Contexts of parsing decisions

Ondřej Navrátil

FIT BUT

7. prosince 2012

Outline

Left context

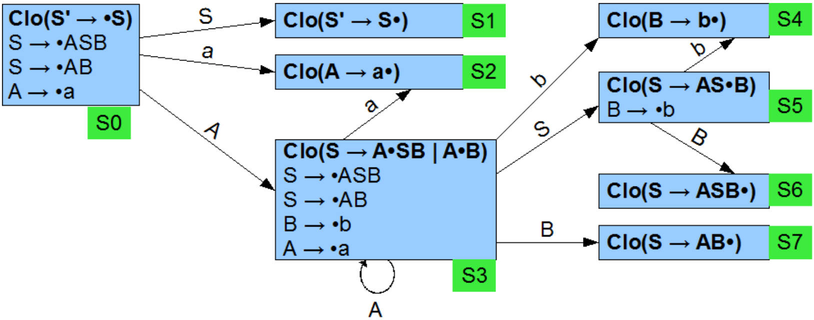
Definition, attributes

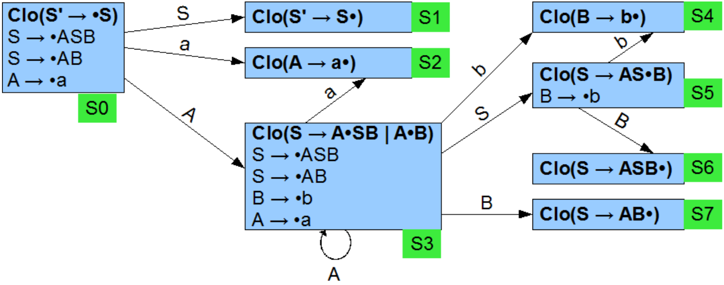
- Bound to a **state**
- Describes all possible stack configurations in that state
- Set of all strings of symbols $(N \cup T)$ that lead to that state
- Usually infinite, but **regular**
- Contexts are **disjoint** across states
- We omit the state information

Left context - Example

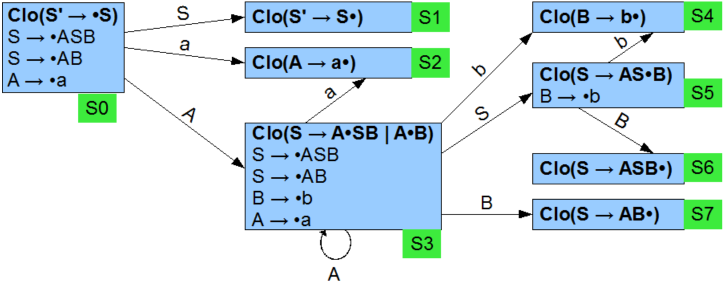
Example CFG $G = (N, T, P, S')$

- $N = \{S', S, A, B\}$
- $T = \{a, b\}$
- $P = \{S' \rightarrow S, S \rightarrow ASB | AB, A \rightarrow a, B \rightarrow b\}$
- $L(G) = a^n b^n$



