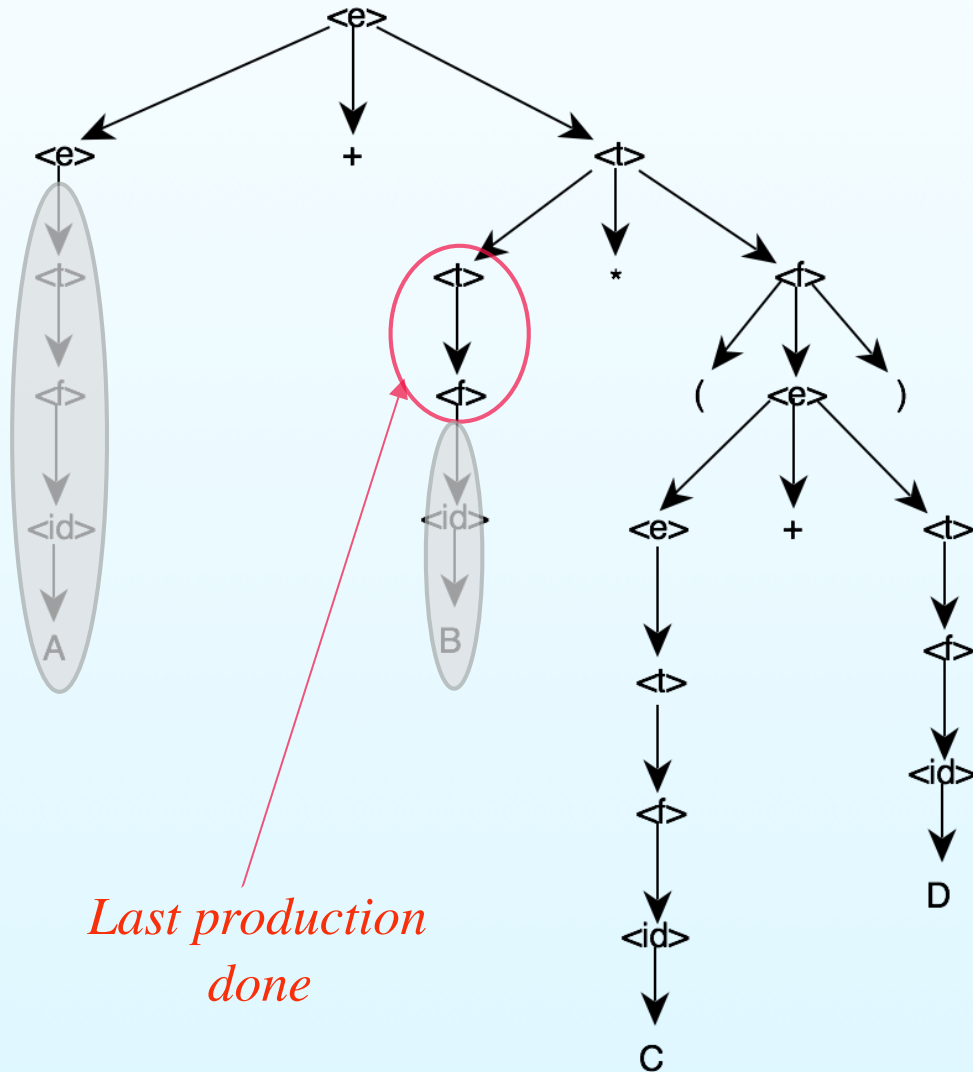
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



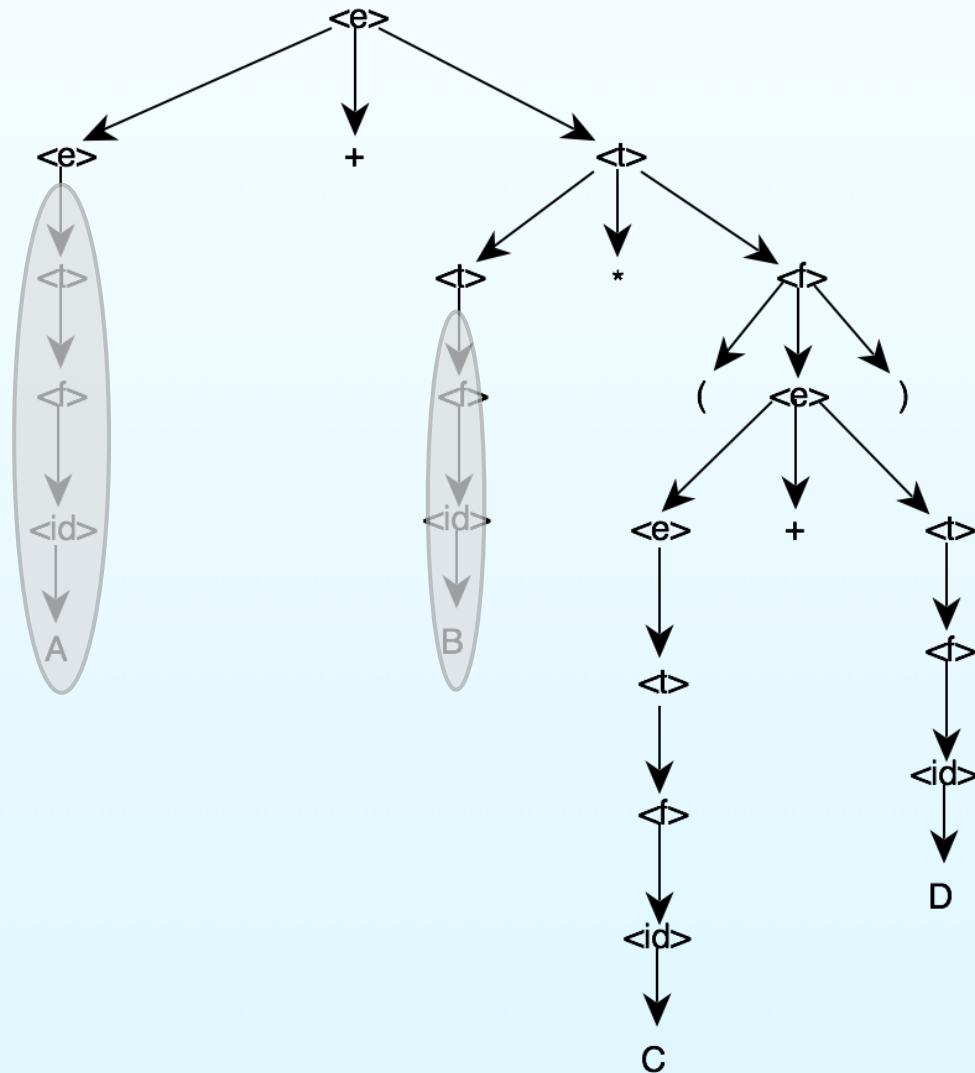
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---



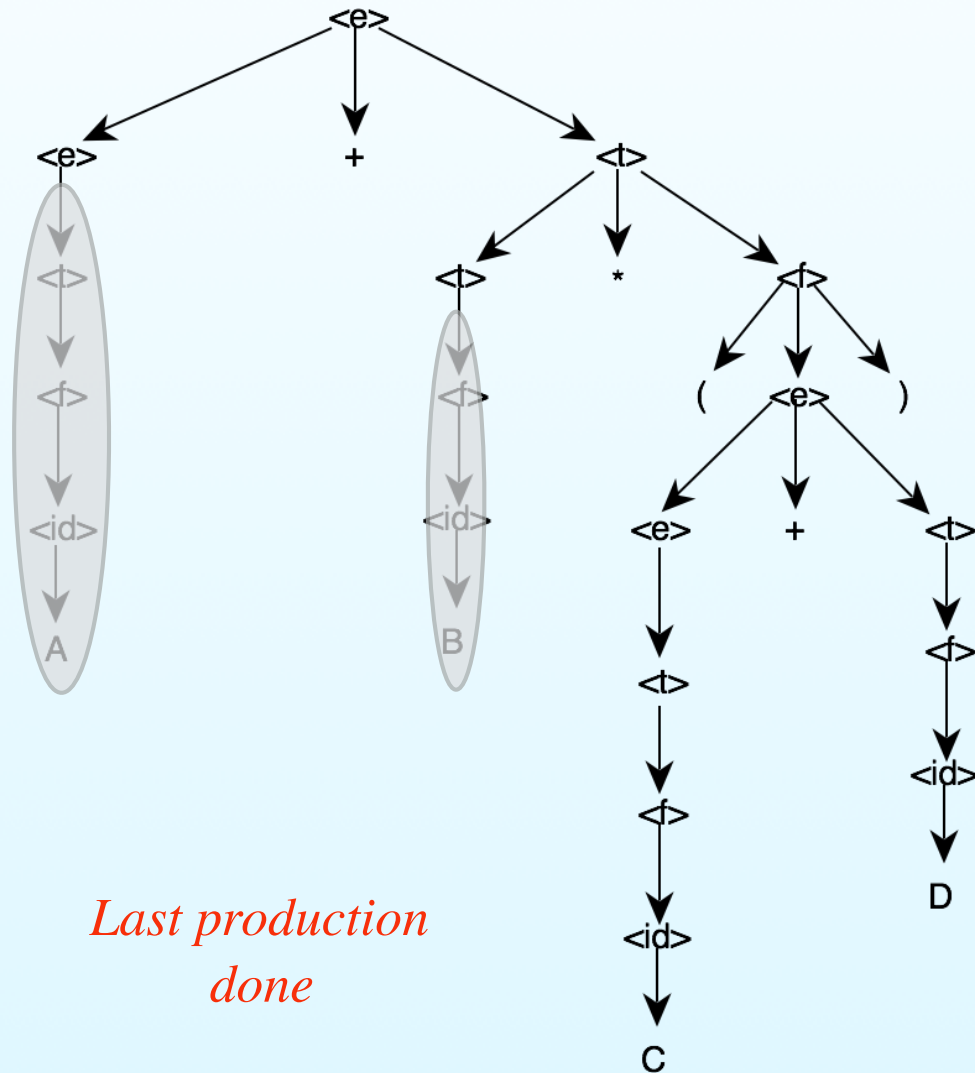


# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$   
 $\langle e \rangle + \langle t \rangle * (C + D)$   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---



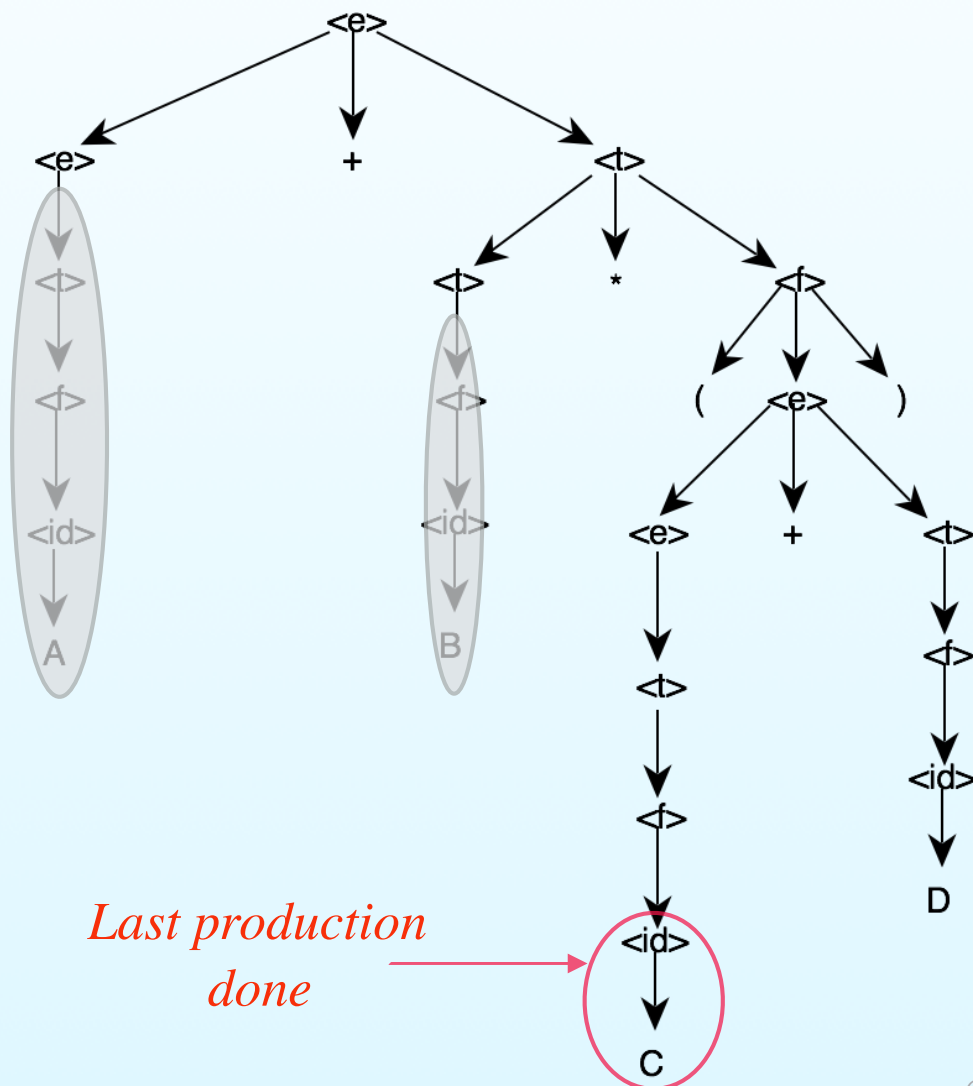
*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~



*Last production done*

# Finding the handle

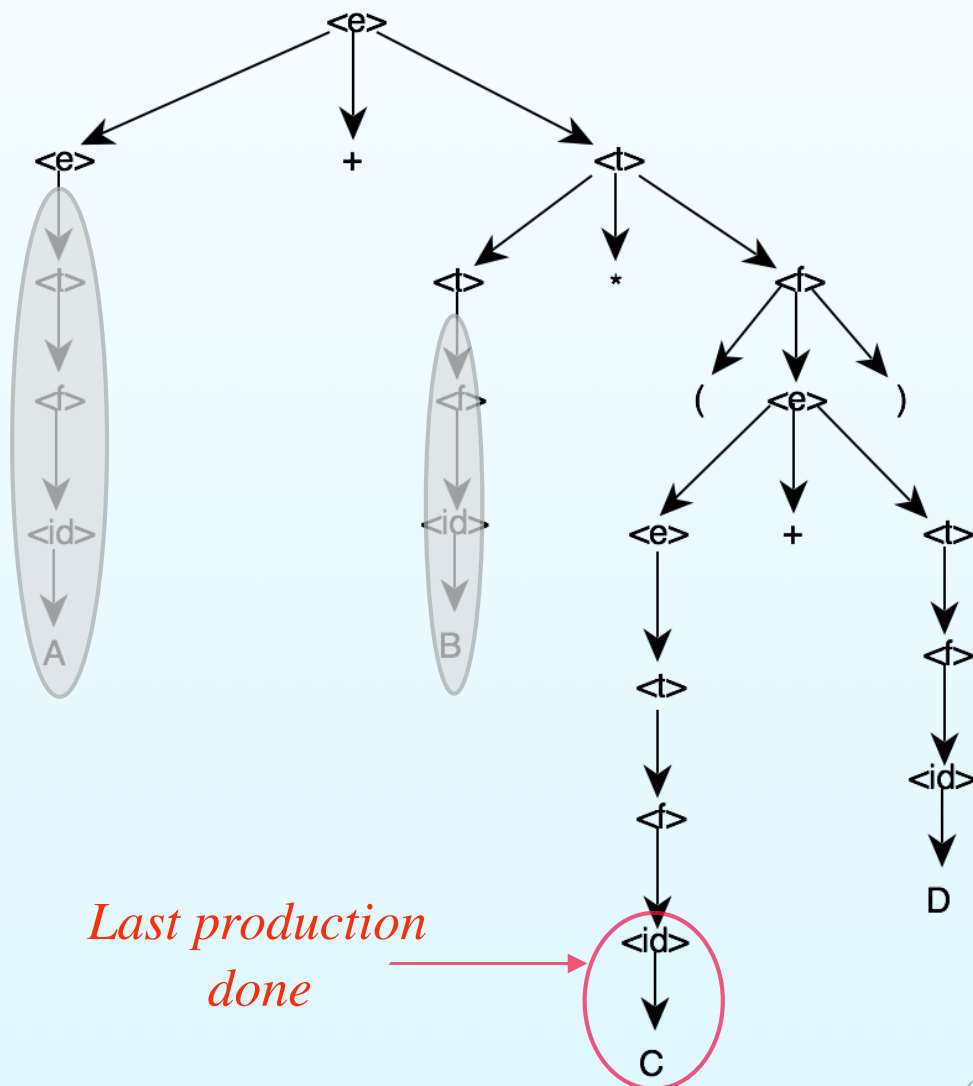
Parse: A + B \* (C + D)

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*



# Finding the handle

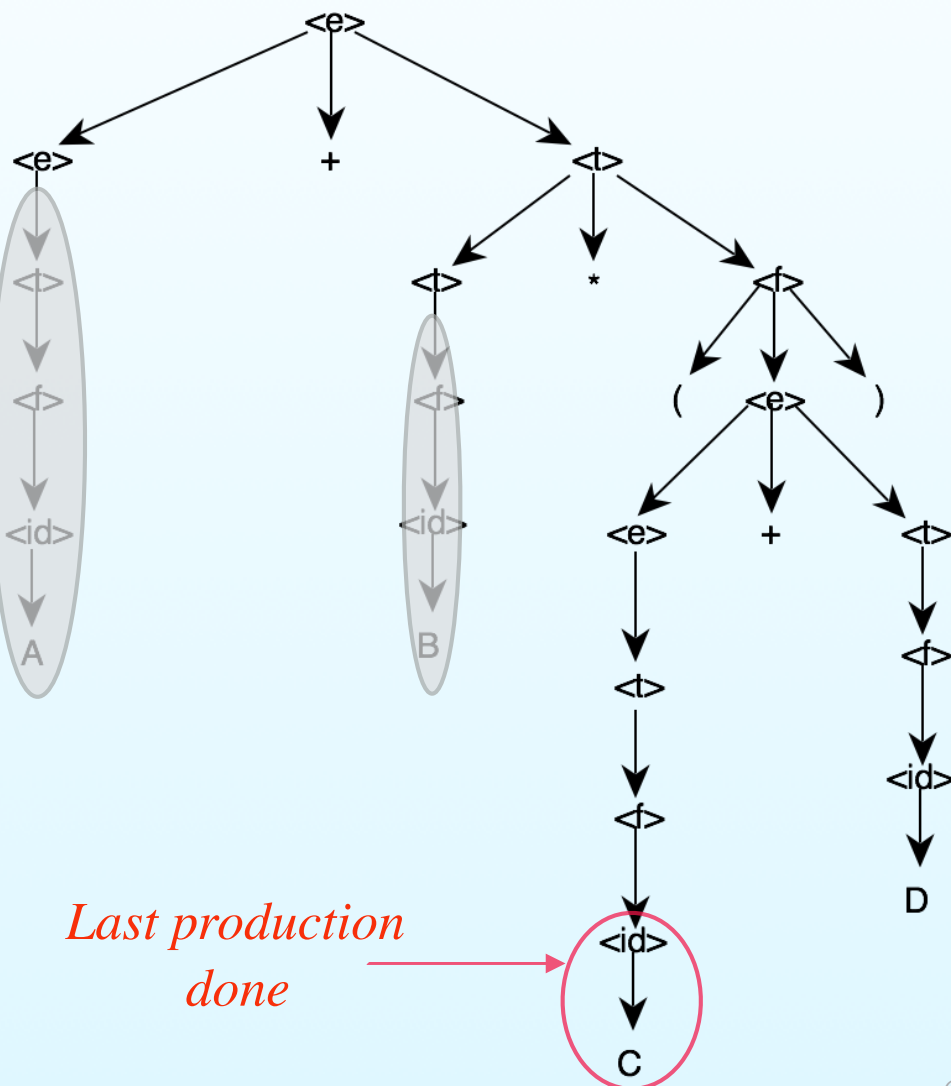
Parse:  $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\mathbf{C} + D)$
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

