Lecture 10

Syntax Analysis

Awanish Pandey

Department of Computer Science and Engineering
Indian Institute of Technology
Roorkee

February 11, 2025

Take aways from the last class

- Example of Shift-Reduce parsing

Take aways from the last class

- Example of Shift-Reduce parsing
- Issuses in bottom-up parsing

Take aways from the last class

- Example of Shift-Reduce parsing
- Issuses in bottom-up parsing
- Handle

Take aways from the last class

- Example of Shift-Reduce parsing
- Issuses in bottom-up parsing
- Handle
- Conflicts

Take aways from the last class

- Example of Shift-Reduce parsing
- Issuses in bottom-up parsing
- Handle
- Conflicts
- Configuration of LR parser

Parser states

Parser states

- Goal is to know the valid reductions at any given point

Parser states

- Goal is to know the valid reductions at any given point
- Summarize all possible stack prefixes α as a parser state

Parser states

- Goal is to know the valid reductions at any given point
- Summarize all possible stack prefixes α as a parser state
- Parser state is defined by a DFA state that reads in the stack α

Parser states

- Goal is to know the valid reductions at any given point
- Summarize all possible stack prefixes α as a parser state
- Parser state is defined by a DFA state that reads in the stack α
- Accept states of DFA are unique reductions

Constructing parse table

Constructing parse table

- Augment the grammar

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes
  - α is a viable prefix of the grammar if

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes
  - α is a viable prefix of the grammar if
    - ★ There is a w such that αw is a right sentential form.

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes
  - α is a viable prefix of the grammar if
    - There is a w such that αw is a right sentential form.
    - $\alpha.w$ is a configuration of the shift reduce parser

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes
  - α is a viable prefix of the grammar if
    - There is a w such that αw is a right sentential form.
    - $\alpha.w$ is a configuration of the shift reduce parser
  - As long as the parser has viable prefixes on the stack no parser error has been seen

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes
  - α is a viable prefix of the grammar if
    - There is a w such that αw is a right sentential form.
    - $\alpha.w$ is a configuration of the shift reduce parser
  - As long as the parser has viable prefixes on the stack no parser error has been seen
  - The set of viable prefixes is a regular language

Constructing parse table

- Augment the grammar
  - G is a grammar with start symbol S
  - The augmented grammar G' for G has a new start symbol S' and an additional production $S' \rightarrow S$
  - When the parser reduces by this rule it will stop with accept
- Viable prefixes
  - α is a viable prefix of the grammar if
    - There is a w such that αw is a right sentential form.
    - $\alpha.w$ is a configuration of the shift reduce parser
  - As long as the parser has viable prefixes on the stack no parser error has been seen
  - The set of viable prefixes is a regular language
  - Construct an automaton that accepts viable prefixes

LR(0) items

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A$\rightarrow$XYZ gives four LR(0) items

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$
  $A \rightarrow X.YZ$

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$
  $A \rightarrow X.YZ$
  $A \rightarrow XY.Z$

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$
  $A \rightarrow X.YZ$
  $A \rightarrow XY.Z$
  $A \rightarrow XYZ.$

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$
  $A \rightarrow X.YZ$
  $A \rightarrow XY.Z$
  $A \rightarrow XYZ.$
- An item indicates how much of a production has been seen at a point in the process of parsing

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$
  $A \rightarrow X.YZ$
  $A \rightarrow XY.Z$
  $A \rightarrow XYZ.$
- An item indicates how much of a production has been seen at a point in the process of parsing
  - Symbols on the left of "." are already on the stacks

LR(0) items

- An LR(0) item of a grammar G is a production of G with a special symbol "." at some position of the right side
- Production A→XYZ gives four LR(0) items
  $A \rightarrow .XYZ$
  $A \rightarrow X.YZ$
  $A \rightarrow XY.Z$
  $A \rightarrow XYZ.$
- An item indicates how much of a production has been seen at a point in the process of parsing
  - Symbols on the left of "." are already on the stacks
  - Symbols on the right of "." are expected in the input

Start State

Start State

- Start state of DFA is an empty stack corresponding to $S' \rightarrow .S$ item

Start State

- Start state of DFA is an empty stack corresponding to $S' \to .S$ item
  - This means no input has been seen

Start State

- Start state of DFA is an empty stack corresponding to $S' \to .S$ item
  - This means no input has been seen
  - The parser expects to see a string derived from S

Start State

- Start state of DFA is an empty stack corresponding to $S' \rightarrow .S$ item
  - This means no input has been seen
  - The parser expects to see a string derived from S
- Closure of a state adds items for all productions whose LHS occurs in an item in the state, just after "."

