

# Computational Linguistics II: Parsing

## *Overview, Left-Recursion, Bottom-up Parsing*

Frank Richter & Jan-Philipp Söhn

`fr@sfs.uni-tuebingen.de, jp.soehn@uni-tuebingen.de`

# The Big Picture

hierarchy	grammar	machine	other
type 3	reg. grammar	DFA	reg. expressions
det. cf.	LR(k) grammar	DPDA	
type 2	CFG	PDA	
type 1	CSG	LBA	
type 0	unrestricted grammar	Turing machine	

DFA: Deterministic finite state automaton

(D)PDA: (Deterministic) Pushdown automaton

CFG: Context-free grammar

CSG: Context-sensitive grammar

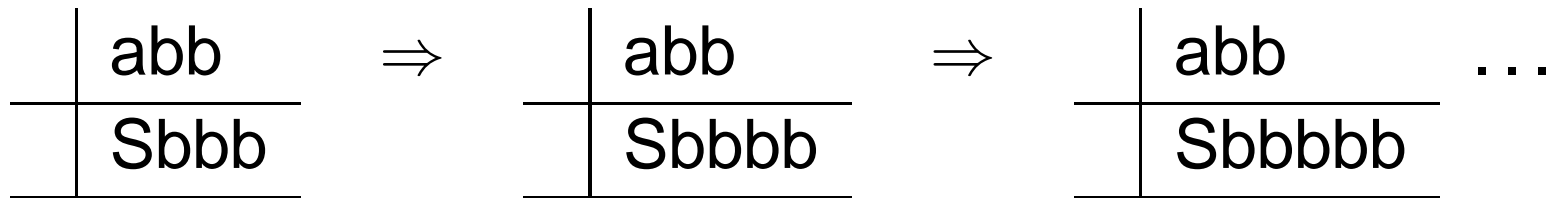
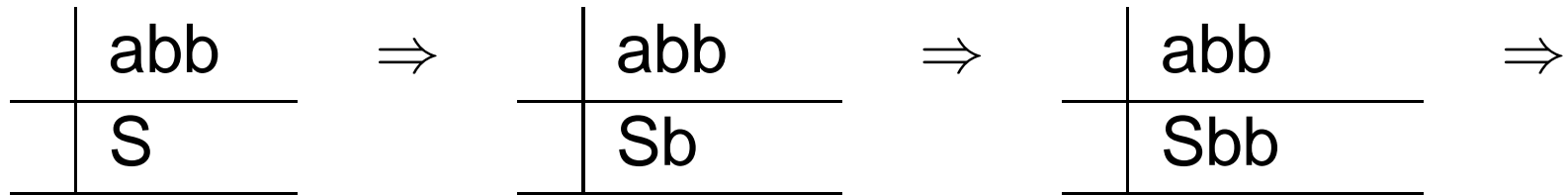
LBA: Linear bounded automaton