*Last production done*



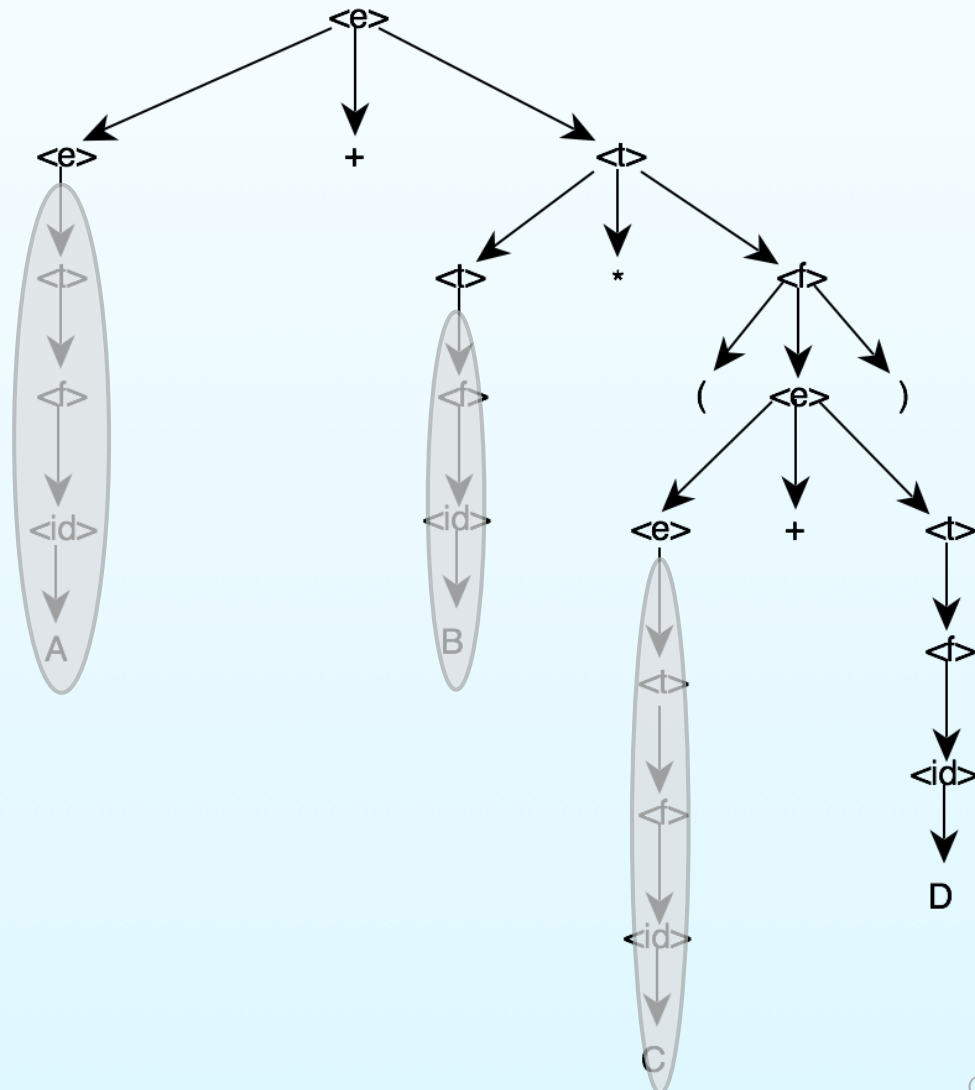
# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$   
 $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...



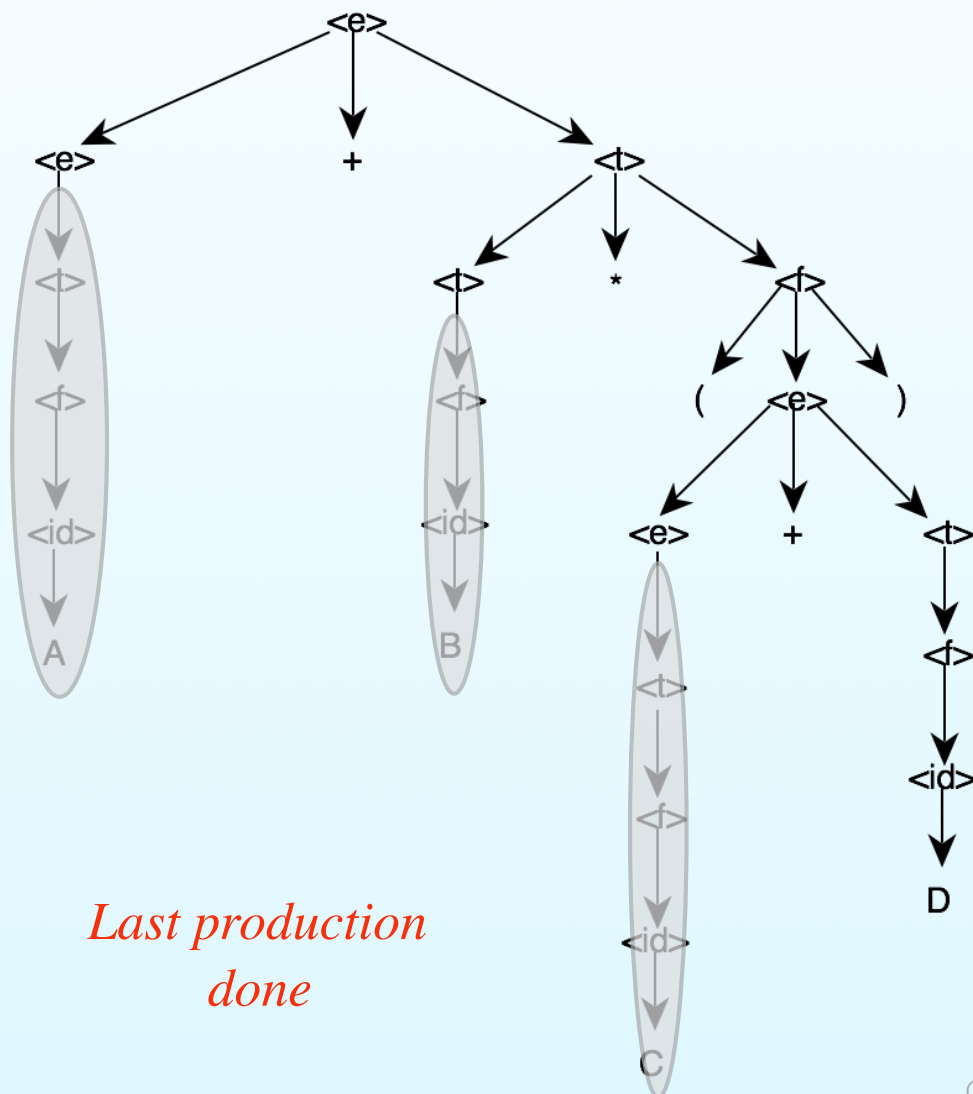
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



*Last production done*

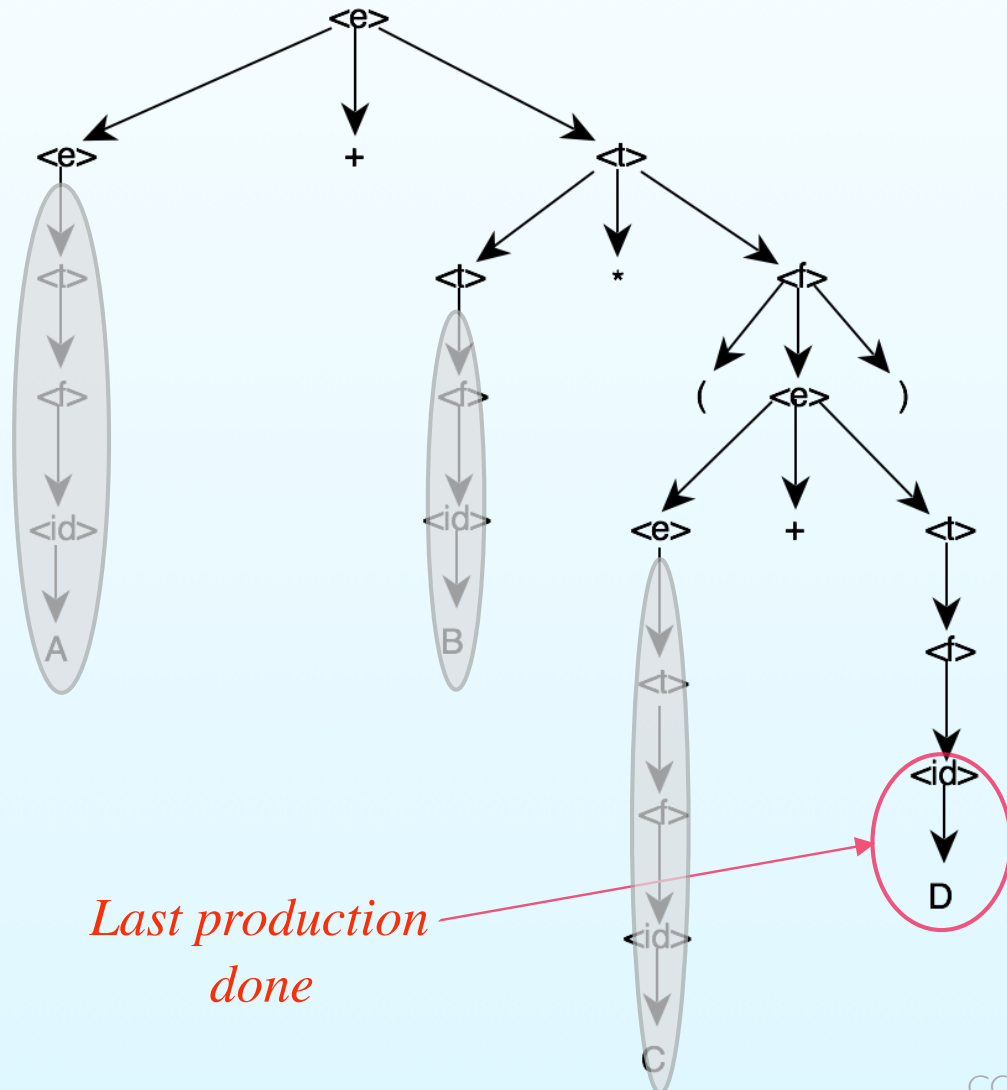
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



*Last production done*

# Finding the handle

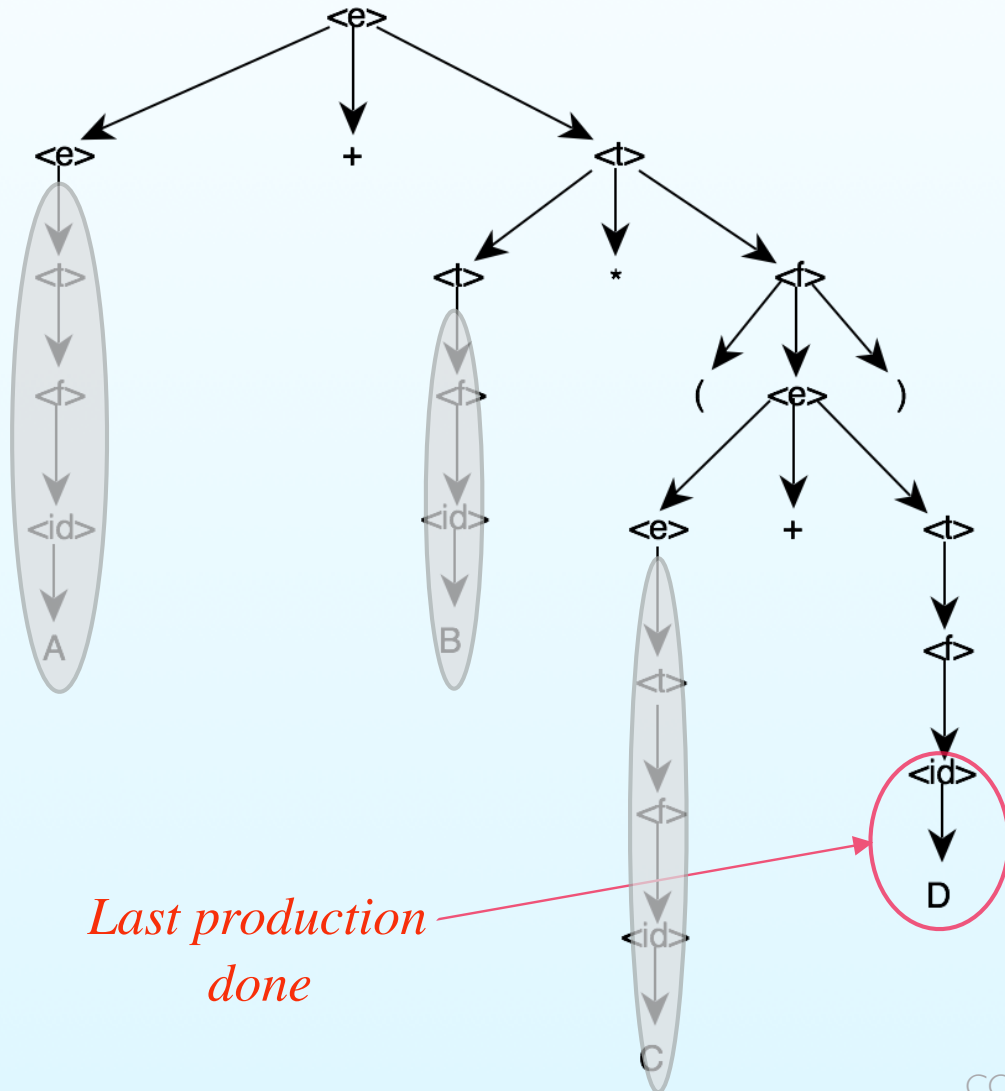
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



*Last production done*



# Finding the handle

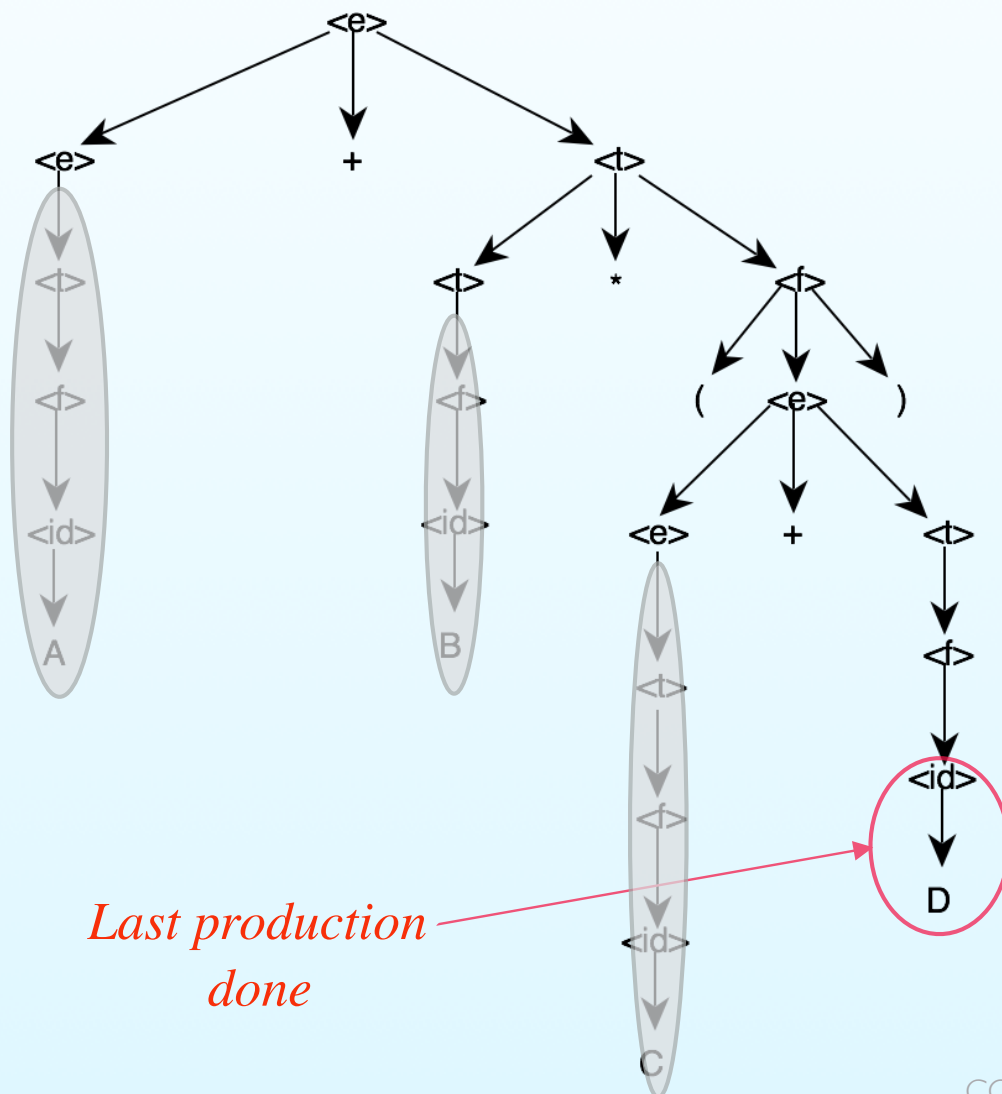
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



*Last production done*

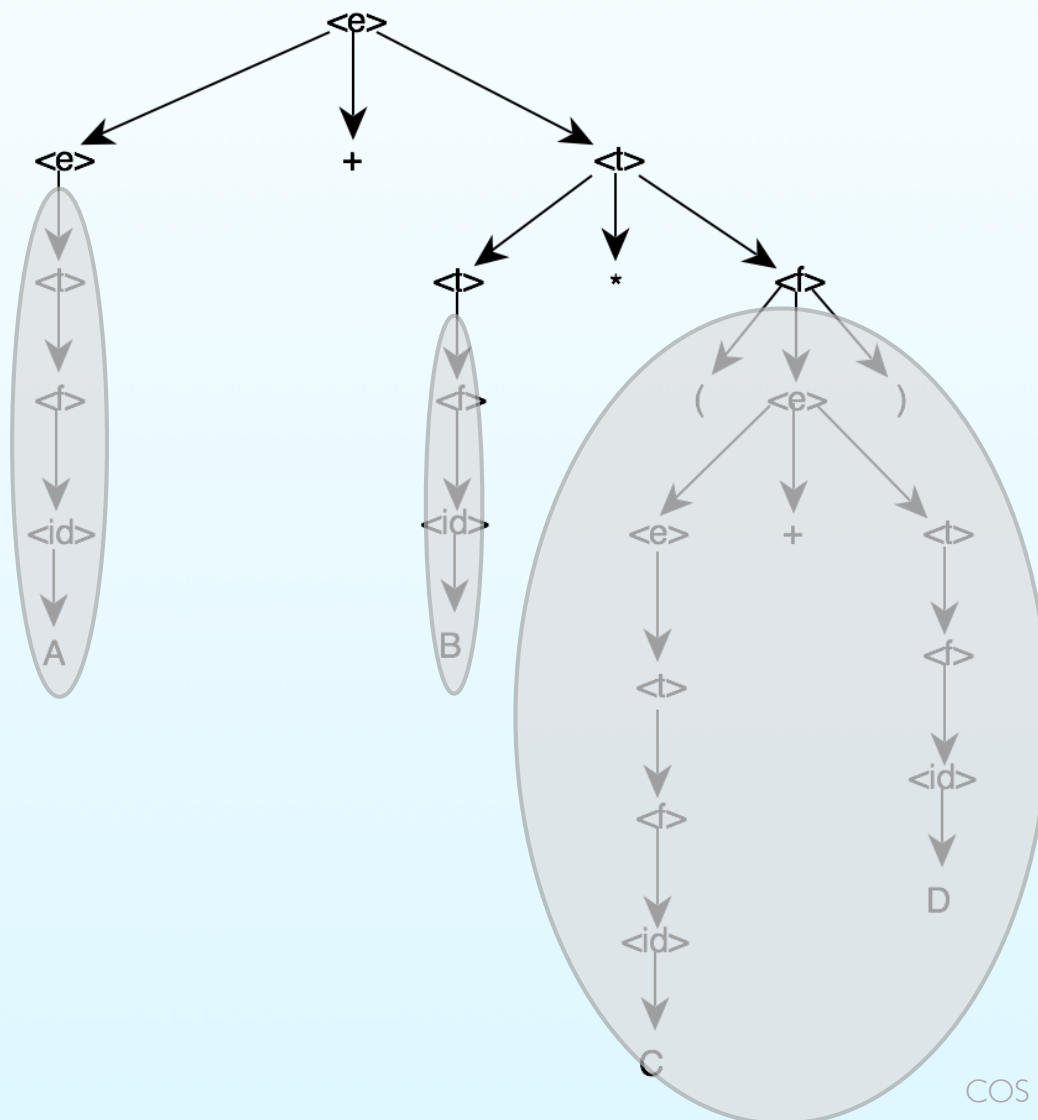
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