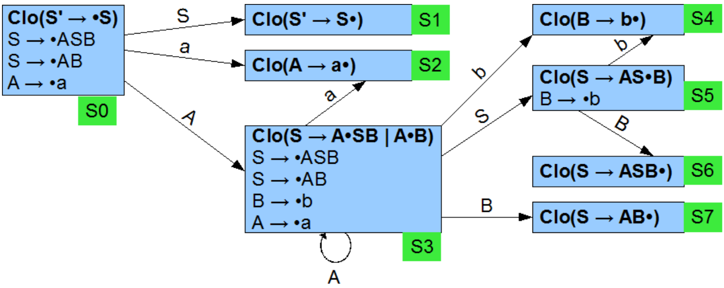


Grammar for left contexts



Grammar for left contexts

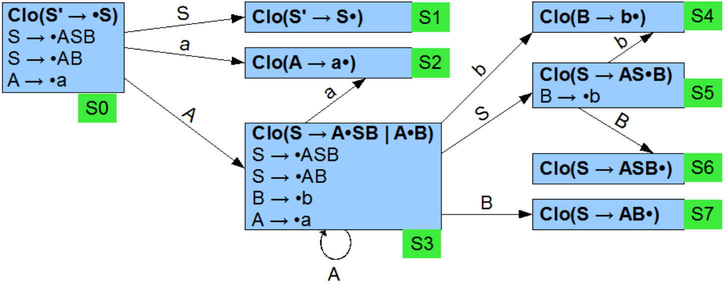
$G = (N = \{P_{0..7}\}, T = \{S, A, B, a, b\}, P, S)$



Grammar for left contexts

$G = (N = \{P_{0..7}\}, T = \{S, A, B, a, b\}, P, S)$

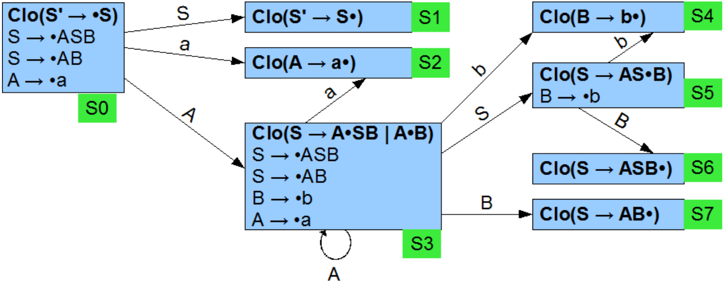
- $P_0 \to \epsilon$



Grammar for left contexts

$G = (N = \{P_{0..7}\}, T = \{S, A, B, a, b\}, P, S)$

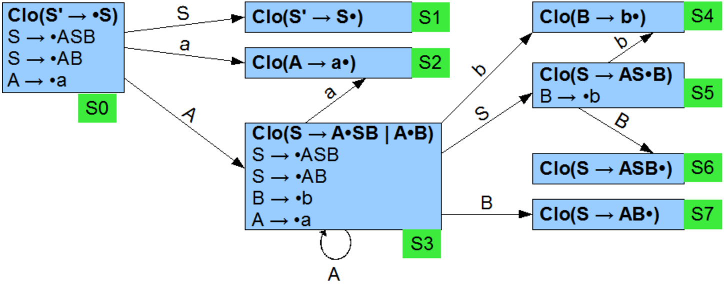
- $P_0 \rightarrow \epsilon$
- $P_1 \rightarrow P_0 S$



Grammar for left contexts

$G = (N = \{P_{0..7}\}, T = \{S, A, B, a, b\}, P, S)$

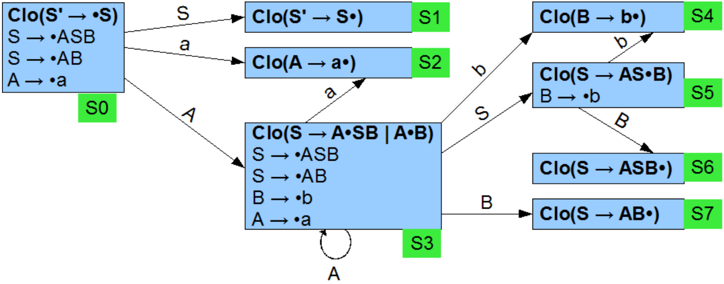
- $P_0 \rightarrow \epsilon$
- $P_1 \rightarrow P_0 S$
- $P_2 \rightarrow P_0 a$
- $P_2 \rightarrow P_3 a$