# Problem: Left-Recursion

- Grammar:  $S \rightarrow Sb \mid a$

# Problem: Left-Recursion

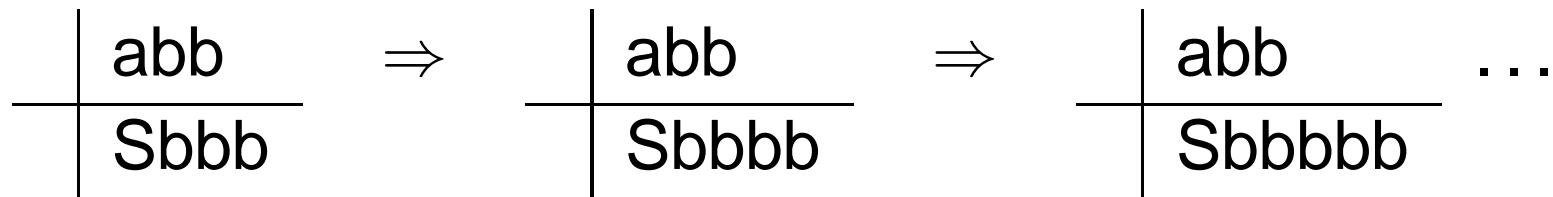
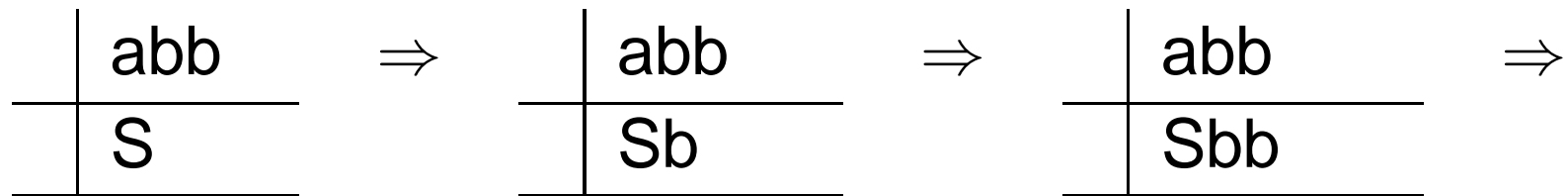
- Grammar:  $S \rightarrow Sb \mid a$
- Derivation for “abb”:



# Problem: Left-Recursion

- Grammar:  $S \rightarrow Sb \mid a$

- Derivation for “abb”:



- Problem: will never get around to finding a terminal in first position which it can match against the input  $\Rightarrow$  infinite loop

# Left-Recursion

- What exactly is left recursion?

# Left-Recursion

- What exactly is left recursion?
- Two cases:

# Left-Recursion

- What exactly is left recursion?
- Two cases:
  1. **immediate left recursion:**  $A \rightarrow A \alpha$   
lefthand side symbol is the same as **first** righthand side symbol



# Left-Recursion

- What exactly is left recursion?
- Two cases:
  1. **immediate left recursion:**  $A \rightarrow A \alpha$   
lefthand side symbol is the same as **first** righthand side symbol
  2. **indirect left recursion:**  $A \rightarrow B \alpha \rightarrow \dots \rightarrow A \beta$   
A extends via intermediate steps into another derivation part that **starts** with A.

# Left-Recursion

- What exactly is left recursion?
- Two cases:
  1. **immediate left recursion:**  $A \rightarrow A \alpha$   
lefthand side symbol is the same as **first** righthand side symbol
  2. **indirect left recursion:**  $A \rightarrow B \alpha \rightarrow \dots \rightarrow A \beta$   
A extends via intermediate steps into another derivation part that **starts** with A.
- Only *first, non-terminal* righthand side symbols are problematic.

# Eliminating Immediate Left-Recursion

- rule again:  $S \rightarrow Sb \mid a$ 
  - $\Rightarrow$  a, ab, abb, abbb
  - $\Rightarrow$   $a(b)^*$

# Eliminating Immediate Left-Recursion

- rule again:  $S \rightarrow Sb \mid a$ 
  - $\Rightarrow a, ab, abb, abbb$
  - $\Rightarrow a(b)^*$
- i.e. one part **in front** which ends the recursion, and the recursive part **at the end**

# Eliminating Immediate Left-Recursion

- rule again:  $S \rightarrow Sb \mid a$   
 $\Rightarrow a, ab, abb, abbb$   
 $\Rightarrow a(b)^*$
- i.e. one part **in front** which ends the recursion, and the recursive part **at the end**
- idea: reformulated rule:  $S \rightarrow a X \mid a$   
 $X \rightarrow b X \mid b$

# Eliminating Immediate Left-Recursion

- rule again:  $S \rightarrow Sb \mid a$   
 $\Rightarrow a, ab, abb, abbb$   
 $\Rightarrow a(b)^*$
- i.e. one part **in front** which ends the recursion, and the recursive part **at the end**
- idea: reformulated rule:  $S \rightarrow a X \mid a$   
 $X \rightarrow b X \mid b$
- more complicated ex.:  $S \rightarrow Sb \mid Sd \mid ac \mid e$