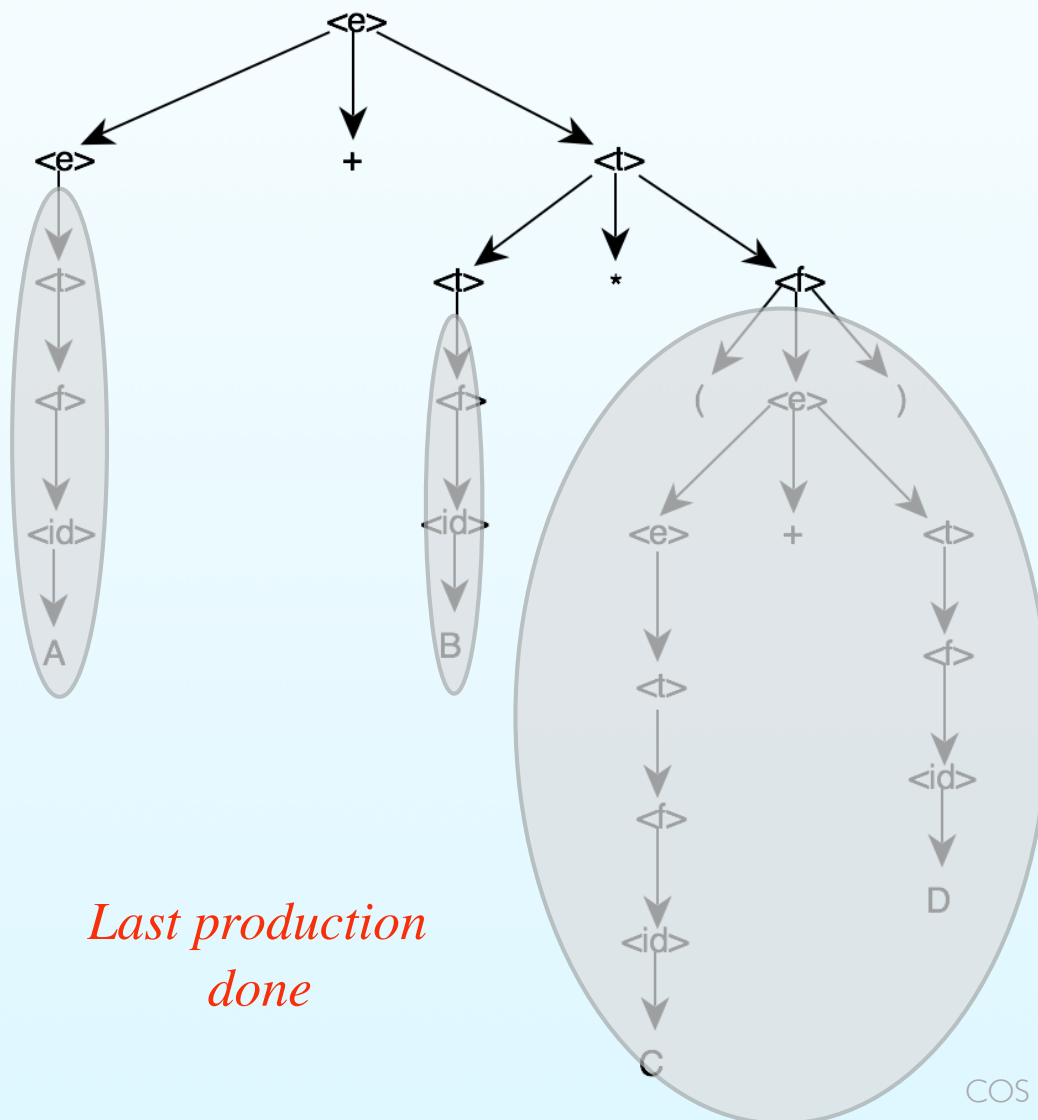
# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



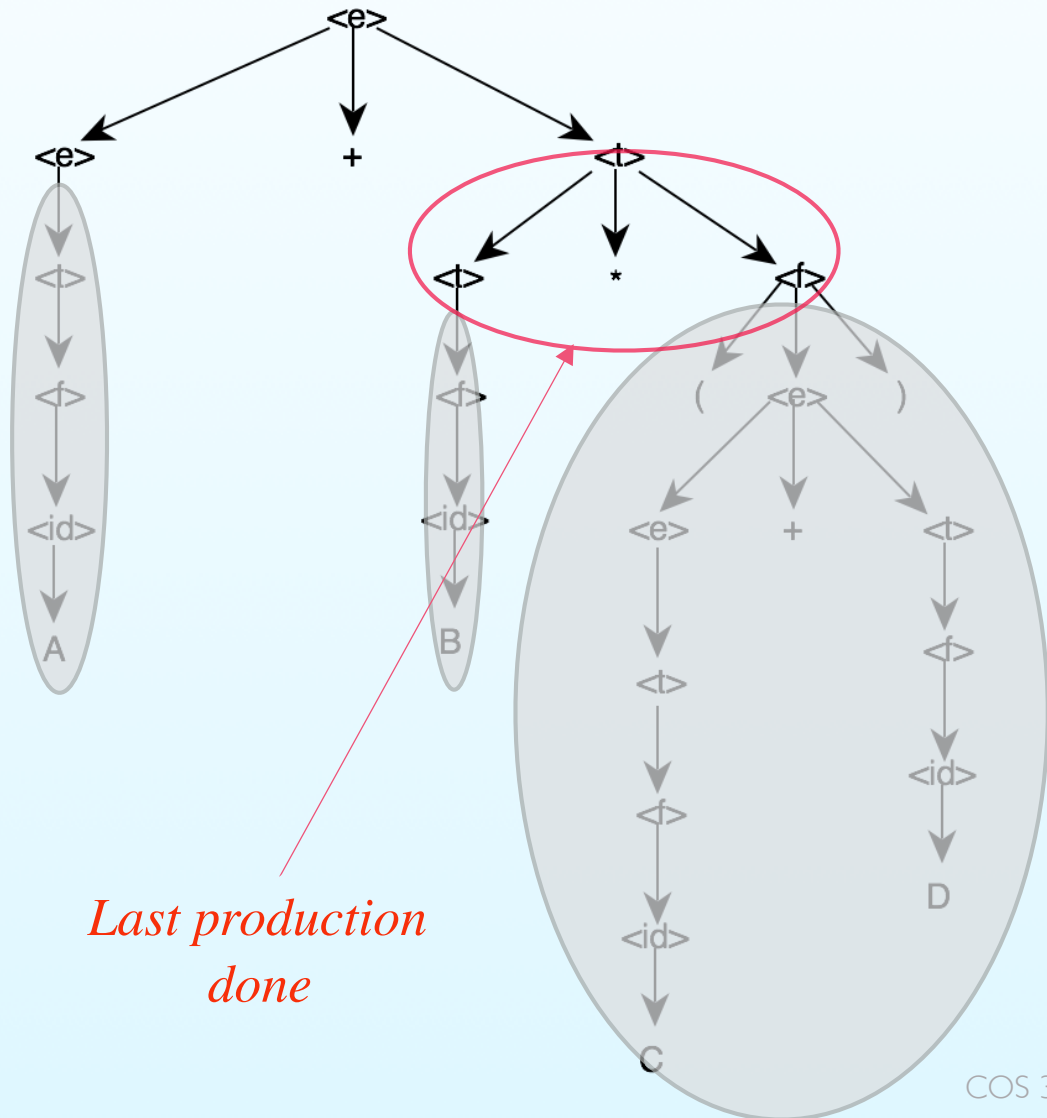
# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...



*Last production done*

# Finding the handle

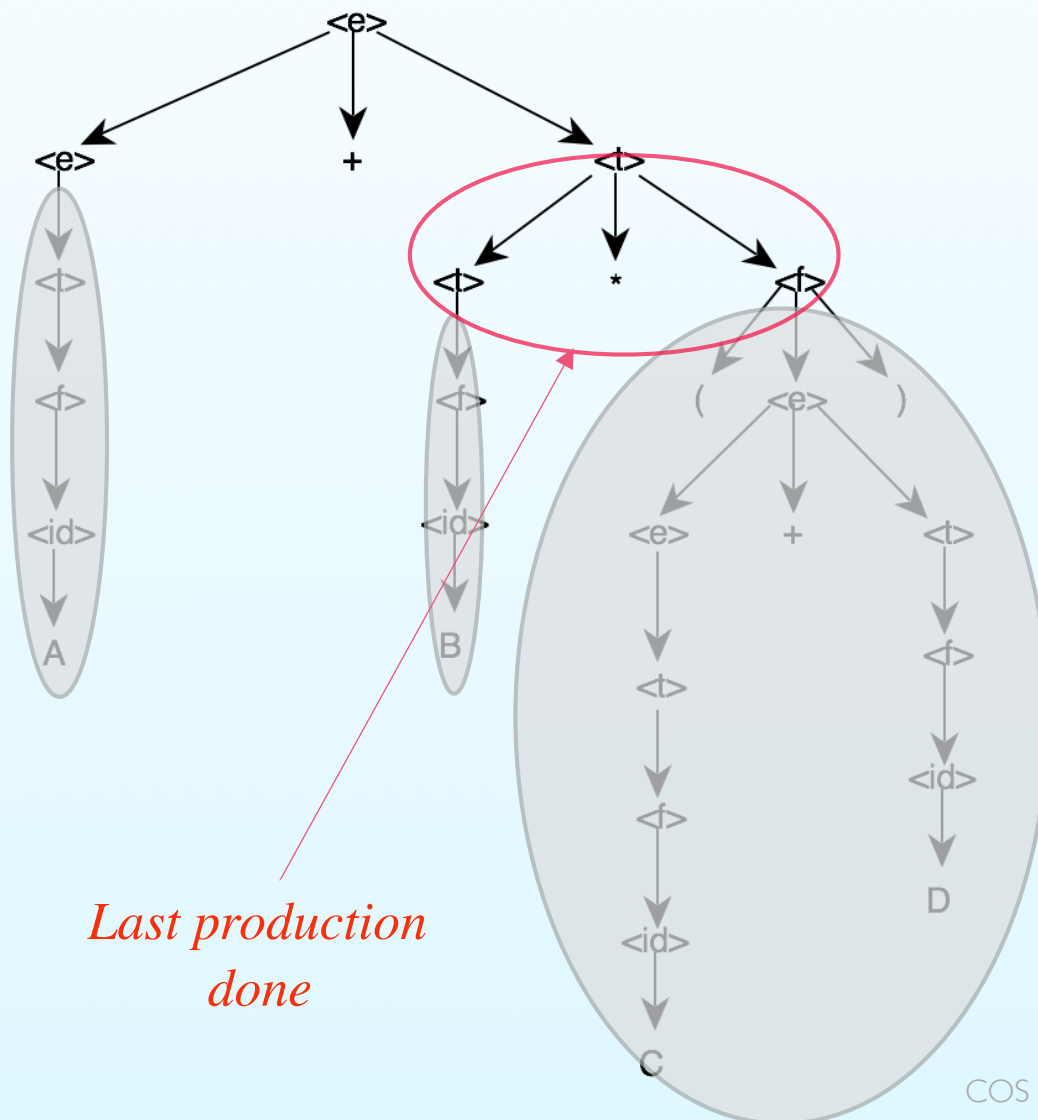
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



*Last production done*

# Finding the handle

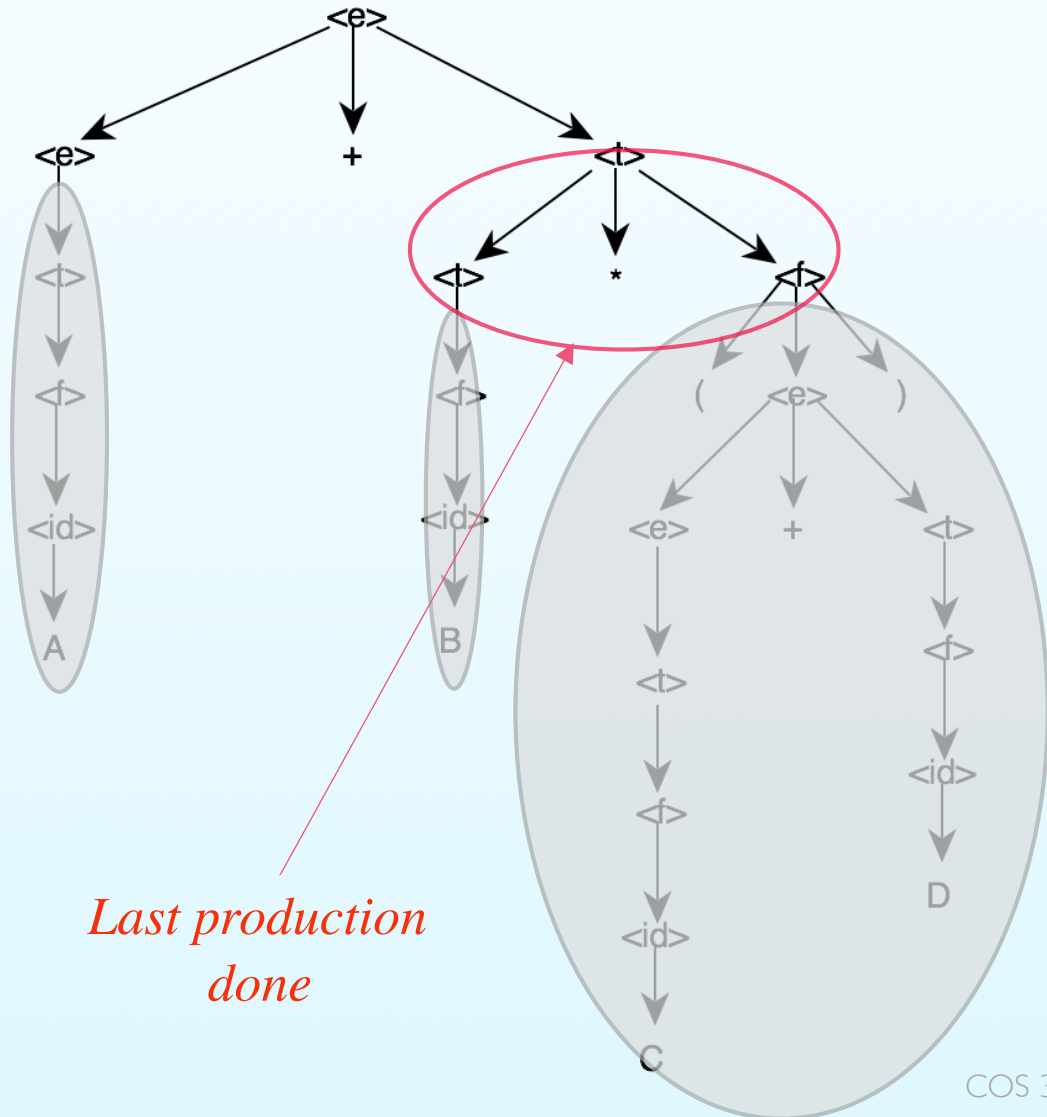
Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...



*Last production done*

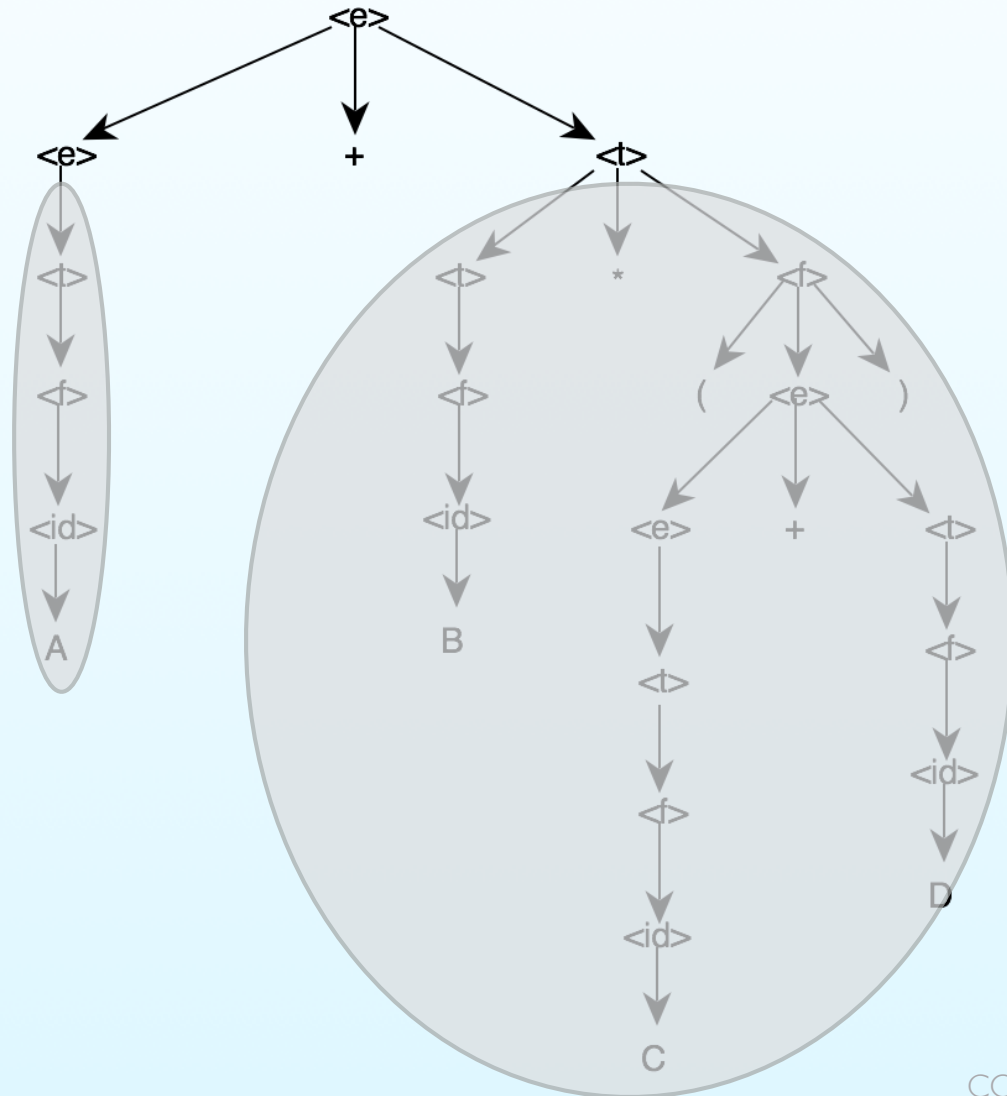
# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 ~~$\langle e \rangle + \langle t \rangle * \langle f \rangle$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...