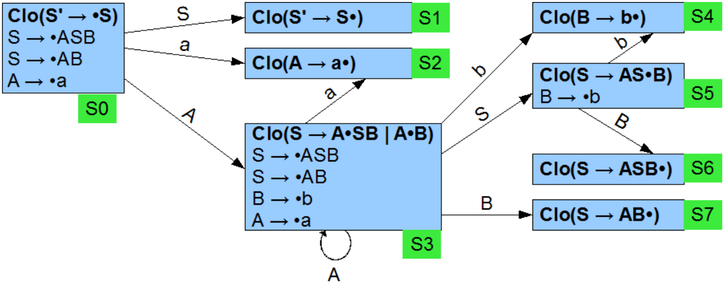
Grammar for left contexts

$G = (N = \{P_{0..7}\}, T = \{S, A, B, a, b\}, P, S)$

- $P_0 \to \epsilon$
- $P_1 \to P_0 S$
- $P_2 \to P_0 a$
- $P_2 \to P_3 a$
- $P_3 \to P_0 A$
- $P_3 \to P_3 A$
- $P_4 \to P_3 b | P_5 b$
- $P_5 \to P_3 S$
- $P_6 \to P_5 B$
- $P_7 \to P_3 B$

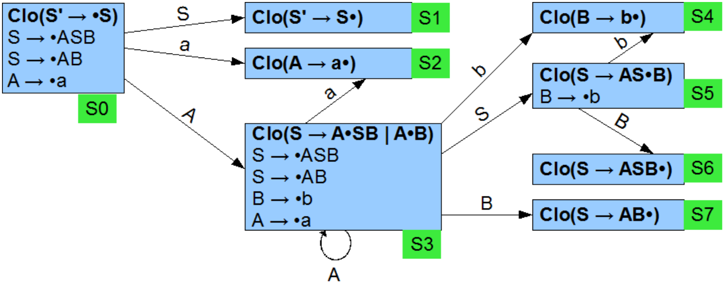


Automaton for left contexts



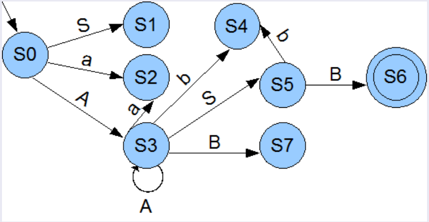
Automaton for left contexts

Same structure, final state = state we want the context for.



Automaton for left contexts

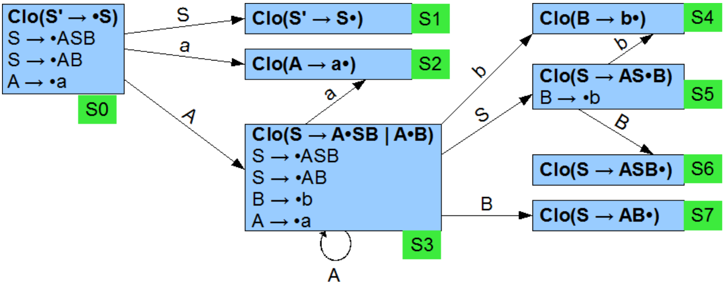
Same structure, final state = state we want the context for.



DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

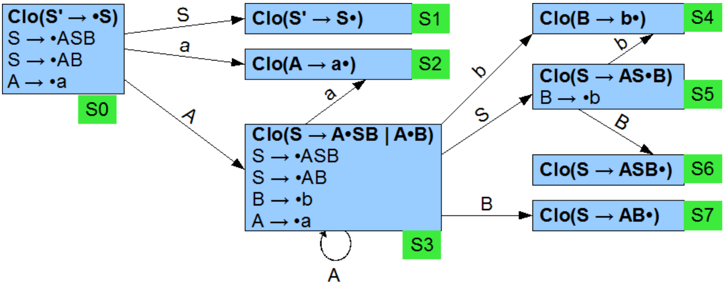


DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

[A b]

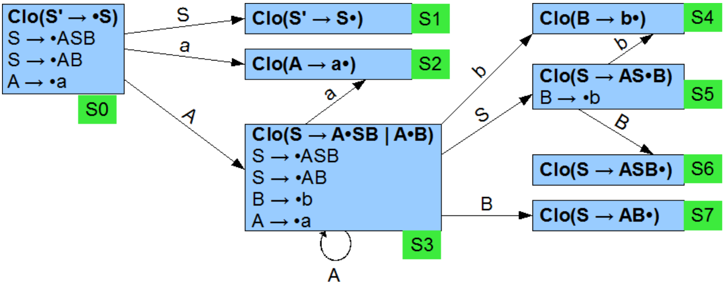


DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

[A b] $\Rightarrow$ S4;