# Eliminating Immediate Left-Recursion

- rule again:  $S \rightarrow Sb \mid a$   
 $\Rightarrow a, ab, abb, abbb$   
 $\Rightarrow a(b)^*$
- i.e. one part **in front** which ends the recursion, and the recursive part **at the end**
- idea: reformulated rule:  $S \rightarrow a X \mid a$   
 $X \rightarrow b X \mid b$
- more complicated ex.:  $S \rightarrow Sb \mid Sd \mid ac \mid e$
- language:  $ac, acb, acbb, acbbb, acbd, acdb, e, eb, ebb, ebd, edb, \dots \Rightarrow (ac \mid e) (b \mid d)^*$

# Eliminating Immediate Left-Recursion

- rule again:  $S \rightarrow Sb \mid a$   
 $\Rightarrow a, ab, abb, abbb$   
 $\Rightarrow a(b)^*$
- i.e. one part **in front** which ends the recursion, and the recursive part **at the end**
- idea: reformulated rule:  $S \rightarrow aX \mid a$   
 $X \rightarrow bX \mid b$
- more complicated ex.:  $S \rightarrow Sb \mid Sd \mid ac \mid e$
- language:  $ac, acb, acbb, acbbb, acbd, acdb, e, eb, ebb, ebd, edb, \dots \Rightarrow (ac \mid e)(b \mid d)^*$
- reformulated:  
 $S \rightarrow acX \mid eX \mid ac \mid e; X \rightarrow bX \mid dX \mid b \mid d$



# Eliminating Immediate Left-Recursion

- rule form:  $A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$

# Eliminating Immediate Left-Recursion

- rule form:  $A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$
- **A\_head**: righthand sides that are not left-recursive:  
betas  
 $A\_head \rightarrow \beta_1 \mid \dots \mid \beta_m$

# Eliminating Immediate Left-Recursion

- rule form:  $A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$
- **A\_head**: righthand sides that are not left-recursive:  
betas  
 $A\_head \rightarrow \beta_1 \mid \dots \mid \beta_m$
- **A\_tail**: righthand sides to be duplicated: alphas  
 $A\_tail \rightarrow \alpha_1 \mid \dots \mid \alpha_n$

# Eliminating Immediate Left-Recursion

- rule form:  $A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$
- **A\_head**: righthand sides that are not left-recursive:  
betas  
 $A\_head \rightarrow \beta_1 \mid \dots \mid \beta_m$
- **A\_tail**: righthand sides to be duplicated: alphas  
 $A\_tail \rightarrow \alpha_1 \mid \dots \mid \alpha_n$
- **A\_tails**: recursion over alphas:  
 $A\_tails \rightarrow A\_tail \mid A\_tail A\_tails$

# Eliminating Immediate Left-Recursion

- rule form:  $A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$
- **A\_head**: righthand sides that are not left-recursive:  
betas  
 $A\_head \rightarrow \beta_1 \mid \dots \mid \beta_m$
- **A\_tail**: righthand sides to be duplicated: alphas  
 $A\_tail \rightarrow \alpha_1 \mid \dots \mid \alpha_n$
- **A\_tails**: recursion over alphas:  
 $A\_tails \rightarrow A\_tail \mid A\_tail A\_tails$
- **A**: puts it together:  
 $A \rightarrow A\_head \mid A\_head A\_tails$

# Immediate Left-Recursion

- left-recursive rule:

$$A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$$

# Immediate Left-Recursion

- left-recursive rule:

$$A \rightarrow A \alpha_1 \mid \dots \mid A \alpha_n \mid \beta_1 \mid \dots \mid \beta_m$$

- non-left-recursive rules:

$$A \rightarrow A\_head \mid A\_head A\_tails$$

$$A\_head \rightarrow \beta_1 \mid \dots \mid \beta_m$$

$$A\_tails \rightarrow A\_tail \mid A\_tail A\_tails$$

$$A\_tail \rightarrow \alpha_1 \mid \dots \mid \alpha_n$$

# Eliminating Indirect Left-Recursion

• grammar:

$S \rightarrow A B$

$A \rightarrow C B \mid b$

$C \rightarrow S a$

$B \rightarrow b$



# Eliminating Indirect Left-Recursion

• grammar:

$$S \rightarrow A B$$
$$A \rightarrow C B \mid b$$
$$C \rightarrow S a$$
$$B \rightarrow b$$

• derivation:  $S \Rightarrow A B \Rightarrow C B B \Rightarrow S a B B \Rightarrow A B a B B$   
 $\Rightarrow C B B a B B \Rightarrow S a B B a B B \Rightarrow A B a B B a B B \Rightarrow$   
 $\dots \Rightarrow bb(abb)^*$

# Eliminating Indirect Left-Recursion

• grammar:

$$S \rightarrow A B$$
$$A \rightarrow C B \mid b$$
$$C \rightarrow S a$$
$$B \rightarrow b$$

• derivation:  $S \Rightarrow A B \Rightarrow C B B \Rightarrow S a B B \Rightarrow A B a B B$   
 $\Rightarrow C B B a B B \Rightarrow S a B B a B B \Rightarrow A B a B B a B B \Rightarrow$   
 $\dots \Rightarrow bb(abb)^*$

• idea: move left recursion closer, until it is in the same rule

# Indirect Left-Recursion (2)

• steps:

# Indirect Left-Recursion (2)

• steps:

1. enumerate all non-terminals:  $A_1, A_2, \dots, A_n$

# Indirect Left-Recursion (2)

● steps:

1. enumerate all non-terminals:  $A_1, A_2, \dots, A_n$
2. check for  $A_1$  whether it is immediately left-recursive, eliminate left-recursion

# Indirect Left-Recursion (2)

• steps:

1. enumerate all non-terminals:  $A_1, A_2, \dots, A_n$
2. check for  $A_1$  whether it is immediately left-recursive, eliminate left-recursion
3. check for  $A_2$  whether it has a rule:  $A_2 \rightarrow A_1 \alpha$

# Indirect Left-Recursion (2)

● steps:

1. enumerate all non-terminals:  $A_1, A_2, \dots, A_n$
2. check for  $A_1$  whether it is immediately left-recursive, eliminate left-recursion
3. check for  $A_2$  whether it has a rule:  $A_2 \rightarrow A_1 \alpha$
4. yes? if  $A_1 \rightarrow \beta_1 \mid \dots \mid \beta_m$ :  
 $A_2 \rightarrow \beta_1 \alpha \mid \dots \mid \beta_m \alpha$

# Indirect Left-Recursion (2)

• steps:

1. enumerate all non-terminals:  $A_1, A_2, \dots, A_n$
2. check for  $A_1$  whether it is immediately left-recursive, eliminate left-recursion
3. check for  $A_2$  whether it has a rule:  $A_2 \rightarrow A_1 \alpha$
4. yes? if  $A_1 \rightarrow \beta_1 \mid \dots \mid \beta_m$ :  
 $A_2 \rightarrow \beta_1 \alpha \mid \dots \mid \beta_m \alpha$
5. check for  $A_n$  whether it has rules:  
 $A_n \rightarrow A_1 \alpha$   
...  
 $A_n \rightarrow A_n \alpha$



# Indirect Left-Recursion (2)

• steps:

1. enumerate all non-terminals:  $A_1, A_2, \dots, A_n$
2. check for  $A_1$  whether it is immediately left-recursive, eliminate left-recursion
3. check for  $A_2$  whether it has a rule:  $A_2 \rightarrow A_1 \alpha$
4. yes? if  $A_1 \rightarrow \beta_1 \mid \dots \mid \beta_m$ :  
 $A_2 \rightarrow \beta_1 \alpha \mid \dots \mid \beta_m \alpha$
5. check for  $A_n$  whether it has rules:  
 $A_n \rightarrow A_1 \alpha$   
...  
 $A_n \rightarrow A_n \alpha$
6. yes? eliminate according to step 4

# Indirect Left-Recursion – Example

• enumerated grammar:

S1  $\rightarrow$  A B

A2  $\rightarrow$  C B | b

C3  $\rightarrow$  S a

B4  $\rightarrow$  b

# Indirect Left-Recursion – Example

- enumerated grammar:

$S1 \rightarrow A B$

$A2 \rightarrow C B \mid b$

$C3 \rightarrow S a$

$B4 \rightarrow b$

- 1: is S immediately left-recursive? no.