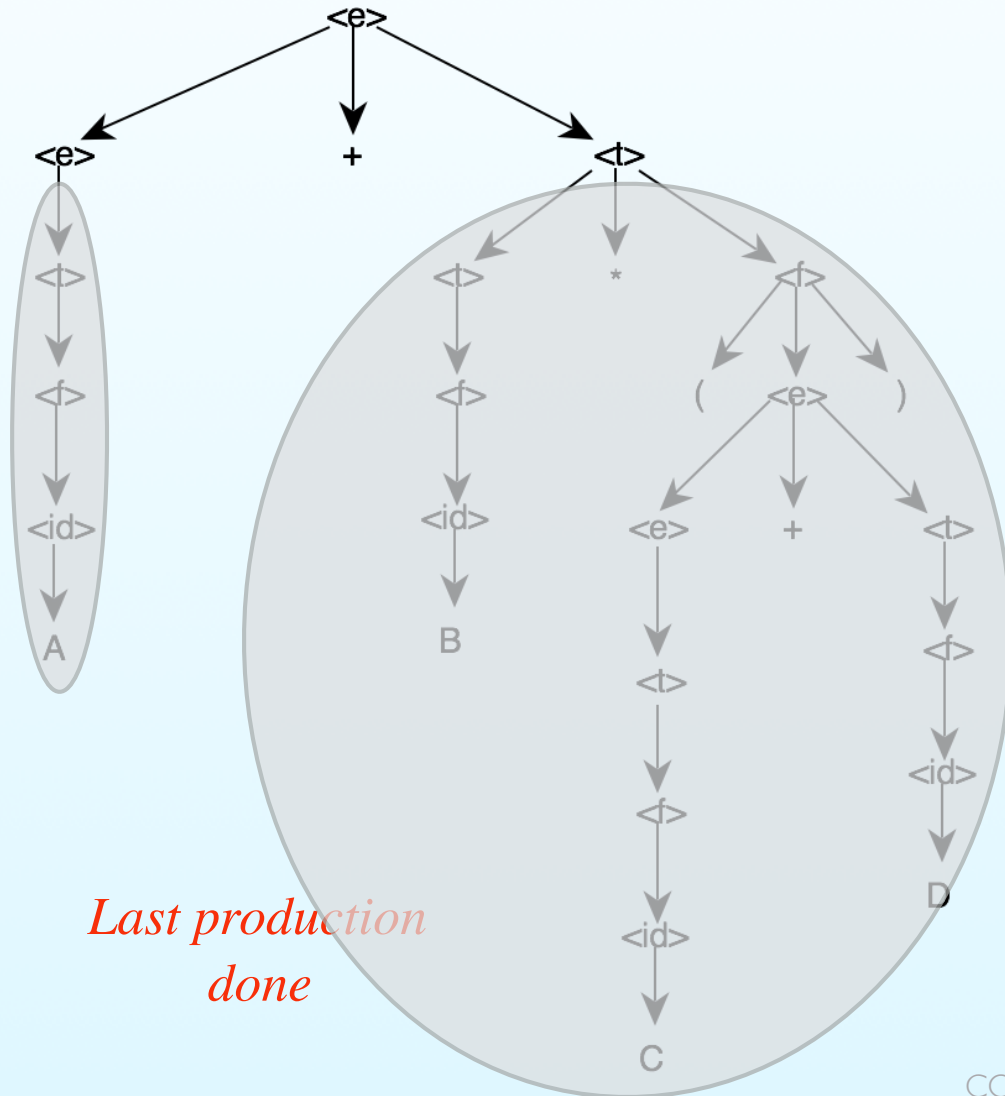
# Finding the handle

Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 ~~$\langle e \rangle + \langle t \rangle * \langle f \rangle$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...





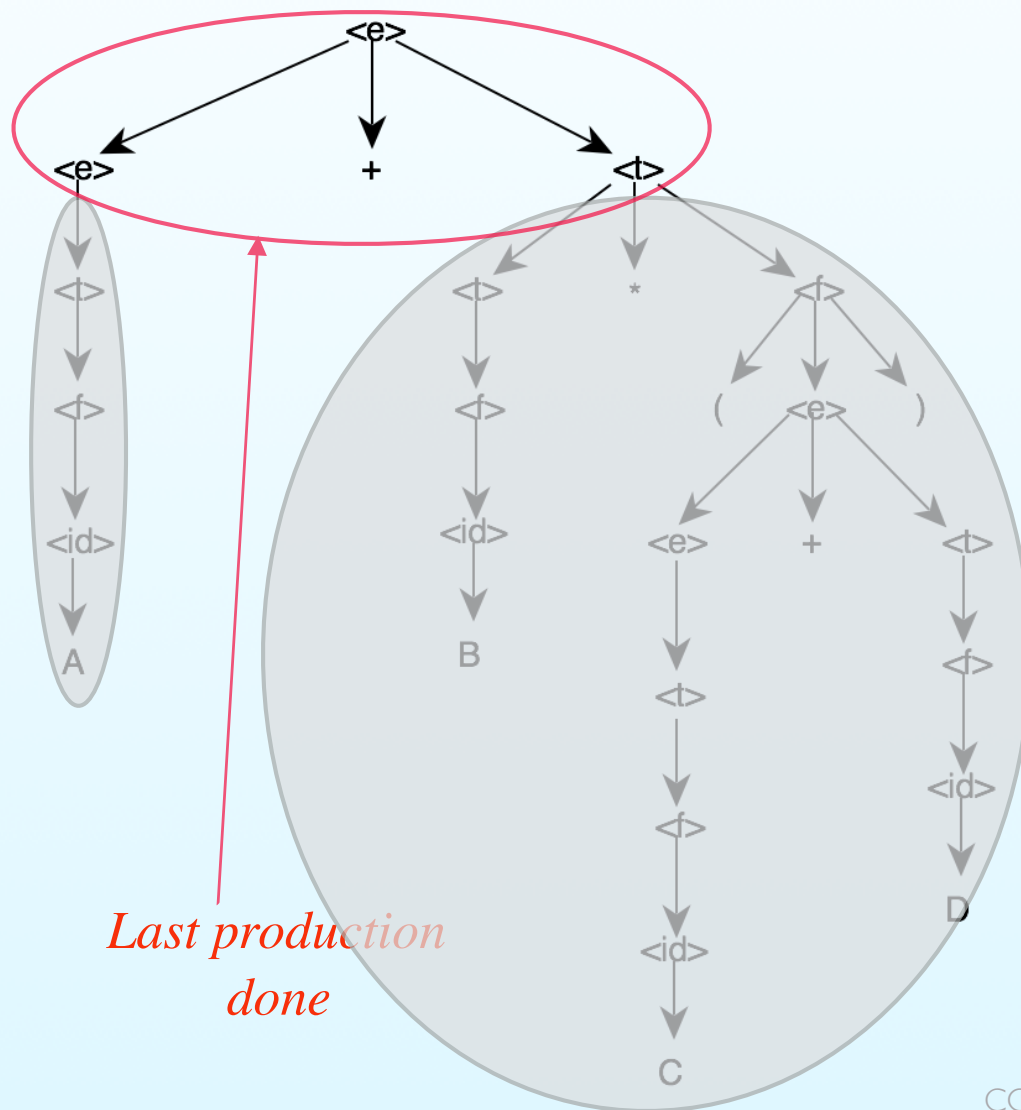
# Finding the handle

Parse: A + B \* (C + D)

~~<e>~~  
~~<e> + <t>~~  
~~<e> + <t> \* <f>~~  
~~<e> + <t> \* (<e>)~~  
~~<e> + <t> \* (<e> + <t>)~~  
~~<e> + <t> \* (<e> + <f>)~~  
~~<e> + <t> \* (<e> + <id>)~~  
~~<e> + <t> \* (<e> + D)~~  
~~<e> + <t> \* (<t> + D)~~  
~~<e> + <t> \* (<f> + D)~~  
~~<e> + <t> \* (<id> + D)~~  
~~<e> + <t> \* (C + D)~~  
~~<e> + <f> \* (C + D)~~  
~~<e> + <id> \* (C + D)~~  
~~<e> + B \* (C + D)~~  
~~<t> + B \* (C + D)~~  
~~<f> + B \* (C + D)~~  
~~<id> + B \* (C + D)~~  
 A + B \* (C + D)

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...



*Last production done*

# Finding the handle

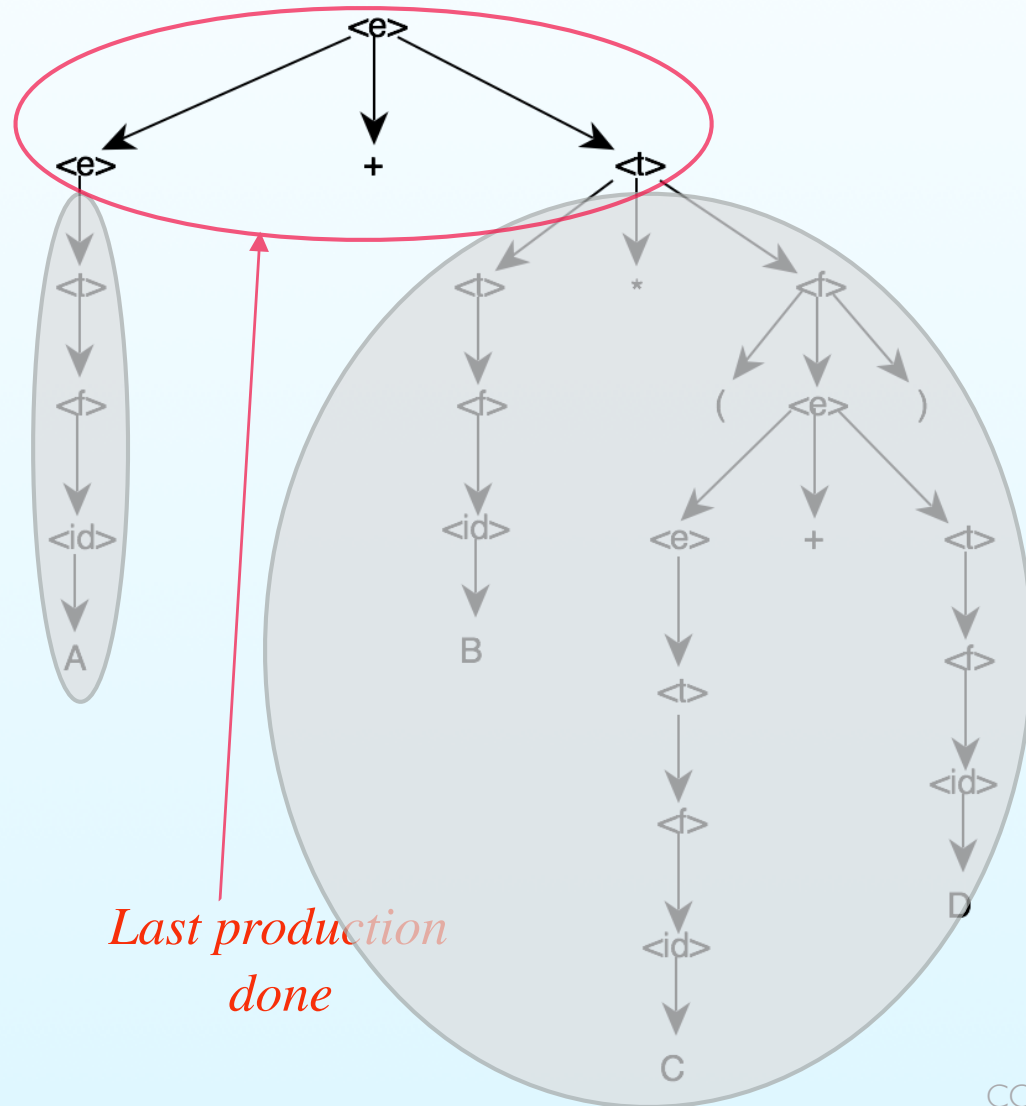
Parse: A + B \* (C + D)

~~<e>~~  
~~<e> + <t>~~  
~~<e> + <t> \* <f>~~  
~~<e> + <t> \* (<e>)~~  
~~<e> + <t> \* (<e> + <t>)~~  
~~<e> + <t> \* (<e> + <f>)~~  
~~<e> + <t> \* (<e> + <id>)~~  
~~<e> + <t> \* (<e> + D)~~  
~~<e> + <t> \* (<t> + D)~~  
~~<e> + <t> \* (<f> + D)~~  
~~<e> + <t> \* (<id> + D)~~  
~~<e> + <t> \* (C + D)~~  
~~<e> + <f> \* (C + D)~~  
~~<e> + <id> \* (C + D)~~  
~~<e> + B \* (C + D)~~  
~~<t> + B \* (C + D)~~  
~~<f> + B \* (C + D)~~  
~~<id> + B \* (C + D)~~  
**A + B \* (C + D)**

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Skip a few steps...



*Last production done*

# Finding the handle

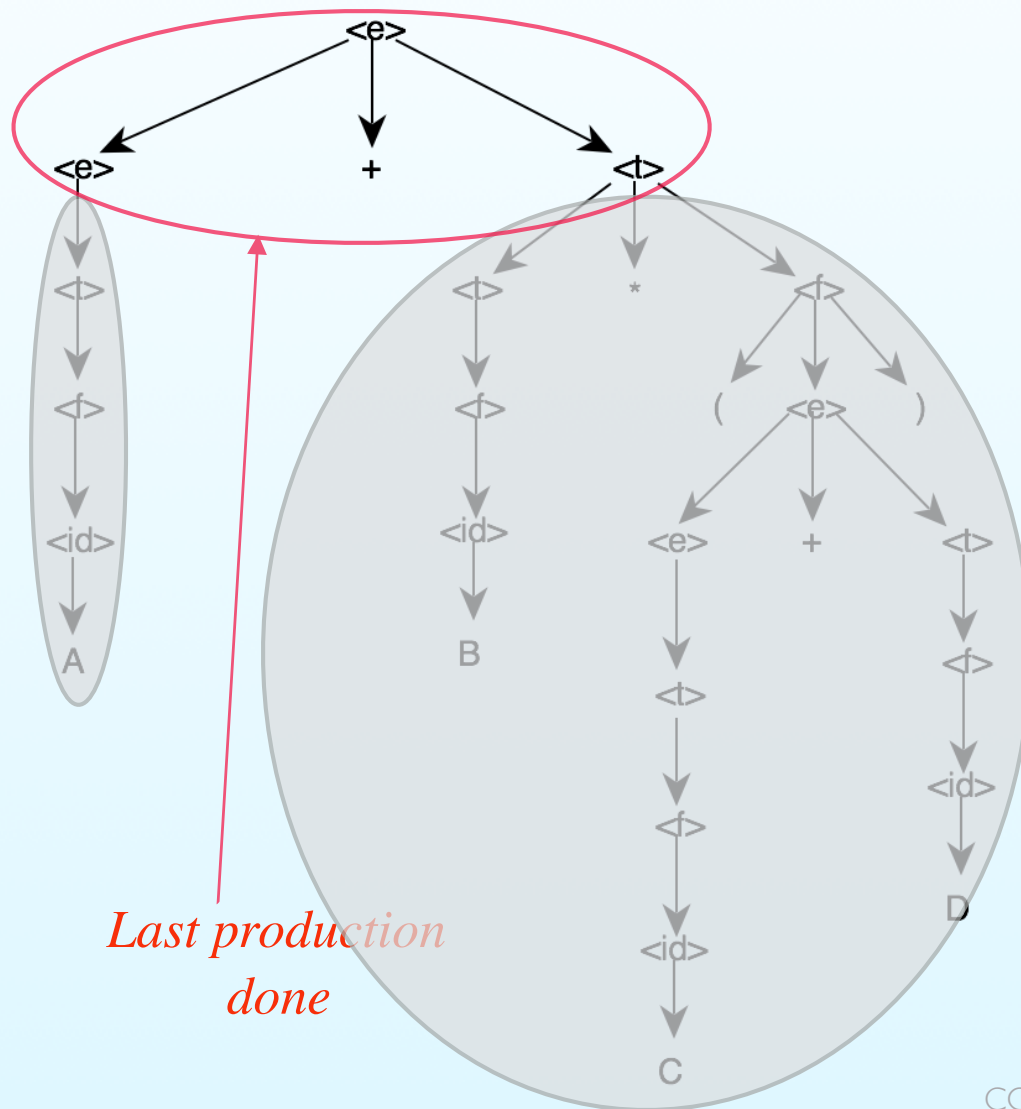
Parse: A + B \* (C + D)

- <e>
- <e> + <t>**
- ~~<e> + <t> \* <f>~~
- ~~<e> + <t> \* (<e>)~~
- ~~<e> + <t> \* (<e> + <t>)~~
- ~~<e> + <t> \* (<e> + <f>)~~
- ~~<e> + <t> \* (<e> + <id>)~~
- ~~<e> + <t> \* (<e> + D)~~
- ~~<e> + <t> \* (<t> + D)~~
- ~~<e> + <t> \* (<f> + D)~~
- ~~<e> + <t> \* (<id> + D)~~
- ~~<e> + <t> \* (C + D)~~
- ~~<e> + <f> \* (C + D)~~
- ~~<e> + <id> \* (C + D)~~
- ~~<e> + B \* (C + D)~~
- ~~<t> + B \* (C + D)~~
- ~~<f> + B \* (C + D)~~
- ~~<id> + B \* (C + D)~~
- A + B \* (C + D)

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Skip a few steps...



*Last production done*

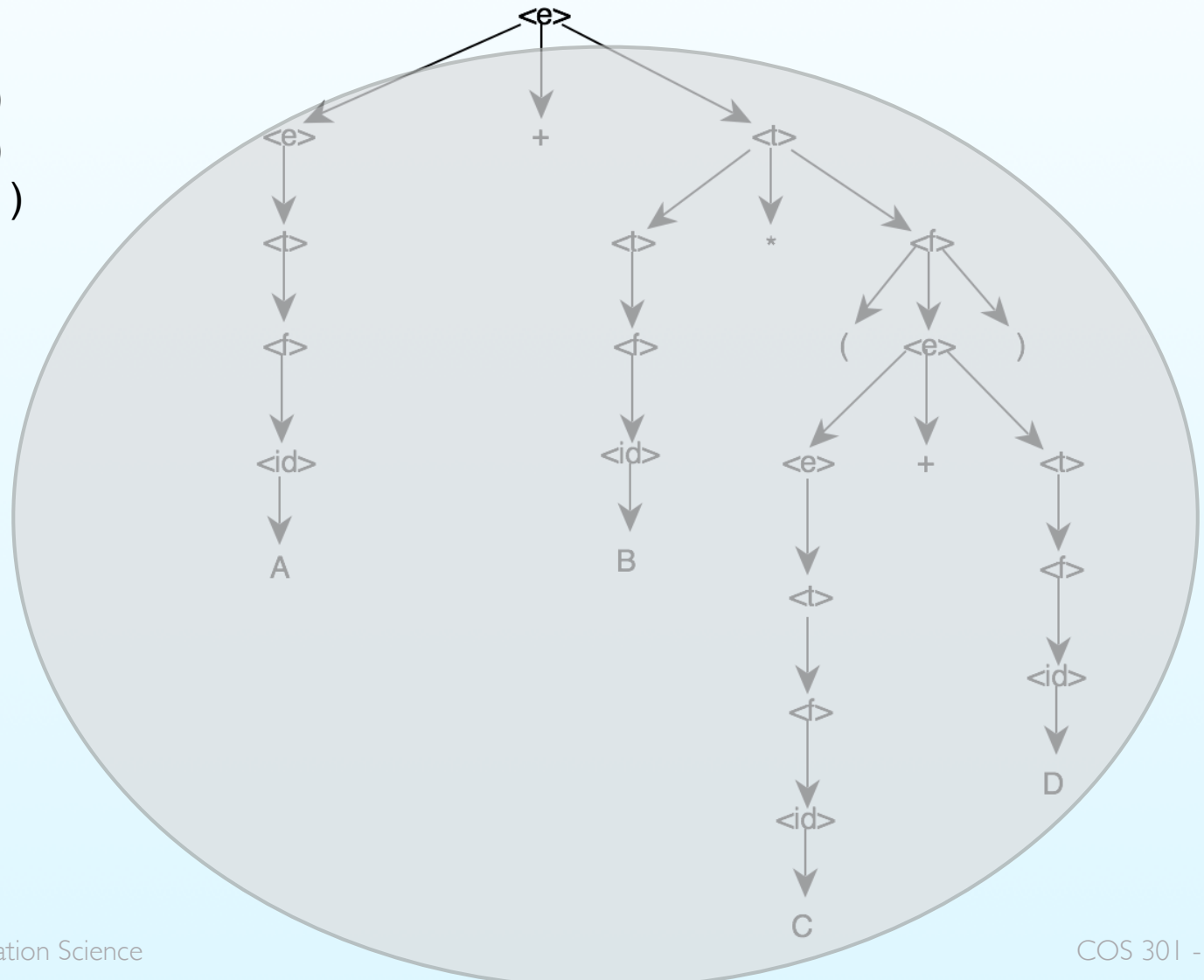
# Finding the handle

Parse:  $A + B * (C + D)$

~~$\langle e \rangle$~~   
 ~~$\langle e \rangle + \langle t \rangle$~~   
 ~~$\langle e \rangle + \langle t \rangle * \langle f \rangle$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Done