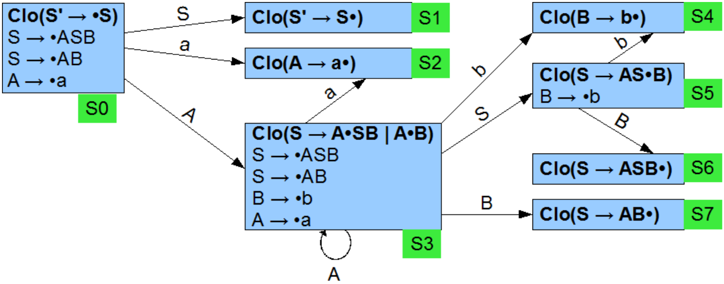


DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

[A b] $\Rightarrow$ S4; [A a]

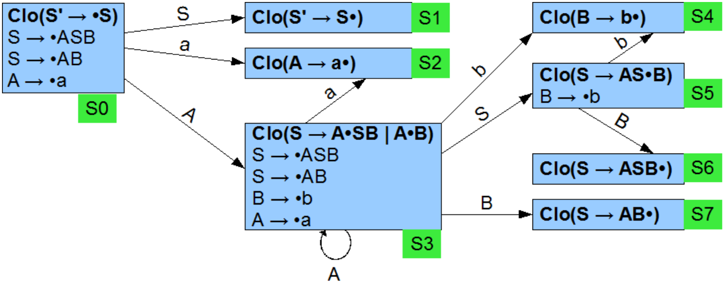


DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

[A b] $\Rightarrow$ S4; [A a] $\Rightarrow$ S2;

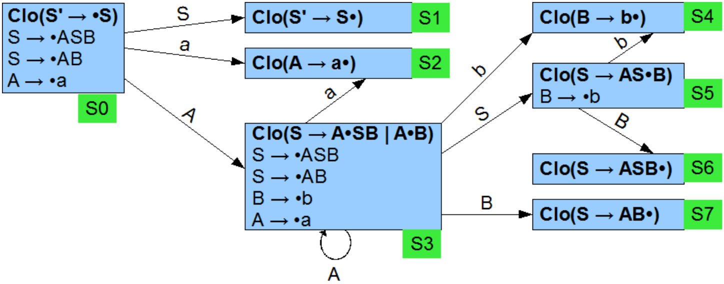


DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

[A b] $\Rightarrow$ S4; [A a] $\Rightarrow$ S2; [A B]

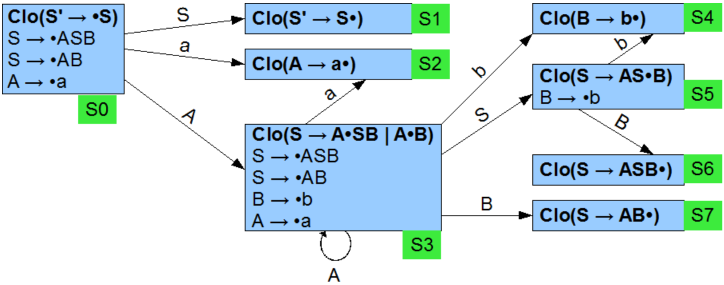


DR-automata

Gist

- We keep states-track on the stack in every step
- We need the state only while reducing to **determine the final state**
- The β-table is potentially large and is used only after a reduce
- Let's **dismiss** both table and trace and determine state from stack

[A b] $\Rightarrow$ S4; [A a] $\Rightarrow$ S2; [A B] $\Rightarrow$ S5;



Simple approach - problems

- The whole stack is searched, which can be very expensive
- Searching the stack from the top is much more convenient

Simple approach - problems

- The whole stack is searched, which can be very expensive
- Searching the stack from the top is much more convenient