Start State

- Start state of DFA is an empty stack corresponding to $S' \rightarrow .S$ item
  - This means no input has been seen
  - The parser expects to see a string derived from S
- Closure of a state adds items for all productions whose LHS occurs in an item in the state, just after "."
  - Set of possible productions to be reduced next

Start State

- Start state of DFA is an empty stack corresponding to $S' \rightarrow .S$ item
  - This means no input has been seen
  - The parser expects to see a string derived from S
- Closure of a state adds items for all productions whose LHS occurs in an item in the state, just after "."
  - Set of possible productions to be reduced next
  - Added items have "." located at the beginning

Start State

- Start state of DFA is an empty stack corresponding to $S' \rightarrow .S$ item
  - This means no input has been seen
  - The parser expects to see a string derived from S
- Closure of a state adds items for all productions whose LHS occurs in an item in the state, just after "."
  - Set of possible productions to be reduced next
  - Added items have "." located at the beginning
  - No symbol of these items is on the stack as yet

Closure operation

Closure operation

- If I is a set of items for a grammar G then $closure(I)$ is a set constructed as follows:

Closure operation

- If I is a set of items for a grammar G then $closure(I)$ is a set constructed as follows:
  - Every item in I is in closure (I)

Closure operation

- If I is a set of items for a grammar G then $closure(I)$ is a set constructed as follows:
  - Every item in I is in closure (I)
  - If $A \rightarrow \alpha.B\beta$ is in $closure(I)$ and $B \rightarrow \gamma$ is a production then $B \rightarrow .\gamma$ is in $closure(I)$

Closure operation

- If I is a set of items for a grammar G then $closure(I)$ is a set constructed as follows:
  - Every item in I is in closure (I)
  - If $A \rightarrow \alpha.B\beta$ is in $closure(I)$ and $B \rightarrow \gamma$ is a production then $B \rightarrow .\gamma$ is in $closure(I)$
- Intuitively $A \rightarrow \alpha.B\beta$ indicates that we might see a string derivable from $B\beta$ as input

Closure operation

- If I is a set of items for a grammar G then $closure(I)$ is a set constructed as follows:
  - Every item in I is in closure (I)
  - If $A \rightarrow \alpha.B\beta$ is in $closure(I)$ and $B \rightarrow \gamma$ is a production then $B \rightarrow .\gamma$ is in $closure(I)$
- Intuitively $A \rightarrow \alpha.B\beta$ indicates that we might see a string derivable from $B\beta$ as input
- If input $B \rightarrow \gamma$ is a production then we might see a string derivable from γ at this point

Example

- Consider the grammar
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$

Example

- Consider the grammar
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- If I is $E' \rightarrow .E$ then closure(I) is

Example

- Consider the grammar
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- If I is $E' \rightarrow .E$ then closure(I) is
  $E' \rightarrow .E$

Example

- Consider the grammar
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- If I is $E' \rightarrow .E$ then closure(I) is
  $E' \rightarrow .E$
  $E \rightarrow .E + T$

Example

- Consider the grammar
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- If I is $E' \rightarrow .E$ then closure(I) is
  $E' \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$

Example

- Consider the grammar
  $E' \to E$
  $E \to E + T | T$
  $T \to T * F | F$
  $F \to (E) | id$
- If I is $E' \to .E$ then closure(I) is
  $E' \to .E$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$

Example

- Consider the grammar
  $E' \to E$
  $E \to E + T | T$
  $T \to T * F | F$
  $F \to (E) | id$
- If I is $E' \to .E$ then closure(I) is
  $E' \to .E$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$