# Finding the handle

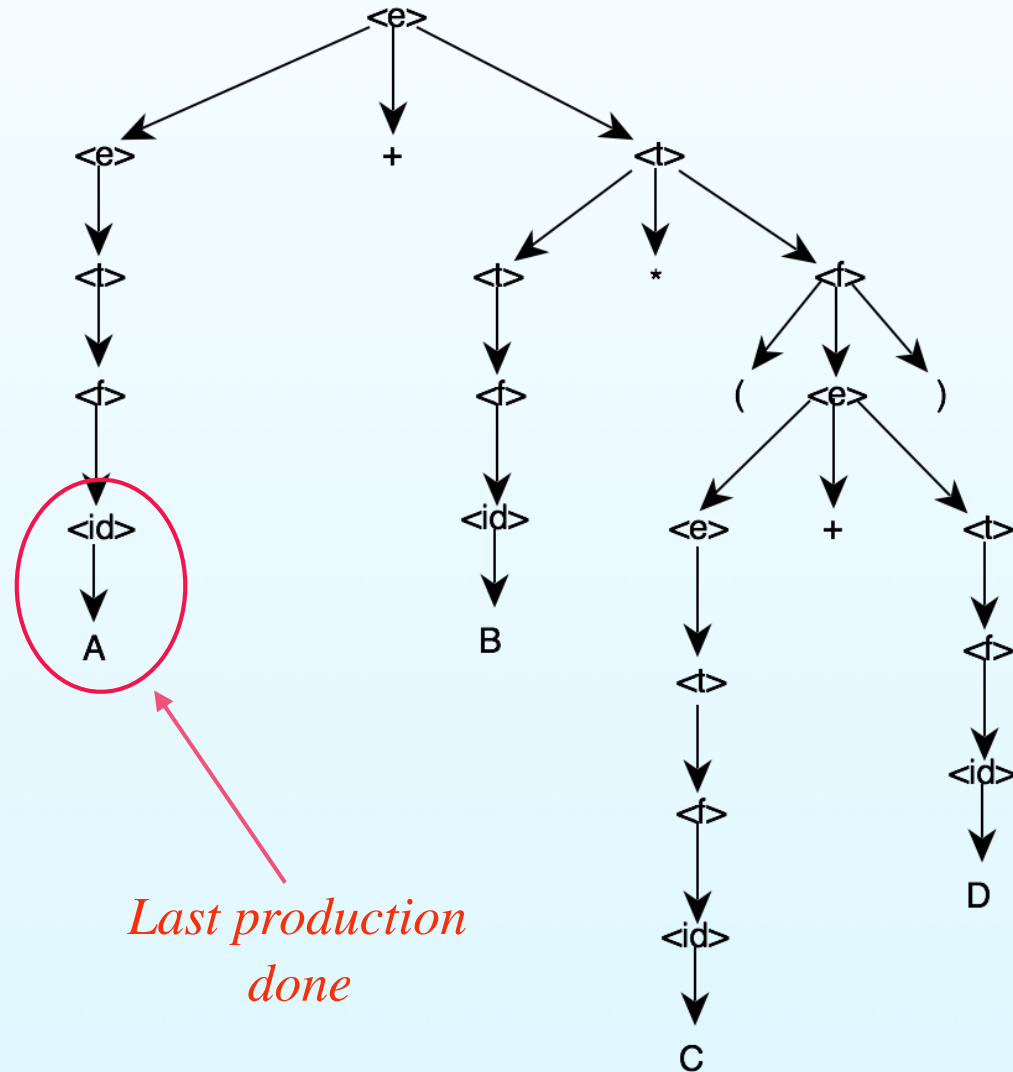
- What was common to all handles?

# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



*Handle*

*Last production done*

# Finding the handle

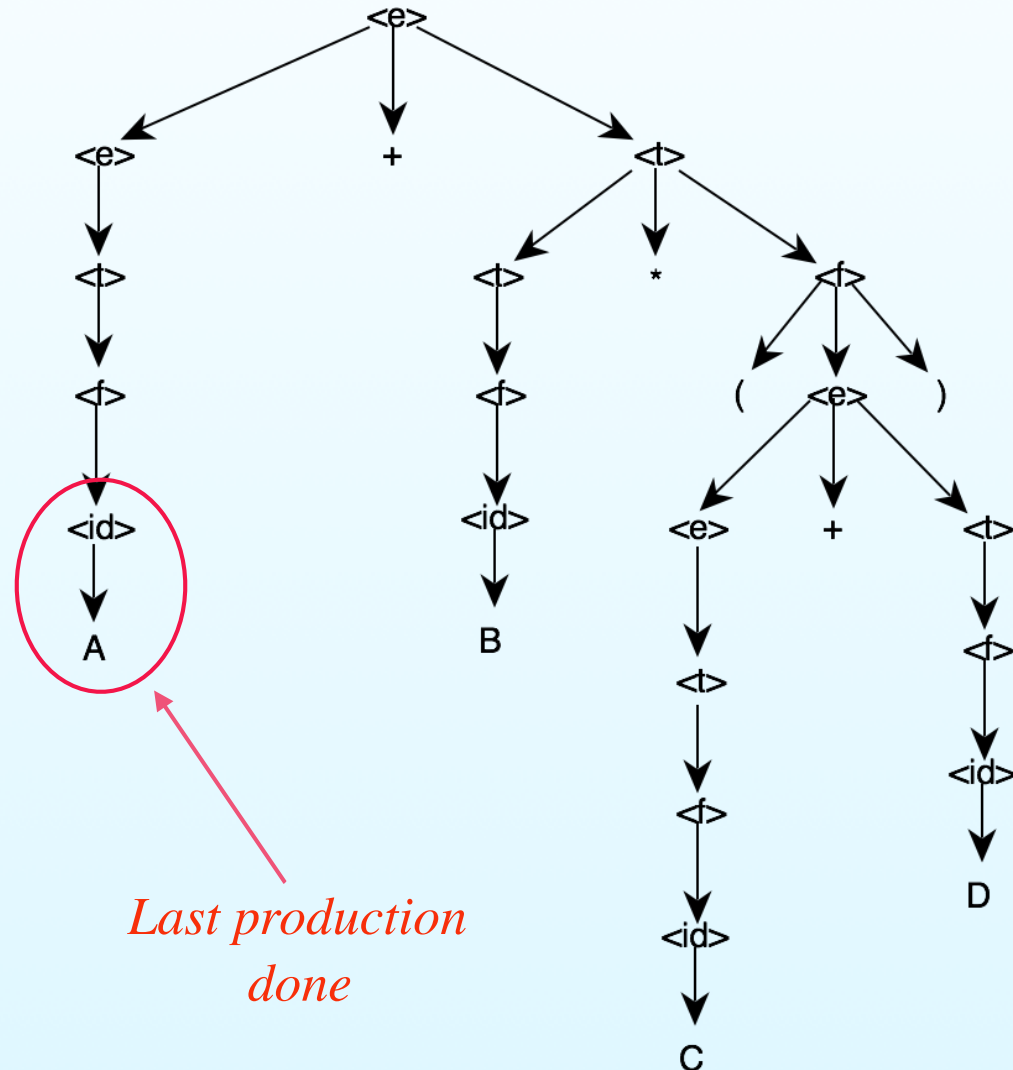
Parse: A + B \* (C + D)

- <e>
- <e> + <t>
- <e> + <t> \* <f>
- <e> + <t> \* (<e>)
- <e> + <t> \* (<e> + <t>)
- <e> + <t> \* (<e> + <f>)
- <e> + <t> \* (<e> + <id>)
- <e> + <t> \* (<e> + D)
- <e> + <t> \* (<t> + D)
- <e> + <t> \* (<f> + D)
- <e> + <t> \* (<id> + D)
- <e> + <t> \* (C + D)
- <e> + <f> \* (C + D)
- <e> + <id> \* (C + D)
- <e> + B \* (C + D)
- <t> + B \* (C + D)
- <f> + B \* (C + D)
- <id> + B \* (C + D)
- A + B \* (C + D)**

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**



*Last production done*

# Finding the handle

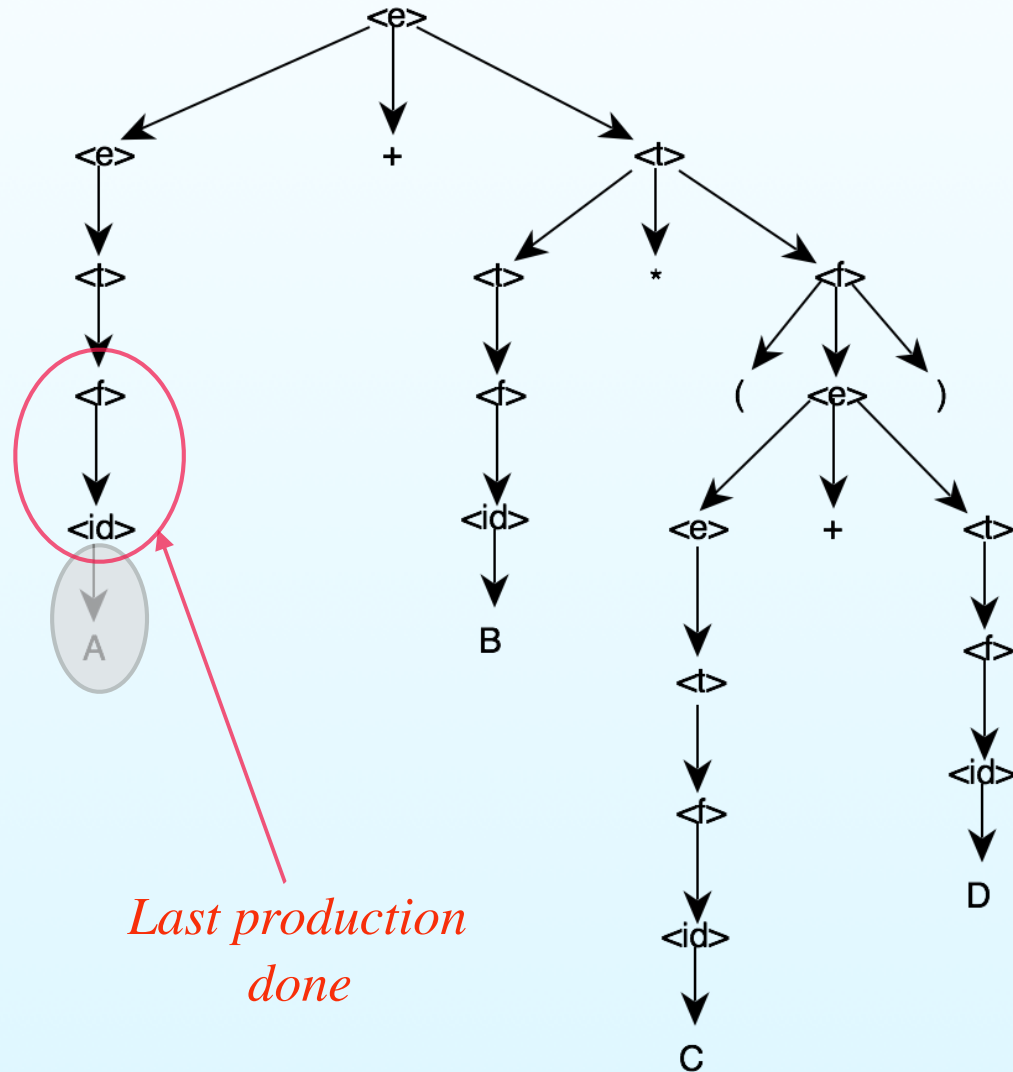
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**



*Last production done*



# Finding the handle

Parse:  $A + B * (C + D)$

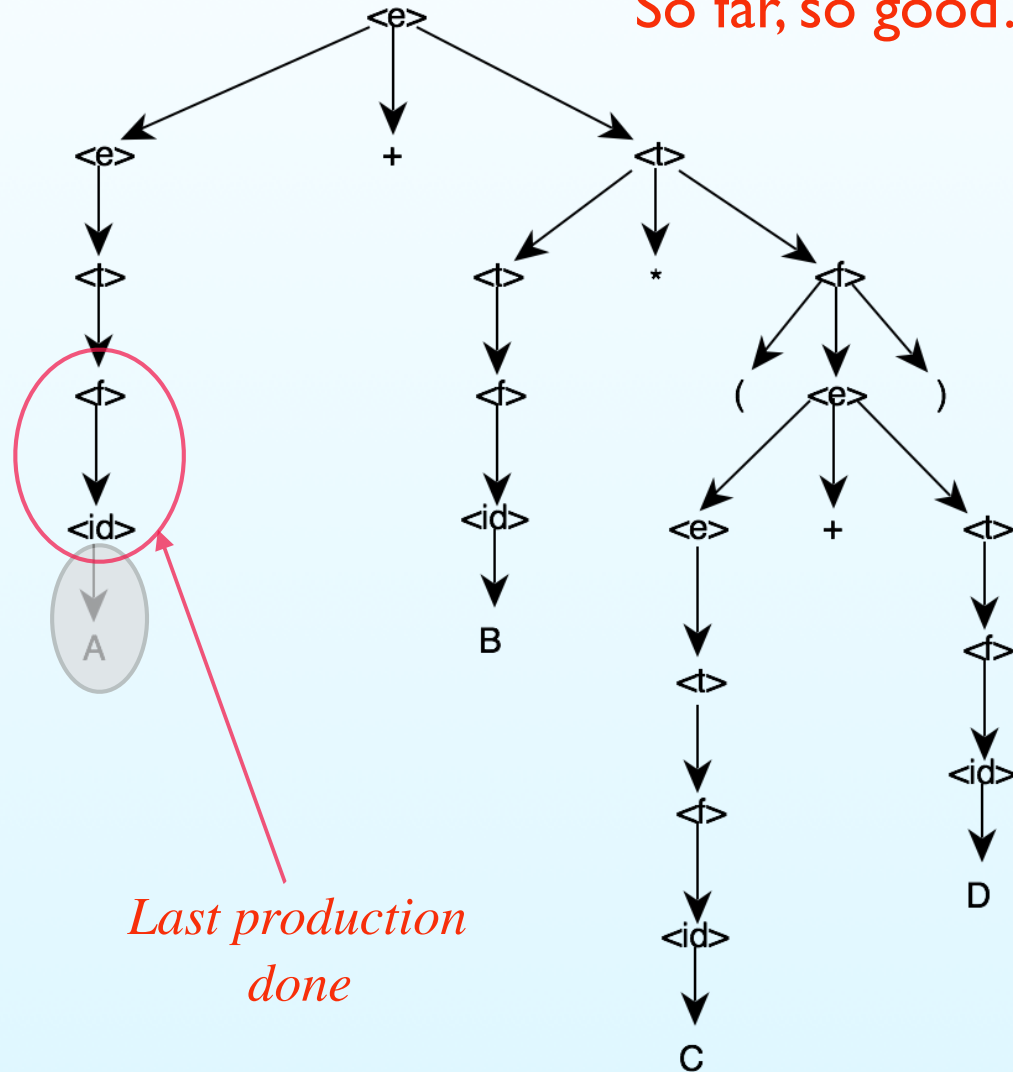
- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- $\langle t \rangle + B * (C + D)$
- $\langle f \rangle + B * (C + D)$
- $\langle id \rangle + B * (C + D)$
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**

**So far, so good...**



*Last production done*

# Finding the handle

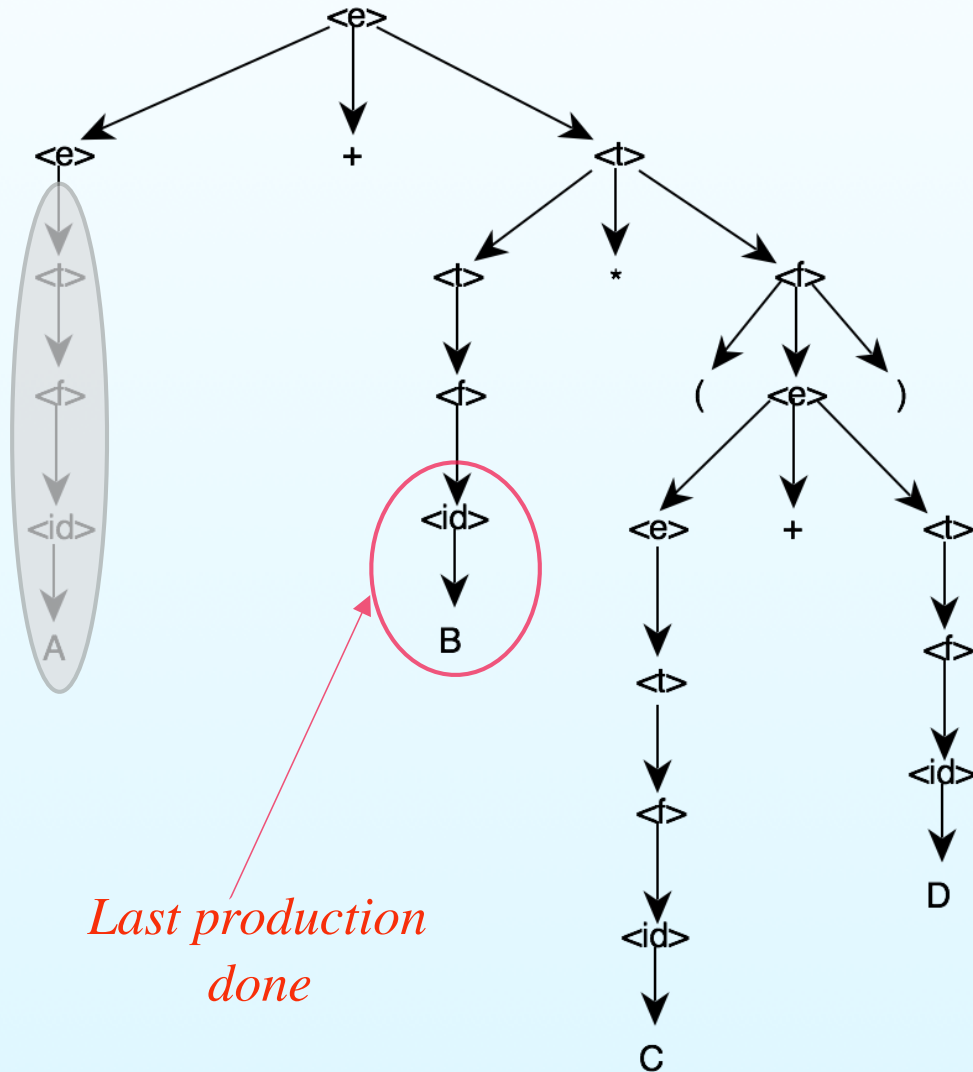
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**