DR-state

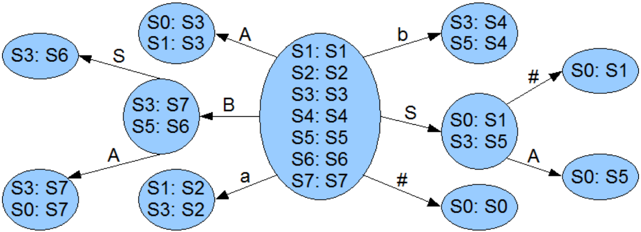
- A set of pairs $S_A : S_B$ with the meaning that a transition $S_A \rightarrow S_B$ is possible by some set of symbols
- Through transitions by stack items, we will try to reduce the set so that only one state appears on the right side = the top state

Simple approach - problems

- The whole stack is searched, which can be very expensive
- Searching the stack from the top is much more convenient

DR-state

- A set of pairs $S_A : S_B$ with the meaning that a transition $S_A \to S_B$ is possible by some set of symbols
- Through transitions by stack items, we will try to reduce the set so that only one state appears on the right side = the top state



Summary

DR-automaton

- From initial state, $|(N \cup T)|$ arrows fan out, but the latter states have significantly less amount of transitions
- Is created on the basis of the LR-automaton states and transitions
- Is generally smaller (in memory) but slower (in performence) then GOTO table

DR-parsing

- Is generally a bit slower (in performance), but smaller (in memory) then LR
- Doesn't need the states to be tracked on the stack
- Uses a DR-automaton instead of a GOTO table to determine final state after reduction

Right Context

Definition, attributes

- Set of terminal strings that can appear in the input
- Bound to **items**
- No longer regular, rather **context-free**
- Dot right context $F_1\{I\}$ vs. item right context $D_1\{I\}$

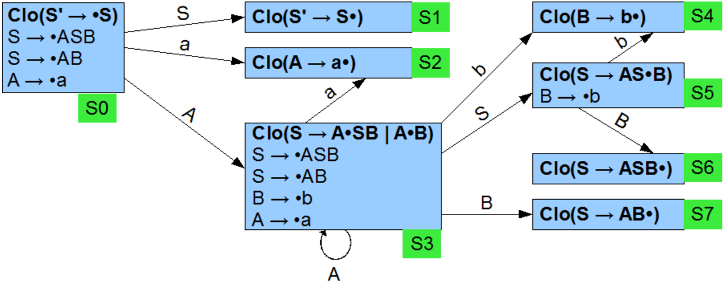
- $S \to aBb$
- $S \to aCc$
- $B, C \to \epsilon$

- $F_x\{B \to \circ\} = \{b\}$
- $F_x\{C \to \circ\} = \{c\}$

Computing right contexts

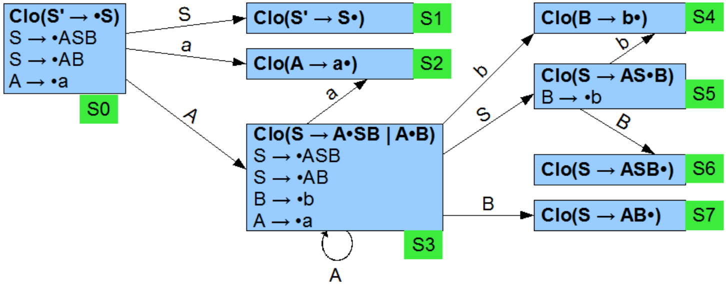
- Item RC
  - $F_x\{A \to \circ\alpha\} \to \gamma F_x\{X \to \beta \circ A\gamma\}$
  - $F_x\{A \to \alpha t \circ \beta\} \to F_y\{A \to \alpha \circ t\beta\}$
- Dot RC
  - $D_x\{A \to \alpha \circ \beta\} \to \beta F_x\{A \to \alpha \circ \beta\}$