# Indirect Left-Recursion – Example

- enumerated grammar:

$S1 \rightarrow A B$

$A2 \rightarrow C B \mid b$

$C3 \rightarrow S a$

$B4 \rightarrow b$

- 1: is S immediately left-recursive? no.
- 2  $\rightarrow$  1: rule  $A \rightarrow S a$ ? no.

# Indirect Left-Recursion – Example

- enumerated grammar:

$S1 \rightarrow A B$

$A2 \rightarrow C B \mid b$

$C3 \rightarrow S a$

$B4 \rightarrow b$

- $1$ : is  $S$  immediately left-recursive? no.
- $2 \rightarrow 1$ : rule  $A \rightarrow S a$ ? no.
- $2 \rightarrow 2$ : is  $A$  immediately left-recursive? no.

# Indirect Left-Recursion – Example

- enumerated grammar:

$S_1 \rightarrow A B$

$A_2 \rightarrow C B \mid b$

$C_3 \rightarrow S a$

$B_4 \rightarrow b$

- 1: is S immediately left-recursive? no.
- 2 → 1: rule  $A \rightarrow S \alpha$ ? no.
- 2 → 2: is A immediately left-recursive? no.
- 3 → 1: rule  $C \rightarrow S \alpha$ ? yes:  $C_3 \rightarrow S_1 a$

# Indirect Left-Recursion – Example

- enumerated grammar:

$S1 \rightarrow A B$

$A2 \rightarrow C B \mid b$

$C3 \rightarrow S a$

$B4 \rightarrow b$

- $1$ : is  $S$  immediately left-recursive? no.
- $2 \rightarrow 1$ : rule  $A \rightarrow S \alpha$ ? no.
- $2 \rightarrow 2$ : is  $A$  immediately left-recursive? no.
- $3 \rightarrow 1$ : rule  $C \rightarrow S \alpha$ ? yes:  $C3 \rightarrow S1 a$
- replace by:  $C \rightarrow A B a$

# Indirect Left-Recursion – Example

- $3 \rightarrow 2$ : rule  $C \rightarrow A \alpha$ ? yes (new rule!):  
 $C3 \rightarrow A2 B a$



# Indirect Left-Recursion – Example

- $3 \rightarrow 2$ : rule  $C \rightarrow A \alpha$ ? yes (new rule!):  
 $C3 \rightarrow A2 B a$
- replace by:  $C \rightarrow C B B a \mid b B a$

# Indirect Left-Recursion – Example

- $3 \rightarrow 2$ : rule  $C \rightarrow A \alpha?$  yes (new rule!):  
 $C3 \rightarrow A2 B a$
- replace by:  $C \rightarrow C B B a \mid b B a$
- $3 \rightarrow 3$  (immediate left recursion): rule  $C \rightarrow C \alpha?$  yes  
(new rule!):  $C3 \rightarrow C3 B B a \mid b B a$

# Indirect Left-Recursion – Example

- $3 \rightarrow 2$ : rule  $C \rightarrow A \alpha?$  yes (new rule!):  
 $C_3 \rightarrow A_2 B a$
- replace by:  $C \rightarrow C B B a \mid b B a$
- $3 \rightarrow 3$  (immediate left recursion): rule  $C \rightarrow C \alpha?$  yes (new rule!):  $C_3 \rightarrow C_3 B B a \mid b B a$
- replace by:  
 $C \rightarrow C\_head \mid C\_head C\_tails$   
 $C\_head \rightarrow b B a$   
 $C\_tails \rightarrow C\_tail \mid C\_tail C\_tails$   
 $C\_tail \rightarrow a B B$