*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

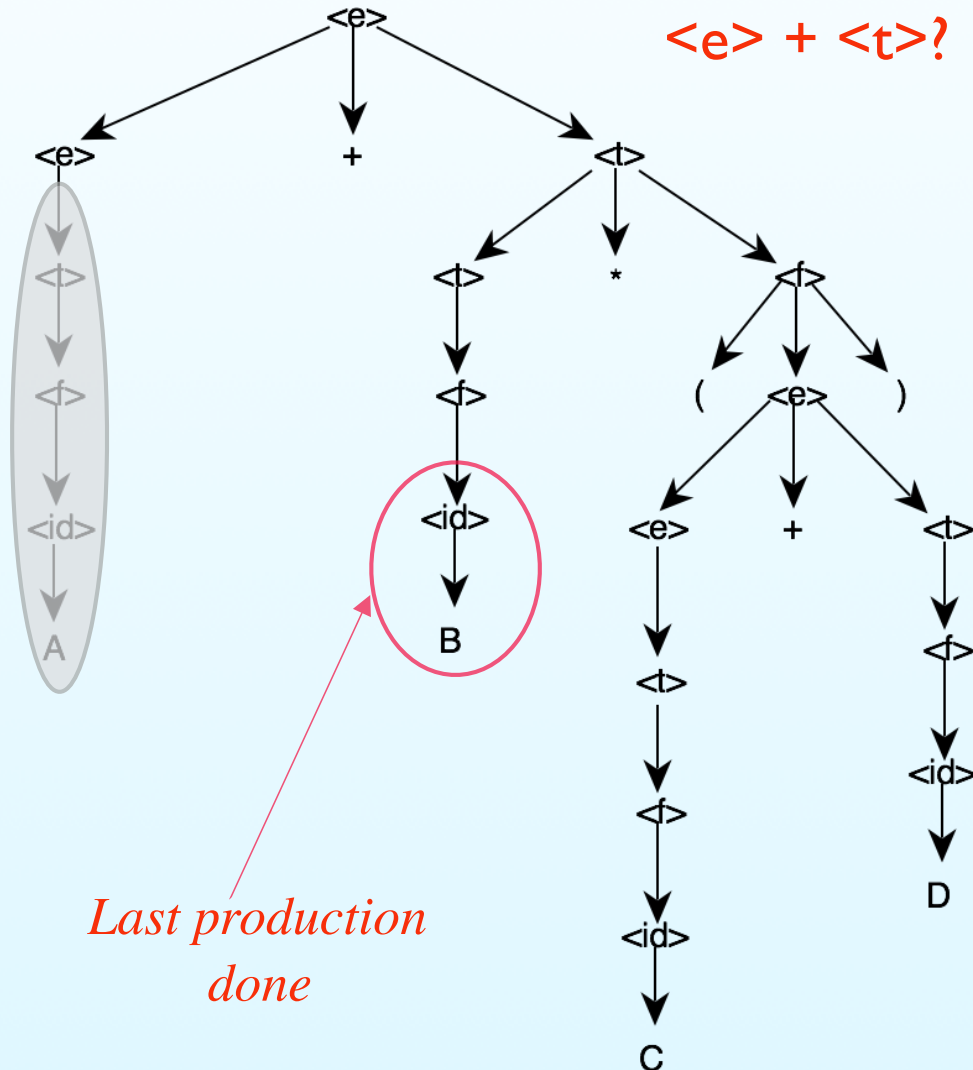
- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**

**No...wouldn't this be  $\langle e \rangle + \langle t \rangle$ ?**



*Last production done*

# Finding the handle

Parse:  $A + B * (C + D)$

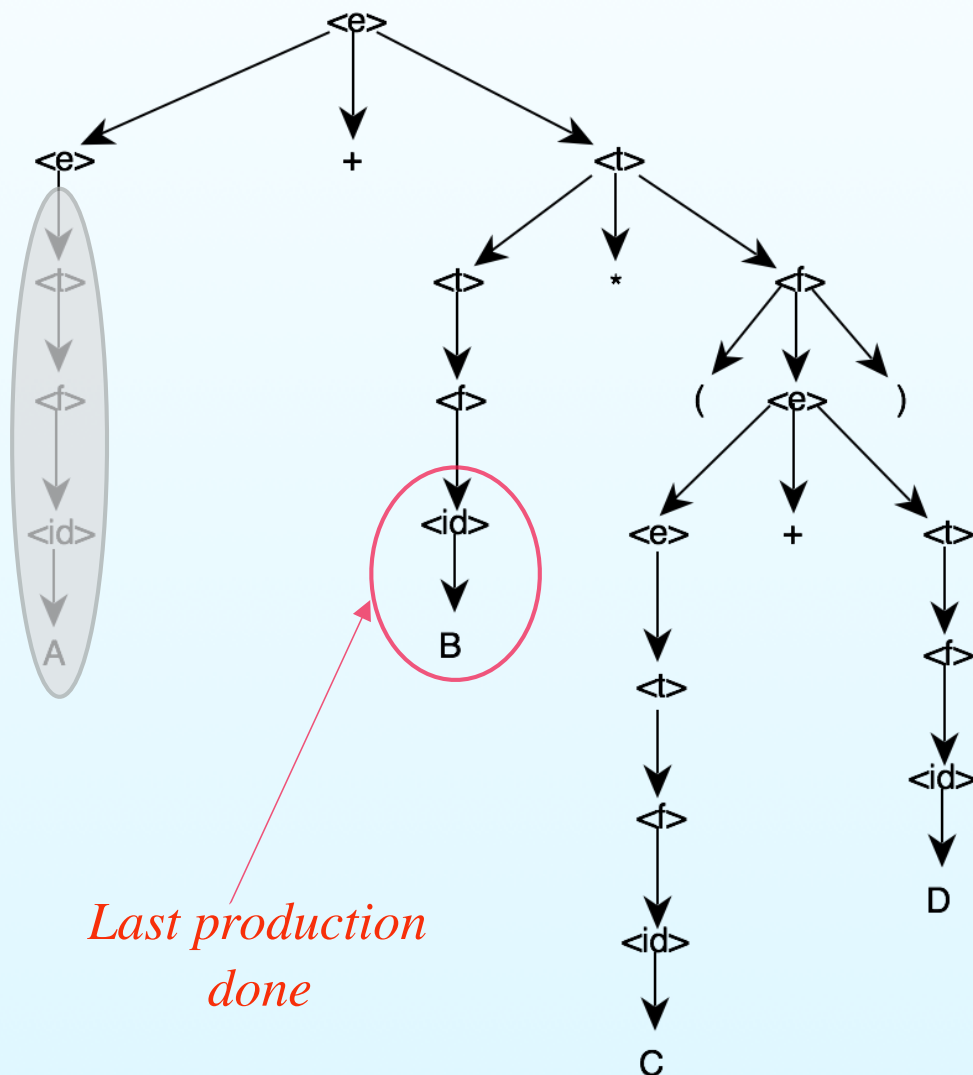
$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Hypothesis: Handle is leftmost RHS?

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

Handle

Last production done



# Finding the handle

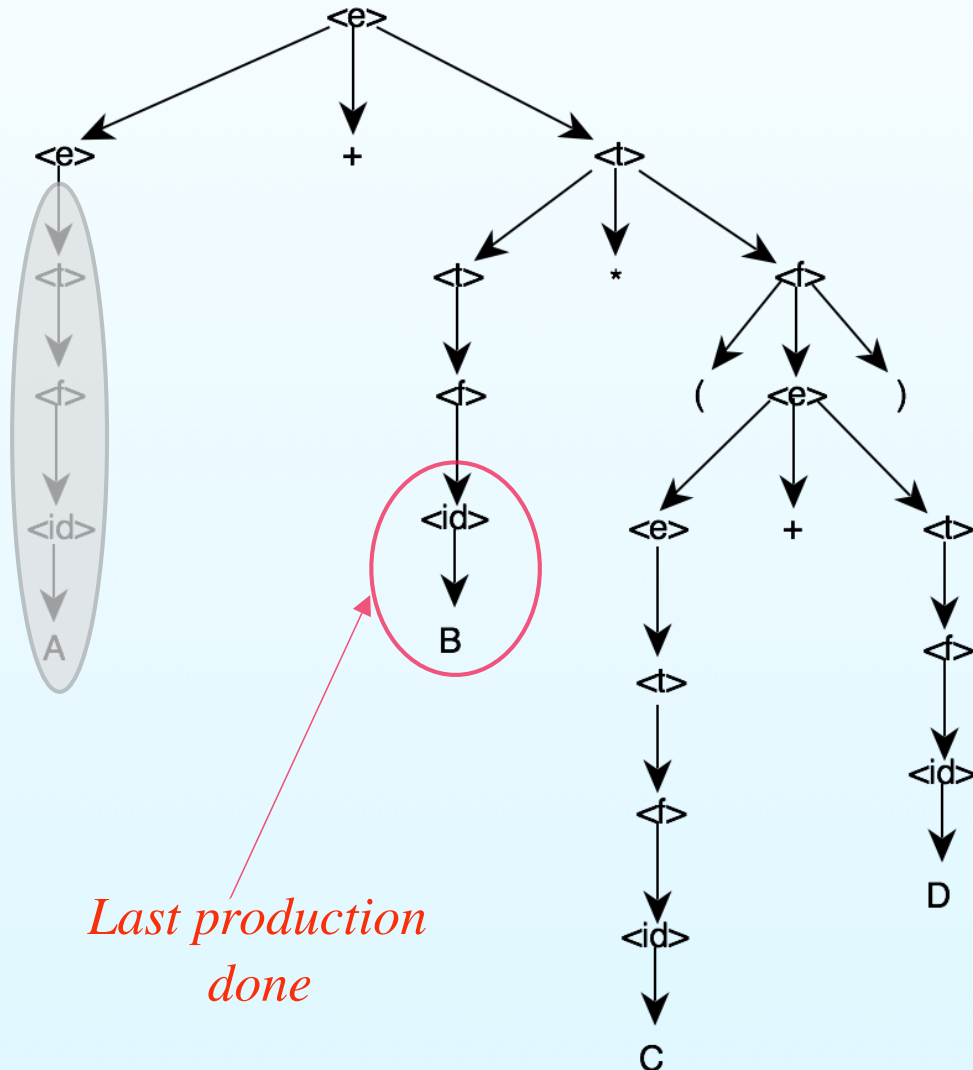
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**  
**What's different?**



*Last production done*

# Finding the handle

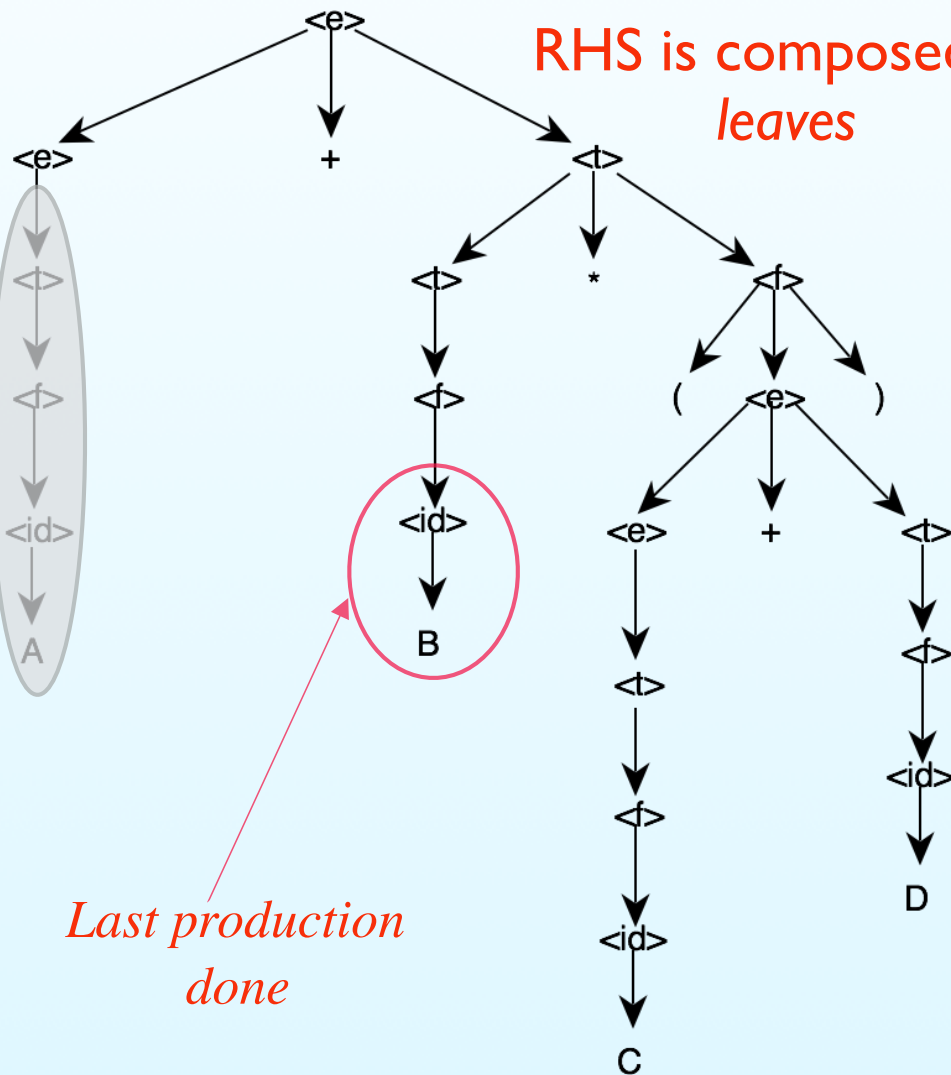
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

*Handle*

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

**Hypothesis: Handle is leftmost RHS?**  
**What's different?**  
**RHS is composed of leaves**



*Last production done*

# Definitions

# Definitions

- Let



# Definitions

- Let
  - $\gamma$ : a right sentential form

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{rm}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume  $rm$ )

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$



# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*



# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Definitions

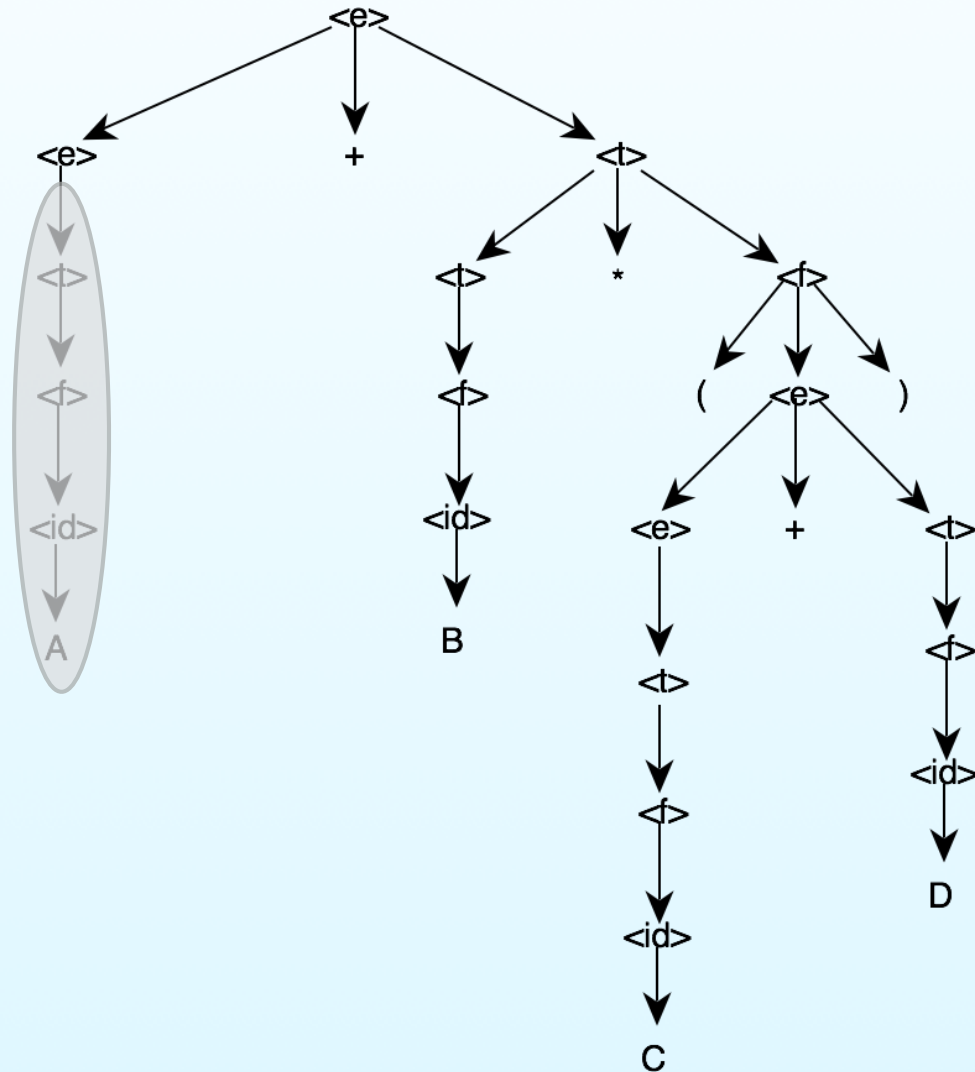
- Let
  - $\gamma$ : a right sentential form
  - $\Rightarrow_{\text{rm}}$  — a single production (right derivation)
  - $\Rightarrow^*$  — 0 or more productions (assume rm)
  - $\Rightarrow^+$  — 1 or more productions
  - $\alpha, \beta, \delta$ : strings of terminals/nonterminals;
  - $w$  — string of terminals
- $\beta$  is a *phrase* of right sentential form  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A \delta \Rightarrow^+ \alpha \beta \delta$$
  - I.e., string of leaves of current parse tree rooted at A
- $\beta$  is a *simple phrase* of  $\gamma = \alpha\beta\delta$  iff
$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \beta \delta$$
  - I.e., string of leaves with single-step derivation from A
- $\beta$  is the *handle* of  $\gamma$  iff it is the *leftmost simple phrase*

# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Phrases:

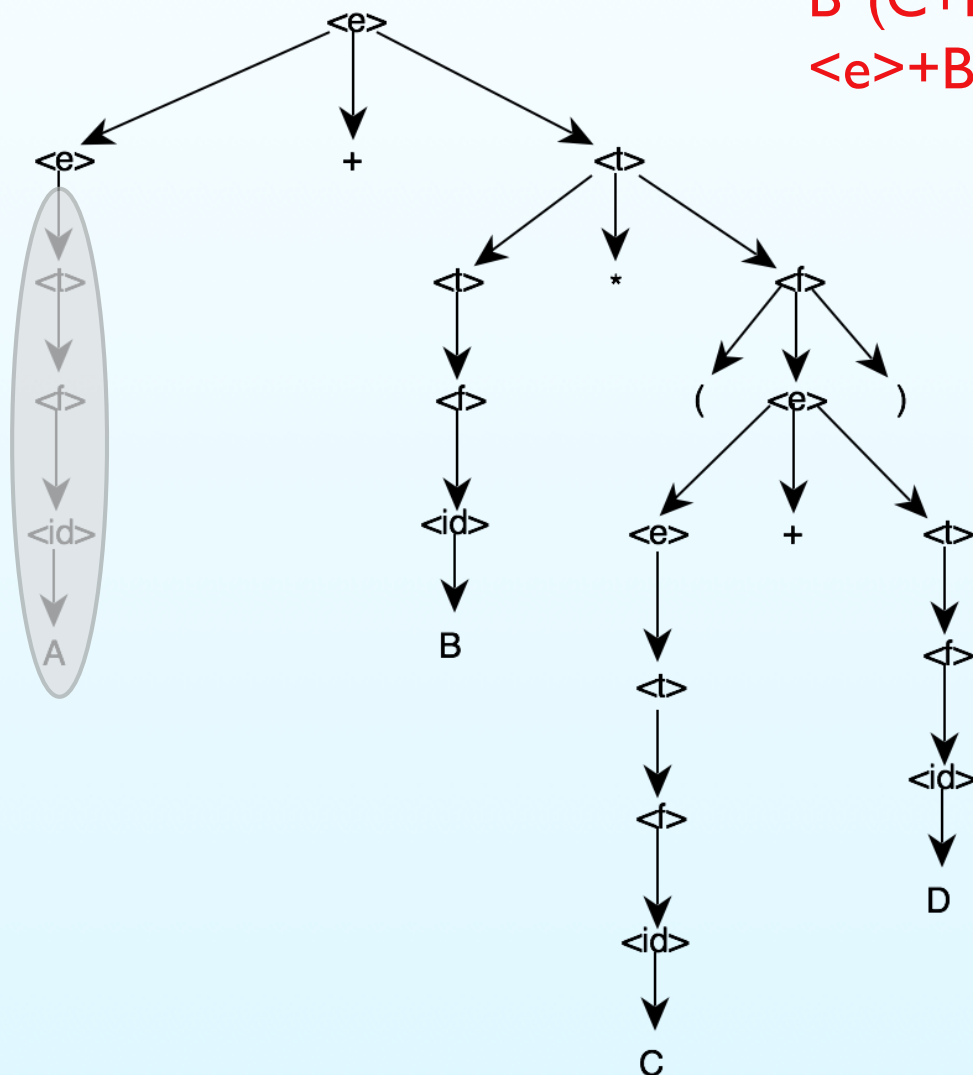
B, C, D

C+D

(C+D)