Right context - example



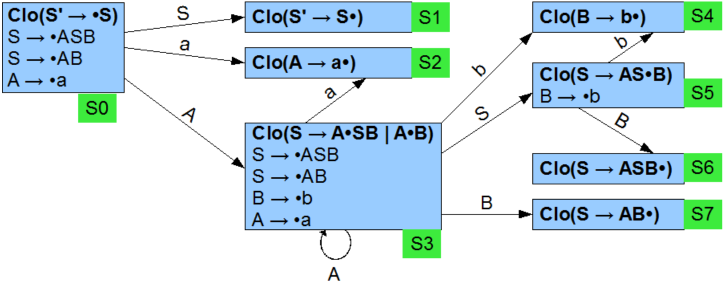
- $F_{S0}\{S' \rightarrow \circ S\} = \epsilon$

Right context - example



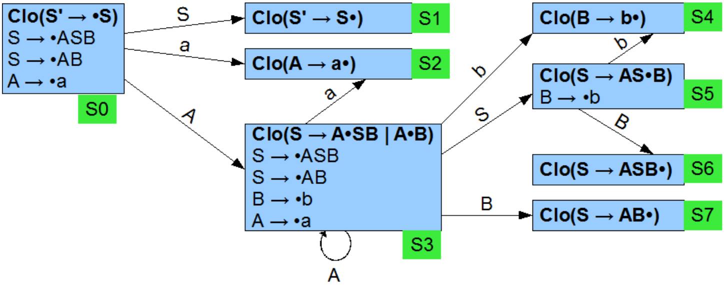
- $F_{S0}\{S' \to \circ S\} = \epsilon$
- $F_{S0}\{S \to \circ ASB\} = F_{S0}\{S' \to \circ S\}$
- $F_{S0}\{S \to \circ AB\} = F_{S0}\{S' \to \circ S\}$
- $F_{S0}\{A \to \circ a\} = B\ F_{S0}\{S \to \circ AB\}$
- $F_{S0}\{A \to \circ a\} = SB\ F_{S0}\{S \to \circ ASB\}$

Right context - example



- $F_{S0}\{S' \to \circ S\} = \epsilon$
- $F_{S0}\{S \to \circ ASB\} = F_{S0}\{S' \to \circ S\}$
- $F_{S0}\{S \to \circ AB\} = F_{S0}\{S' \to \circ S\}$
- $F_{S0}\{A \to \circ a\} = B\ F_{S0}\{S \to \circ AB\}$
- $F_{S0}\{A \to \circ a\} = SB\ F_{S0}\{S \to \circ ASB\}$
- $F_{S1}\{S' \to S\circ\} = F_{S0}\{S' \to \circ S\}$
- $F_{S2}\{A \to a\circ\} = F_{S0}\{A \to \circ A\}$

Right context - example



- $F_{S0}\{S' \to \circ S\} = \epsilon$
- $F_{S0}\{S \to \circ ASB\} = F_{S0}\{S' \to \circ S\}$
- $F_{S0}\{S \to \circ AB\} = F_{S0}\{S' \to \circ S\}$
- $F_{S0}\{A \to \circ a\} = B\ F_{S0}\{S \to \circ AB\}$
- $F_{S0}\{A \to \circ a\} = SB\ F_{S0}\{S \to \circ ASB\}$
- $F_{S1}\{S' \to S\circ\} = F_{S0}\{S' \to \circ S\}$
- $F_{S2}\{A \to a\circ\} = F_{S0}\{A \to \circ A\}$
- $D_{S0}\{S' \to \circ S\} = S\ F_{S0}\{S' \to \circ S\}$
- $D_{S0}\{S \to \circ ASB\} = ASB\ F_{S0}\{S \to \circ ASB\}$

LR-Regular parsing

Right contexts

- Right contexts = super look-ahead
- May be helpful when dealing with inadequate states
- However, intersection of CFG's is undecidable
- Checking the rest of the input against CFG makes no sense

LR-Regular parsing

Right contexts

- Right contexts = super look-ahead
- May be helpful when dealing with inadequate states
- However, intersection of CFG's is undecidable
- Checking the rest of the input against CFG makes no sense

LR-Regular

- Uses **regular envelopes** of the right contexts
- Approximation "from above"
- For non-terminals only
- Many heuristics and techniques possible, result is not guaranteed

Thank you for your attention.
Questions?