Example

- Consider the grammar
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- If I is $E' \rightarrow .E$ then closure(I) is
  $E' \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .id$

Example

- Consider the grammar
  $E' \to E$
  $E \to E + T | T$
  $T \to T * F | F$
  $F \to (E) | id$
- If I is $E' \to .E$ then closure(I) is
  $E' \to .E$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .id$
  $F \to .(E)$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$

Example

- Grammar:
  $E' \to E$
  $E \to E + T | T$
  $T \to T * F | F$
  $F \to (E) | id$
- $I_0 : closure(E' \to .E)$
  $E \to .E$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$
- $I_1 : goto(I_0, E)$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$
  $E \rightarrow T.$
  $T \rightarrow T. * F$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$
  $E \rightarrow T.$
  $T \rightarrow T. * F$
- $I_3 : goto(I_0, F)$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$
  $E \rightarrow T.$
  $T \rightarrow T. * F$
- $I_3 : goto(I_0, F)$
  $T \rightarrow F.$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$
  $E \rightarrow T.$
  $T \rightarrow T. * F$
- $I_3 : goto(I_0, F)$
  $T \rightarrow F.$
- $I_4 : goto(I_0, ()$

Example

- Grammar:
  $E' \to E$
  $E \to E + T | T$
  $T \to T * F | F$
  $F \to (E) | id$
- $I_0 : closure(E' \to .E)$
  $E \to .E$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$
- $I_1 : goto(I_0, E)$
  $E \to E.$
  $E \to E. + T$
- $I_2 : goto(I_0, T)$
  $E \to T.$
  $T \to T. * F$
- $I_3 : goto(I_0, F)$
  $T \to F.$
- $I_4 : goto(I_0, ()$
  $F \to (.E)$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$

Example

- Grammar:
  $E' \to E$
  $E \to E + T | T$
  $T \to T * F | F$
  $F \to (E) | id$
- $I_0 : closure(E' \to .E)$
  $E \to .E$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$
- $I_1 : goto(I_0, E)$
  $E \to E.$
  $E \to E. + T$
- $I_2 : goto(I_0, T)$
  $E \to T.$
  $T \to T. * F$
- $I_3 : goto(I_0, F)$
  $T \to F.$
- $I_4 : goto(I_0, ()$
  $F \to (.E)$
  $E \to .E + T$
  $E \to .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$
- $I_5 : goto(I_0, id)$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$
  $E \rightarrow T.$
  $T \rightarrow T. * F$
- $I_3 : goto(I_0, F)$
  $T \rightarrow F.$
- $I_4 : goto(I_0, ()$
  $F \rightarrow (.E)$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_5 : goto(I_0, id)$
  $F \rightarrow id.$

Example

- Grammar:
  $E' \rightarrow E$
  $E \rightarrow E + T | T$
  $T \rightarrow T * F | F$
  $F \rightarrow (E) | id$
- $I_0 : closure(E' \rightarrow .E)$
  $E \rightarrow .E$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_1 : goto(I_0, E)$
  $E \rightarrow E.$
  $E \rightarrow E. + T$
- $I_2 : goto(I_0, T)$
  $E \rightarrow T.$
  $T \rightarrow T. * F$
- $I_3 : goto(I_0, F)$
  $T \rightarrow F.$
- $I_4 : goto(I_0, ()$
  $F \rightarrow (.E)$
  $E \rightarrow .E + T$
  $E \rightarrow .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_5 : goto(I_0, id)$
  $F \rightarrow id.$

Example

- $I_6 : goto(I_1, +)$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$

Example

- $I_6 : goto(I_1, +)$
  $E \to E + .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$
- $I_7 : goto(I_2, *)$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  F $\rightarrow (E.)$
  $E \rightarrow E. + T$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  F $\rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  F $\rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$

Example

- $I_6 : goto(I_1, +)$
  $E \to E + .T$
  $T \to .T * F$
  $T \to .F$
  $F \to .(E)$
  $F \to .id$
- $I_7 : goto(I_2, *)$
  $T \to T * .F$
  $F \to .(E)$
  $F \to .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \to (E.)$
  $E \to E. + T$
- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \to E + T.$
  $T \to T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$