B\*(C+D)

$\langle e \rangle + B * (C + D)$





# Finding the handle

Parse:  $A + B * (C + D)$

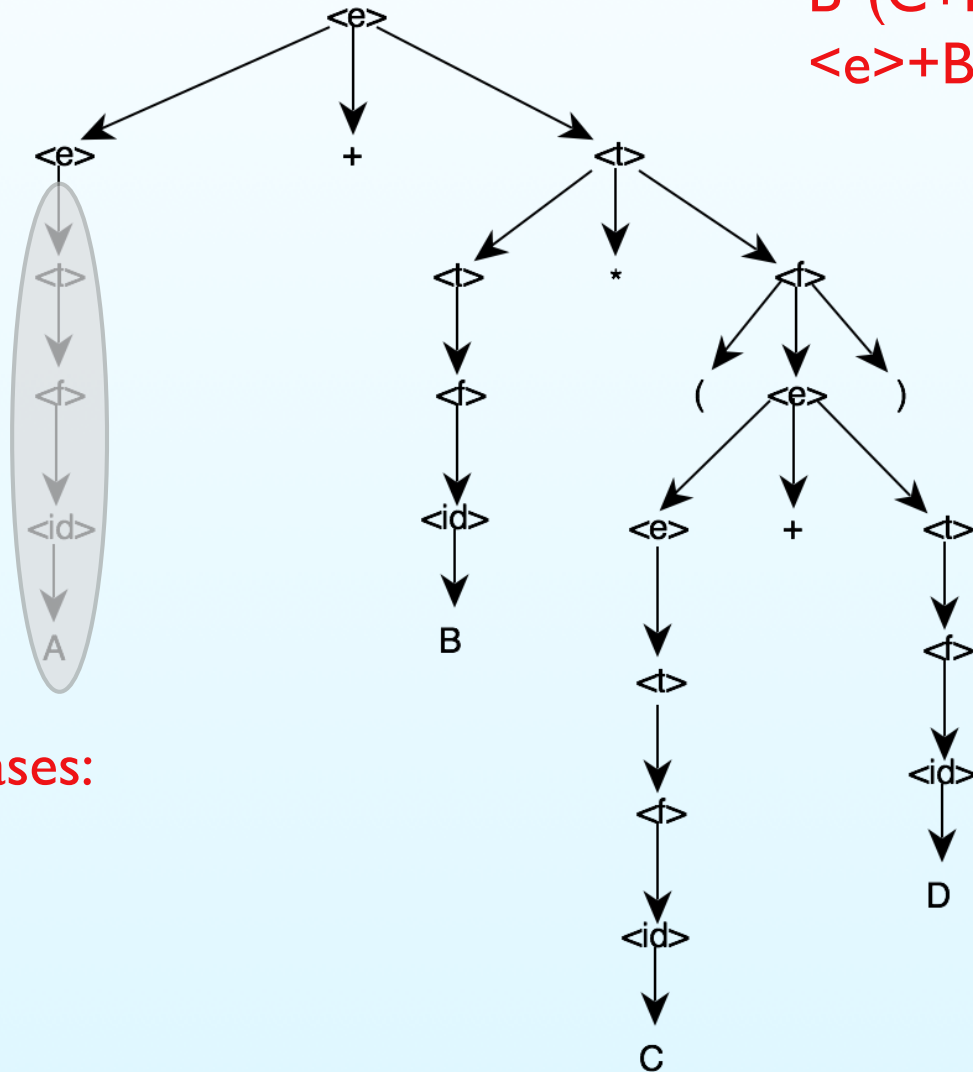
- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Phrases:

- B, C, D
- C+D
- (C+D)
- B\*(C+D)
- $\langle e \rangle + B * (C + D)$

Simple phrases:  
B, C, D



# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$
- $\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$
- $\langle e \rangle + \langle t \rangle * (C + D)$
- $\langle e \rangle + \langle f \rangle * (C + D)$
- $\langle e \rangle + \langle id \rangle * (C + D)$
- $\langle e \rangle + B * (C + D)$
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Phrases:

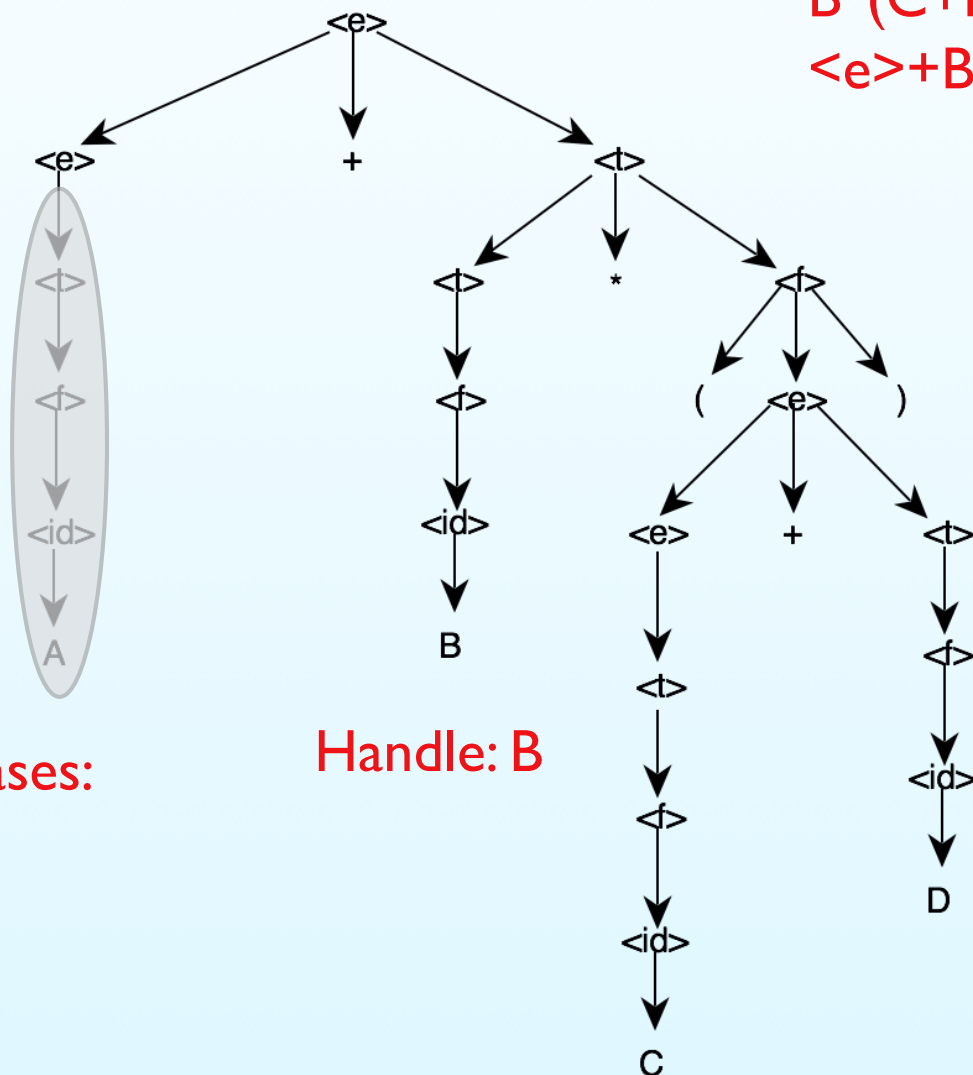
B, C, D

C+D

(C+D)

$B*(C+D)$

$\langle e \rangle + B*(C+D)$



Simple phrases:  
B, C, D

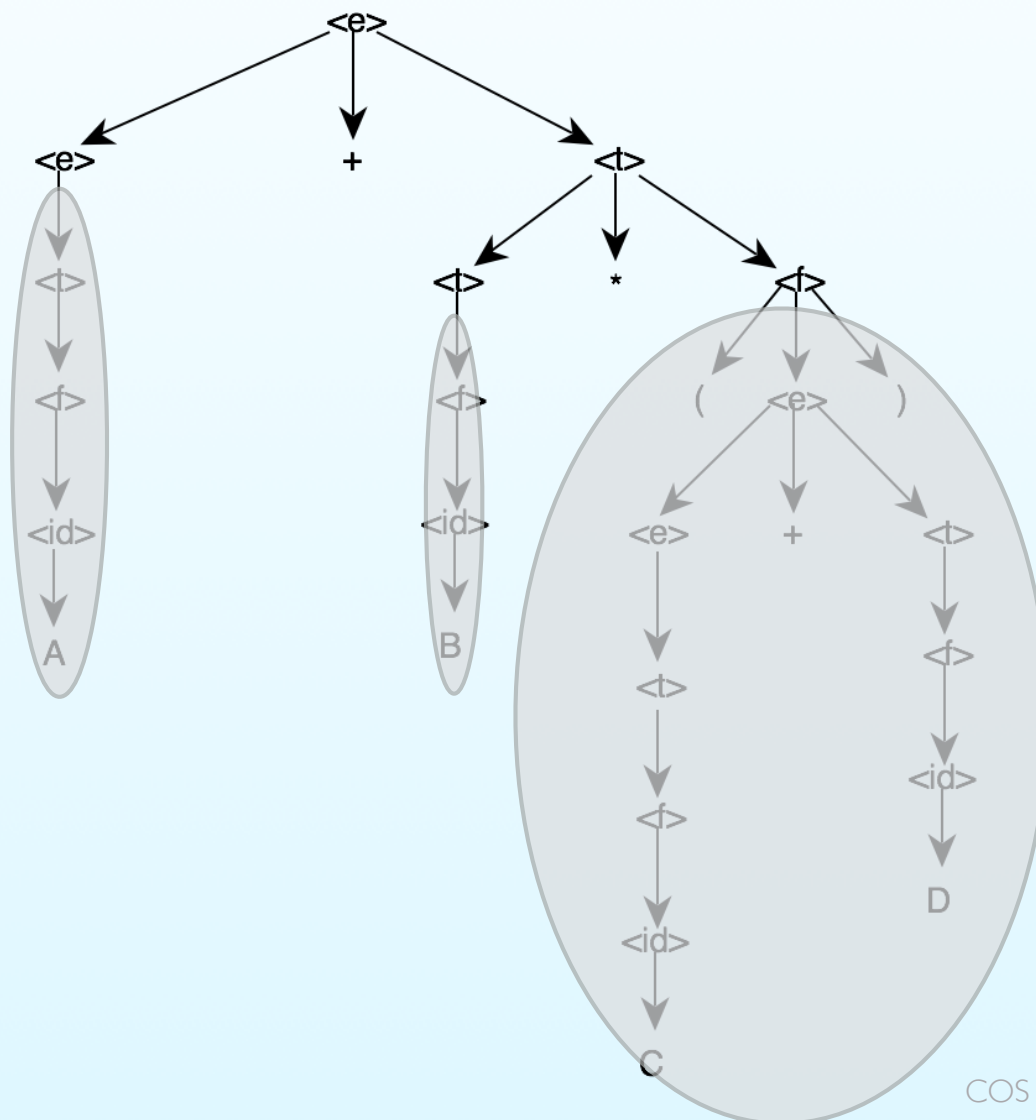
Handle: B

# Finding the handle

Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$



# Finding the handle

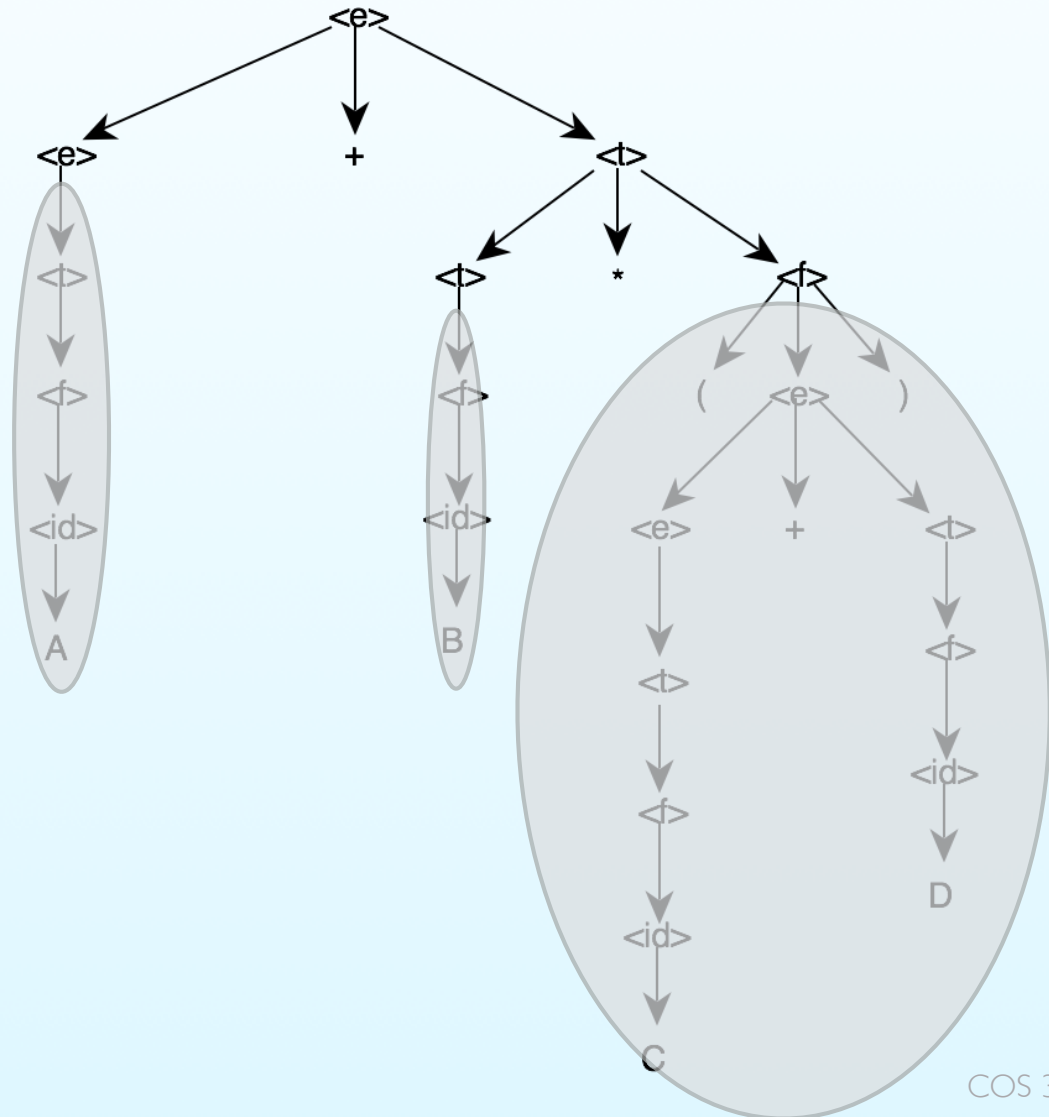
Parse:  $A + B * (C + D)$

- $\langle e \rangle$
- $\langle e \rangle + \langle t \rangle$
- $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~
- ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~
- ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~
- ~~$\langle e \rangle + B * (C + D)$~~
- ~~$\langle t \rangle + B * (C + D)$~~
- ~~$\langle f \rangle + B * (C + D)$~~
- ~~$\langle id \rangle + B * (C + D)$~~
- $A + B * (C + D)$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$

Phrases:

$\langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$



# Finding the handle

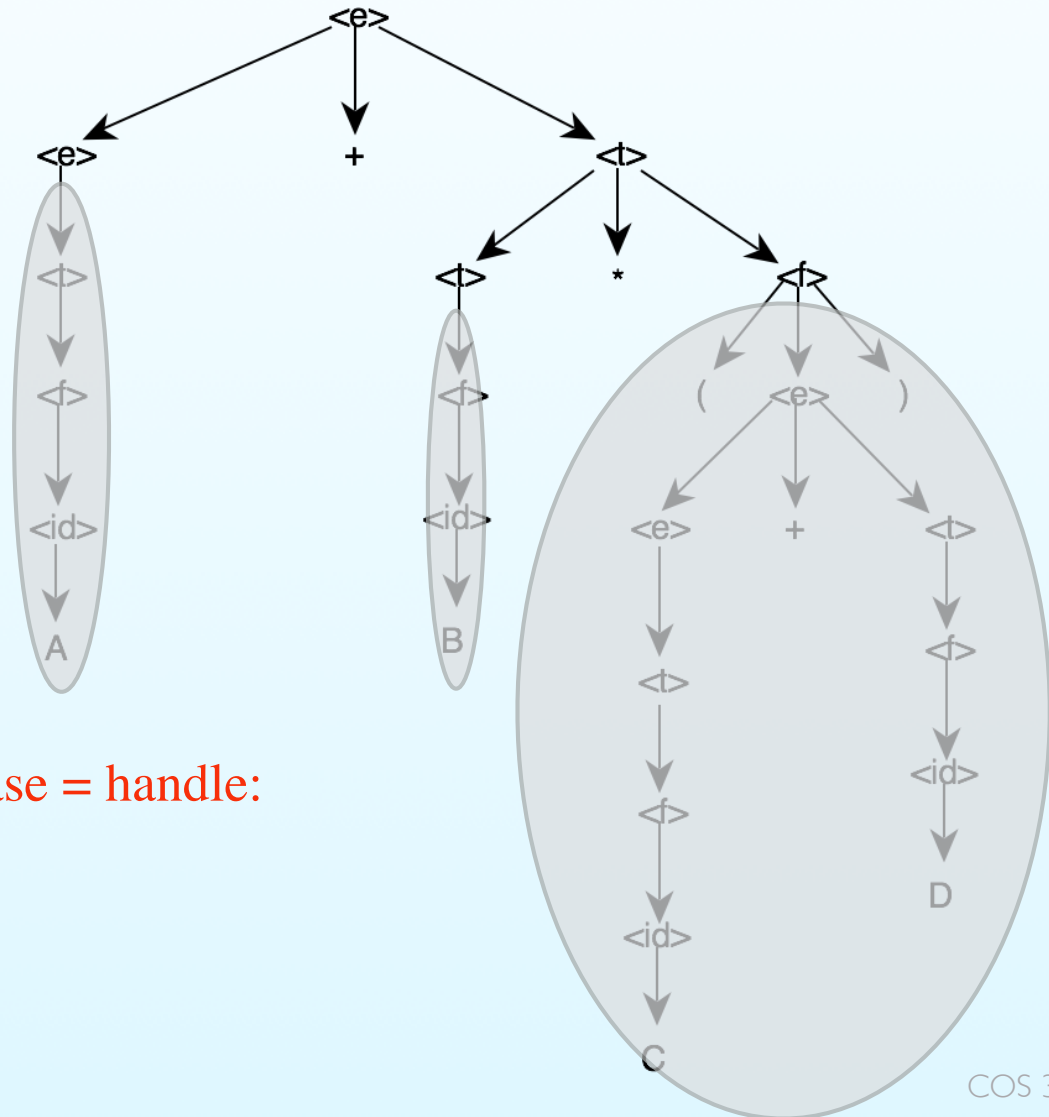
Parse:  $A + B * (C + D)$

$\langle e \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle t \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle f \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + \langle id \rangle)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle e \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle t \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle f \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (\langle id \rangle + D)$~~   
 ~~$\langle e \rangle + \langle t \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle f \rangle * (C + D)$~~   
 ~~$\langle e \rangle + \langle id \rangle * (C + D)$~~   
 ~~$\langle e \rangle + B * (C + D)$~~   
 ~~$\langle t \rangle + B * (C + D)$~~   
 ~~$\langle f \rangle + B * (C + D)$~~   
 ~~$\langle id \rangle + B * (C + D)$~~   
 ~~$A + B * (C + D)$~~

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow (\langle e \rangle) \mid \langle id \rangle$
---

Phrases:

$\langle t \rangle * \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * \langle f \rangle$



Simple phrase = handle:

$\langle t \rangle * \langle f \rangle$

# Finding handles

- Consider:  $\langle e \rangle + \langle t \rangle * \langle f \rangle$
- If have parse tree, know  $\langle t \rangle * \langle f \rangle$  is simple phrase
- But if had parse tree...wouldn't have to parse!
- How to know  $\langle e \rangle + \langle t \rangle$  isn't handle?
- Suppose we chose it as handle
  - $\Rightarrow \langle e \rangle * \langle f \rangle$
  - No rule RHS begins  $\langle e \rangle *$
  - So  $\langle e \rangle + \langle t \rangle$  isn't the handle
- *Lookahead*

# LR(1)

- How far to look ahead?
- Want minimal amount: ideally 1 token
- Grammars that support this: *LR(1) grammars*
  - L: left-to-right (& bottom-up) parsing
  - R: rightmost derivation
  - (1): single token lookahead

# Bottom-up LR(1) parsing

- Start with leftmost token
- If most recently-read leftmost tokens are handle
  - Replace with LHS
- If not, then read next token
- Handle:
  - if most recently-read
  - if lookahead token doesn't conflict with potential LHS



# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle \text{id} \rangle + \langle \text{id} \rangle * ( \langle \text{id} \rangle + \langle \text{id} \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle \text{id} \rangle$

# Parsing example

Sentential form:

Next    Rest

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle \text{id} \rangle + \langle \text{id} \rangle * ( \langle \text{id} \rangle + \langle \text{id} \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle \text{id} \rangle$

# Parsing example

Sentential form:

<u>Next</u>	<u>Rest</u>
<id>	+ <id> * ( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
<id>	<id>	+ <id> * ( <id> + <id> )
<id> +	+	<id> * ( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>



# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )
	don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> *	
<e> + <t> *	(	<id> + <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )
	don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> *	
<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )
	don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> *	
<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )
	don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> *	
<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>



# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )
don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> *		
<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )
don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> *		
<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	
<e> + <t> * ( <e> + <t>	)	

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	
<e> + <t> * ( <e> + <t>	)	
<e> + <t> * ( <e>	)	

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	
<e> + <t> * ( <e> + <t>	)	
<e> + <t> * ( <e>	)	
<e> + <t> * ( <e>	)	

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e>  $\rightarrow$  <e> + <t> | <t>  
 <t>  $\rightarrow$  <t> \* <f> | <f>  
 <f>  $\rightarrow$  (<e>) | <id>



# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	
<e> + <t> * ( <e> + <t>	)	
<e> + <t> * ( <e>	)	
<e> + <t> * ( <e>	)	
<e> + <t> * <f>		

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

	<u>Next</u>	<u>Rest</u>
	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	
<e> + <t> * ( <e> + <t>	)	
<e> + <t> * ( <e>	)	
<e> + <t> * ( <e>	)	
<e> + <t> * <f>		
<e> + <t>		

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing example

Sentential form:

Next

Rest

	<id>	+ <id> * ( <id> + <id> )
<id>	+	<id> * ( <id> + <id> )
<f>	+	<id> * ( <id> + <id> )
<t>	+	<id> * ( <id> + <id> )
<e>	+	<id> * ( <id> + <id> )
<e> +	<id>	* ( <id> + <id> )
<e> + <id>	*	( <id> + <id> )
<e> + <f>	*	( <id> + <id> )
<e> + <t>	*	( <id> + <id> )

don't replace <t> or <e> + <t> w/ <e>: no rule RHS starting w/ <e> \*

<e> + <t> *	(	<id> + <id> )
<e> + <t> * (	<id>	+ <id> )
<e> + <t> * ( <id>	+	<id> )
<e> + <t> * ( <f>	+	<id> )
<e> + <t> * ( <t>	+	<id> )
<e> + <t> * ( <e>	+	<id> )
<e> + <t> * ( <e> +	<id>	)
<e> + <t> * ( <e> + <id>	)	
<e> + <t> * ( <e> + <f>	)	
<e> + <t> * ( <e> + <t>	)	
<e> + <t> * ( <e>	)	
<e> + <t> * ( <e> )		
<e> + <t> * <f>		
<e> + <t>		
<e>		

Parse: A + B \* (C + D)

Assume lex. analy.  $\Rightarrow$  <id> + <id> \* ( <id> + <id> )

<e> $\rightarrow$ <e> + <t>   <t>
<t> $\rightarrow$ <t> * <f>   <f>
<f> $\rightarrow$ (<e>)   <id>

# Parsing

- Intuitively, this is simple...
- ...but how to implement?

# Shift-reduce parsing

- *Shift-reduce parsing:*
  - Bottom-up parsing of LR(1) grammars
  - Automates the process of finding handles
- CFG needs *pushdown automaton*  $\Rightarrow$  *stack*
  - Top-down, recursive-descent parser: runtime stack
  - Shift-reduce parser  $\Rightarrow$  explicit stack
- Stack:
  - Contains parsing *state*
  - Current right sentential form
- Only 2 actions:
  - **shift** a token from input  $\rightarrow$  stack
  - **reduce** top-of-stack ( $n$  frames)  $\rightarrow$  LHS
- Reduce only done when top  $n$  frames = handle

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow \langle \text{id} \rangle + \langle \text{id} \rangle * ( \langle \text{id} \rangle + \langle \text{id} \rangle )$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle \text{id} \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

Next Action

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$\mid \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Next Action

shift



# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$\mid \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Next Action

shift

reduce

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$

$\langle f \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$\mid \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Next Action

shift

reduce

reduce

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$

$\langle f \rangle$

$\langle t \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$\mid \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Next Action

shift

reduce

reduce

reduce

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$\langle id \rangle$	+	$\langle id \rangle$	*	(	$\langle id \rangle$	+	$\langle id \rangle$	)
$\langle id \rangle$	+	$\langle id \rangle$	*	(	$\langle id \rangle$	+	$\langle id \rangle$	)
$\langle id \rangle$	+	$\langle id \rangle$	*	(	$\langle id \rangle$	+	$\langle id \rangle$	)
$\langle id \rangle$	+	$\langle id \rangle$	*	(	$\langle id \rangle$	+	$\langle id \rangle$	)
$\langle id \rangle$	+	$\langle id \rangle$	*	(	$\langle id \rangle$	+	$\langle id \rangle$	)

Next Action

shift  
reduce  
reduce  
reduce  
shift

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$   
 $\langle e \rangle +$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$  \langle id \rangle$	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle +  $		$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$

Next Action

shift  
reduce  
reduce  
reduce  
shift  
shift

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$   
 $\langle e \rangle +$   
 $\langle e \rangle + \langle id \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$  \langle id \rangle$	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle  $	$+$	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$
$\langle id \rangle +  $	$\langle id \rangle$	$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$	
$\langle id \rangle + \langle id \rangle  $		$*$	$($	$\langle id \rangle$	$+$	$\langle id \rangle$	$)$	

Next Action

shift  
reduce  
reduce  
reduce  
shift  
shift  
reduce

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$   
 $\langle e \rangle +$   
 $\langle e \rangle + \langle id \rangle$   
 $\langle e \rangle + \langle f \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$| \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + | \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$

Next Action

shift  
reduce  
reduce  
reduce  
shift  
shift  
reduce  
reduce

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle id \rangle$

$\langle f \rangle$

$\langle t \rangle$

$\langle e \rangle$

$\langle e \rangle +$

$\langle e \rangle + \langle id \rangle$

$\langle e \rangle + \langle f \rangle$

$\langle e \rangle + \langle t \rangle$

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$

$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$

$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

Input

$\mid \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle + \mid \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle + \langle id \rangle \mid * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle + \langle id \rangle \mid * ( \langle id \rangle + \langle id \rangle )$

$\langle id \rangle + \langle id \rangle \mid * ( \langle id \rangle + \langle id \rangle )$

Next Action

shift

reduce

reduce

reduce

shift

shift

reduce

reduce

shift



# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

<u>Stack</u>	$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$ $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$ $\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$	<u>Input</u>	<u>Next Action</u>
		$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$		$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$		$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$		$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$		$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$		$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$		$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$		$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$		$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$		$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$   
 $\langle e \rangle +$   
 $\langle e \rangle + \langle id \rangle$   
 $\langle e \rangle + \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle *$   
 $\langle e \rangle + \langle t \rangle * ($

Input

$| \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + | \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * | ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( | \langle id \rangle + \langle id \rangle )$

Next Action

shift  
reduce  
reduce  
reduce  
shift  
shift  
reduce  
reduce  
shift  
shift  
shift  
shift

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$   
 $\langle e \rangle +$   
 $\langle e \rangle + \langle id \rangle$   
 $\langle e \rangle + \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle *$   
 $\langle e \rangle + \langle t \rangle * ($   
 $\langle e \rangle + \langle t \rangle * ( \langle id \rangle$

Input

$| \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle | + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + | \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle | * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * | ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( | \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( \langle id \rangle | + \langle id \rangle )$

Next Action

shift  
reduce  
reduce  
reduce  
shift  
shift  
reduce  
reduce  
shift  
shift  
shift  
shift  
reduce

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$   
 $\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$   
 $\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

$\langle id \rangle$   
 $\langle f \rangle$   
 $\langle t \rangle$   
 $\langle e \rangle$   
 $\langle e \rangle +$   
 $\langle e \rangle + \langle id \rangle$   
 $\langle e \rangle + \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle *$   
 $\langle e \rangle + \langle t \rangle * ($   
 $\langle e \rangle + \langle t \rangle * ( \langle id \rangle$   
 $\langle e \rangle + \langle t \rangle * ( \langle f \rangle$   
 $\langle e \rangle + \langle t \rangle * ( \langle t \rangle$   
 $\langle e \rangle + \langle t \rangle * ( \langle e \rangle$

Input

$\mid \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle \mid + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \mid \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle \mid * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle \mid * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle \mid * ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * \mid ( \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( \mid \langle id \rangle + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( \langle id \rangle \mid + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( \langle id \rangle \mid + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( \langle id \rangle \mid + \langle id \rangle )$   
 $\langle id \rangle + \langle id \rangle * ( \langle id \rangle \mid + \langle id \rangle )$

Next Action

shift  
reduce  
reduce  
reduce  
shift  
shift  
shift  
reduce  
reduce  
reduce  
shift

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$



# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

<u>Stack</u>	<u>Input</u>	<u>Next Action</u>
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	shift

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce
$\langle e \rangle + \langle t \rangle * \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce
$\langle e \rangle + \langle t \rangle * \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Parsing example

Parse:  $A + B * (C + D)$

Assume lex. analy.  $\Rightarrow$   $\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$

Stack	Input	Next Action
	$  \langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle id \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle f \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle t \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle$	$\langle id \rangle   + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle +$	$\langle id \rangle +   \langle id \rangle * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle   * ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle *$	$\langle id \rangle + \langle id \rangle *   ( \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ($	$\langle id \rangle + \langle id \rangle * (   \langle id \rangle + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle   + \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle +$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle +   \langle id \rangle )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle id \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	reduce
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle   )$	shift
$\langle e \rangle + \langle t \rangle * ( \langle e \rangle )$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce
$\langle e \rangle + \langle t \rangle * \langle f \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce
$\langle e \rangle + \langle t \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	reduce
$\langle e \rangle$	$\langle id \rangle + \langle id \rangle * ( \langle id \rangle + \langle id \rangle )  $	DONE

$\langle e \rangle \rightarrow \langle e \rangle + \langle t \rangle \mid \langle t \rangle$
$\langle t \rangle \rightarrow \langle t \rangle * \langle f \rangle \mid \langle f \rangle$
$\langle f \rangle \rightarrow ( \langle e \rangle ) \mid \langle id \rangle$

# Shift-reduce parser

- Consider all of stack + 1 character (for LR(1)) to determine what to do
- Problem:
  - read input  $\Rightarrow$  stack grows  $\Rightarrow$  slower
- Solution:
  - At any time, parser is in particular *state*
  - Can think of it as a state machine
  - Table-based implementation
  - Index: state + lookahead
  - Specifies what to do, next state



# LR parsers

- *LR parser*: table-based shift-reduce parser
  - Small program
  - Most knowledge of parsing encoded in *parsing table*
- Parsing table:
  - built from grammar
  - producing table can be difficult
  - usually by automated tool (e.g., yacc)
- First LR (“canonical LR”) ← Don Knuth (1965)
  - simpler LR algorithms used to simplify table-building (e.g., SLR, LALR)...
  - ...but work on smaller classes of grammars

# Advantages of LR parsers

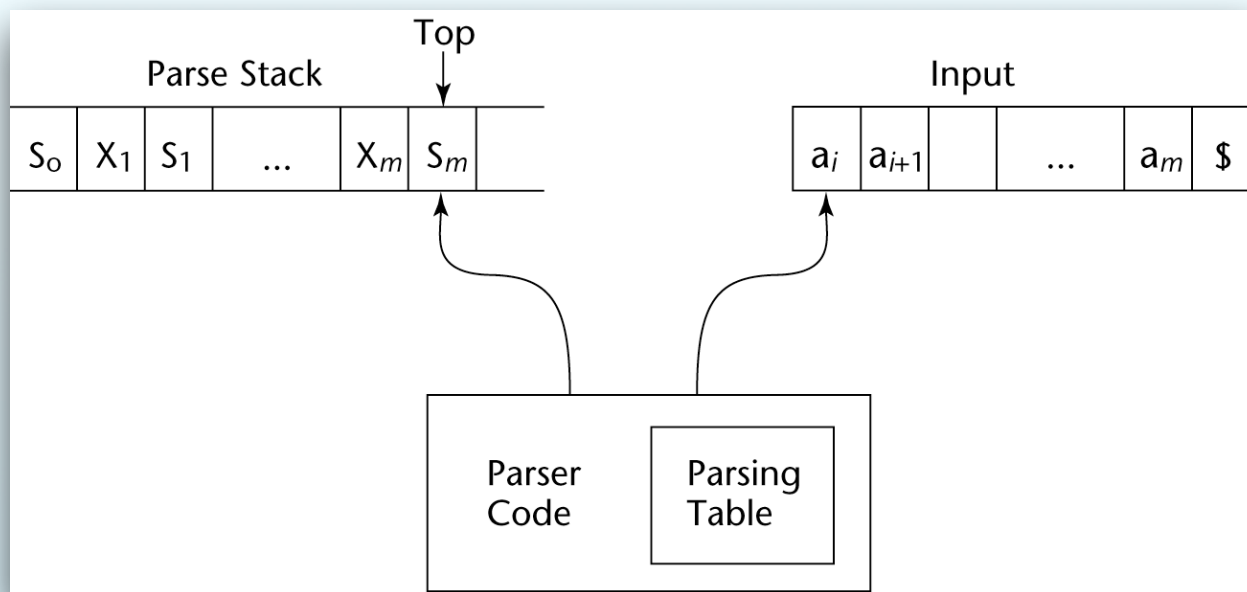
- Can be used for all programming languages
- Detect syntax errors as soon as possible (reading left-to-right)
- LR grammar class  $\supset$  LL grammar class (e.g., LR includes many left recursive grammars)

# LR parsing

- Knuth's insight: entire history of parse is available (in stack) for parser to use to make decision
- There is a finite, small set of different parse situations that have occurred
- So the parse stack can contain the parser state

# LR parsers

- Uses a stack + input string
- At any given time, parser is considered to be in a particular, named *state*
- Stack alternates token/state



# LR parsers

- Parsing table – really two tables:
  - **Action** table:
    - state + next token  $\Rightarrow$  next action
    - actions: shift & go to state; reduce by a rule; accept; error
  - **Goto** table: state + token on TOS  $\Rightarrow$  next state
- Built by (e.g.) yacc or bison
- On error, call error-handling routine

# LR parsing: example

1.  $E \rightarrow E + T$

2.  $E \rightarrow T$

3.  $T \rightarrow T * F$

4.  $T \rightarrow F$

5.  $F \rightarrow (E)$

6.  $F \rightarrow id$

	Action						Goto		
State	id	+	*	(	)	\$	E	T	F
0	S5		S4				1	2	3
1		S6				accept			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			

# LR parsing: example

Stack	Input	Action									
0	id+id*id\$	Shift 5									
0id5			Action							Goto	
0F3		State	id	+	*	(	)	\$	E	T	
0T2		0	S5		S4				1	2	
0EI		1		S6				accept			
0EI+6		2		R2	S7		R2	R2			
0EI+6id5		3		R4	R4		R4	R4			
0EI+6F3		4	S5			S4			8	2	
0EI+6T9		5		R6	R6		R6	R6			
0EI+6T9*7		6	S5			S4				9	
0EI+6T9*7id5		7	S5			S4					
0EI+6T9*7F10		8		S6			S11				
0EI+6T9		9		R1	S7		R1	R1			
0EI		10		R3	R3		R3	R3			
		11		R5	R5		R5	R5			

# LR parsing: example

Which grammar rule to use

Stack		Input						Action			
0		id+id*id\$						Shift 5			
0id5		+id*id\$						Reduce 6, goto 3 (0/F)			
0F3		Action						Goto			
0T2		State	id	+	*	(	)	\$	E	T	F
0E1		0	S5		S4				1	2	3
0E1+6		1		S6				accept			
0E1+6id5		2		R2	S7		R2	R2			
0E1+6F3		3		R4	R4		R4	R4			
0E1+6T9		4	S5			S4			8	2	3
0E1+6T9*7		5		R6	R6		R6	R6			
0E1+6T9*7id		6	S5			S4				9	3
0E1+6T9*7F1		7	S5			S4					10
0E1+6T9		8		S6			S11				
0E1		9		R1	S7		R1	R1			
		10		R3	R3		R3	R3			
		11		R5	R5		R5	R5			



# LR parsing: example

Stack	Input	Action
0	id+id*id\$	Shift 5
0id5	+id*id\$	Reduce 6, goto 3 (0/F)
0F3	+id*id\$	R4, goto 2 (0/T)
0T2	+id*id\$	R2, goto 1 (0/E)
0E1		

State	Action							Goto		
	id	+	*	(	)	\$	E	T	F	
0	S5		S4				1	2	3	
1		S6				accept				
2		R2	S7		R2	R2				
3		R4	R4		R4	R4				
4	S5			S4			8	2	3	
5		R6	R6		R6	R6				
6	S5			S4				9	3	
7	S5			S4					10	
8		S6			S11					
9		R1	S7		R1	R1				
10		R3	R3		R3	R3				
11										

# LR parsing: example

Stack	Input	Action
0	id+id*id\$	Shift 5
0id5	+id*id\$	Reduce 6, goto 3 (0/F)
0F3	+id*id\$	R4, goto 2 (0/T)
0T2	+id*id\$	R2, goto 1 (0/E)
0E1	+id*id\$	Shift 6
0E1+6	id*id\$	Shift 5
0E1+6id5	*id\$	R6, goto 3 (6/F)

Stack	State	Action						Goto		
		id	+	*	(	)	\$	E	T	F
0E1+6F3	0	S5		S4				1	2	3
0E1+6T9	1		S6			accept				
0E1+6T9*7	2		R2	S7		R2	R2			
0E1+6T9*7id	3		R4	R4		R4	R4			
0E1+6T9*7FI	4	S5			S4			8	2	3
0E1+6T9	5		R6	R6		R6	R6			
0E1	6	S5			S4				9	3
	7	S5			S4					10
	8		S6			S11				

State	Action						Goto		
	id	+	*	(	)	\$	E	T	F
0	S5		S4				1	2	3
1		S6				accept			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			

Stack		
0		
0id5		
0F3		
0T2		
0E1		
0E1+6		
0E1+6id5	^id\$	R6, goto 3 (6/F)
0E1+6F3	*id\$	R4, goto 9 (6/T)
0E1+6T9	*id\$	Shift 7
0E1+6T9*7	id\$	Shift 5
0E1+6T9*7id5	\$	R6, goto 10 (7/F)
0E1+6T9*7F10	\$	R3, goto 9 (6/T)
0E1+6T9	\$	R1, goto 1 (0/E)
0E1	R	Accept

# LR parsing: For you

1.  $E \rightarrow E + T$

2.  $E \rightarrow T$

3.  $T \rightarrow T * F$

4.  $T \rightarrow F$

5.  $F \rightarrow (E)$

6.  $F \rightarrow id$

Use table to parse:  
 $id * (id + id)$

	Action						Goto		
State	id	+	*	(	)	\$	E	T	F
0	S5		S4				1	2	3
1		S6				accept			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			