Example

- I_6 : $goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- I_7 : $goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- I_8 : $goto(I_4, E)$
  F $\rightarrow (E.)$
  $E \rightarrow E. + T$
- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- I_9 : $goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$
- $goto(I_6, id) is \quad I_5$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$
- $goto(I_6, id) is \quad I_5$

- $I_{10} : goto(I_7, F)$
  $T \rightarrow T * F.$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  F $\rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$
- $goto(I_6, id) is \quad I_5$

- $I_{10} : goto(I_7, F)$
  $T \rightarrow T * F.$
- $goto(I_7, () is \quad I_4$
  $goto(I_7, id) is \quad I_5$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  F $\rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$
- $goto(I_6, id) is \quad I_5$

- $I_{10} : goto(I_7, F)$
  $T \rightarrow T * F.$
- $goto(I_7, () is \quad I_4$
  $goto(I_7, id) is \quad I_5$
- $I_1 1 : goto(I8,))$
  $F \rightarrow (E).$

Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$
- $goto(I_6, id) is \quad I_5$

- $I_{10} : goto(I_7, F)$
  $T \rightarrow T * F.$
- $goto(I_7, () is \quad I_4$
  $goto(I_7, id) is \quad I_5$
- $I_1 1 : goto(I8,))$
  $F \rightarrow (E).$
- $goto(I_8, +) is \quad I_6$

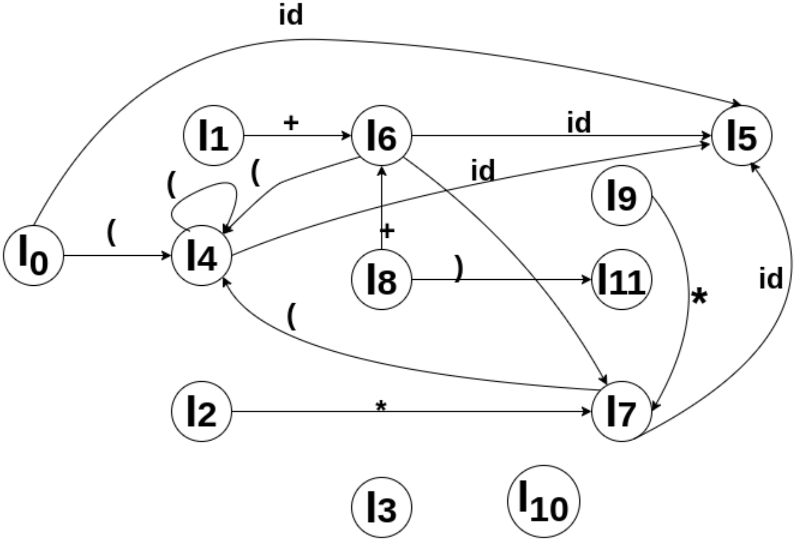
Example

- $I_6 : goto(I_1, +)$
  $E \rightarrow E + .T$
  $T \rightarrow .T * F$
  $T \rightarrow .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_7 : goto(I_2, *)$
  $T \rightarrow T * .F$
  $F \rightarrow .(E)$
  $F \rightarrow .id$
- $I_8 : goto(I_4, E)$
  $\mathrm{F} \rightarrow (E.)$
  $E \rightarrow E. + T$

- $goto(I_4, T) is \quad I_2$
- $goto(I_4, F) is \quad I_3$
- $goto(I_4, () is \quad I_4$
- $goto(I_4, id) is \quad I_5$
- $I_9 : goto(I_6, T)$
  $E \rightarrow E + T.$
  $T \rightarrow T. * F$
- $goto(I_6, F) is \quad I_3$
- $goto(I_6, () is \quad I_4$
- $goto(I_6, id) is \quad I_5$

- $I_{10} : goto(I_7, F)$
  $T \rightarrow T * F.$
- $goto(I_7, () is \quad I_4$
  $goto(I_7, id) is \quad I_5$
- $I_1 1 : goto(I8,))$
  $F \rightarrow (E).$
- $goto(I_8, +) is \quad I_6$
- $goto(I_9, *) is \quad I_7$

Example



Example