◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 - のへで

1/32

## Contexts of parsing decisions

Ondřej Navrátil

FIT BUT

7. prosince 2012

		Right Context	LR-regular
Outline			

1 Left context

- 2 DR-automata
- **3** Summary
- 4 Right Context
- 5 LR-regular

## 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')
  - $\blacksquare N = \{S', S, A, B\}$

• 
$$T = \{a, b\}$$

 $\blacksquare P = \{S' \rightarrow S, S \rightarrow ASB | AB, A \rightarrow a, B \rightarrow b\}$ 

•  $L(G) = a^n b^n$ 







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

- - - - -



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

 $\blacksquare P_0 \to \epsilon$ 



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

$$P_0 \to \epsilon$$
$$P_1 \to P_0 S$$



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

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



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

 $P_{0} \rightarrow \epsilon$   $P_{1} \rightarrow P_{0}S$   $P_{2} \rightarrow P_{0}a$   $P_{2} \rightarrow P_{3}a$   $P_{2} \rightarrow P_{3}a$   $P_{2} \rightarrow P_{3}a$   $P_{2} \rightarrow P_{3}a$   $P_{2} \rightarrow P_{3}B$ 



#### 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		Right Context	LR-regular
DR	-automata				
ſ	Gist				
	We keep	states-track c	on the stack in ev	very step	
	We need	the state only	while reducing t	to determine the final	state
	<b>The</b> $\beta$ -tal	ble is potentio	onally large and i	s used only after a redu	JCe
	Let's disr	<b>niss</b> both tab	le and trace and	determine state from s	tack
	$Cio(S' \rightarrow \bullet S)$ $S \rightarrow \bullet ASB$		Clo(S' → S•) S1	$\frac{\text{Clo}(B \to b^{\bullet})}{\sqrt{2}}$	54
	S → •AB		Clo(A → a•) S2	$\nabla$ Clo(S $\rightarrow$ AS•B)	



	DR-auto	omata		Right Context	LR-regular
DR-aı	itomata				
Gist [A	We keep state We need the s The $\beta$ -table is Let's <b>dismiss</b>	s-track on the state only whi potentionally both table an	e stack in eve le reducing to r large and is d trace and o	ery step o <b>determine th</b> used only after letermine state	<b>ne final state</b> <sup>•</sup> a reduce from stack
	$ \begin{array}{c} \textbf{Clo(S' \rightarrow \bullet S)} \\ \textbf{S} \rightarrow \bullet \textbf{ASB} \\ \textbf{S} \rightarrow \bullet \textbf{AB} \\ \textbf{A} \rightarrow \bullet \textbf{a} \end{array} $	S Clo(S' a Clo(A S $\rightarrow$ · A S $\rightarrow$ · A B $\rightarrow$ · b A $\rightarrow$ · a	→ $S \cdot$ ) S1 → $a \cdot$ ) S2 $a \cdot$ → $A \cdot SB   A \cdot B$ ) SB B $A \cdot S3 A$	$Clo(B \rightarrow Clo(S \rightarrow A))$ $Clo(S \rightarrow A)$ $Clo(S \rightarrow A)$ $Clo(S \rightarrow A)$	b•)     S4       AS•B)     S5       ASB•)     S6       AB•)     S7

・ロト・(中下・(中下・(日下・(日下)))

	DR-automata		Right Context	LR-regular
DR-a	automata			
G	ist ■ We keep states-track @ ■ We need the state onl	on the stack in ev y while reducing t	ery step o <b>determine the final s</b>	tate
[A	<ul> <li>The β-table is potenti</li> <li>Let's dismiss both tab</li> <li>Δ] ⇒ S4;</li> </ul>	onally large and is ble and trace and	used only after a reduce determine state from sta	: ck
	$ \begin{array}{c} \text{Clo(S' \rightarrow \bullet \text{S}) \\ \text{S} \rightarrow \bullet \text{ASB} \\ \text{S} \rightarrow \bullet \text{AB} \\ \text{A} \rightarrow \bullet a \end{array} $	$\begin{array}{c} \text{Clo}(\text{S}^{\text{`}} \rightarrow \text{S}^{\text{`}}) & \text{S1} \\ \hline \\ \text{Clo}(\text{A} \rightarrow \text{a}^{\text{`}}) & \text{S2} \\ \hline \\ \text{S} & \text{Clo}(\text{S} \rightarrow \text{A}^{\text{`}}\text{SB} \mid \text{A}^{\text{`}}\text{B}) \\ \text{S} \rightarrow \text{A}\text{SB} \\ \text{S} \rightarrow \text{A}\text{AB} \\ \text{B} \rightarrow \text{\cdot}\text{b} \\ \text{A} \rightarrow \text{\cdot}\text{a} \end{array}$	$Clo(B \rightarrow b \bullet) \qquad S4$ $0 \qquad \qquad$	

S3

Ā

★ロト ★御 と ★ 注 と ★ 注 と … 注

	DR-automata		Right Context	LR-regular
DR-autom	nata			
Gist				
■ We I ■ We I	keep states-track o need the state only	n the stack in ev while reducing	very step to <b>determine the fina</b>	l state
<ul><li>The</li><li>Let's</li></ul>	$\beta$ -table is potentic <b>dismiss</b> both tab	onally large and i le and trace and	s used only after a redu determine state from s	uce stack
$[A b] \Rightarrow \vdots$	S4; [A a]			
$\begin{array}{c} Clo(S \\ S \rightarrow \cdot \\ S \rightarrow \cdot \\ A \rightarrow \end{array}$	ASB AB a SO AB a C SS AB a C SS AB a C SS ASB AB a C SS S C S C S C S C S C S C S C S C	$ \begin{array}{c} \text{Clo}(\text{S}' \to \text{S} \cdot) & \text{S1} \\ \hline \text{Clo}(\text{A} \to \text{a} \cdot) & \text{S2} \\ & & & & \\ & & & \\ \hline \text{Clo}(\text{S} \to \text{A} \cdot \text{SB} \mid \text{A} \cdot \text{B}) \\ & & & & & \\ \hline \text{S} \to \cdot \text{ASB} \\ & & & & & \\ \hline \text{S} \to \cdot \text{AB} \\ & & & & & \\ \hline \text{A} \to \cdot \text{a} \\ \hline \end{array} $	$Clo(B \rightarrow b^{\bullet})$ $Clo(S \rightarrow AS^{\bullet}B)$ $B \rightarrow b$ $Clo(S \rightarrow ASB^{\bullet})$ $B$ $Clo(S \rightarrow AB^{\bullet})$	54 55 56 57

Ā

	ext Df	R-automata			Right Context	LR-regular
DR-	-automata					
	aatomata					
	Gist					
	We keep st	ates-track on	the stack in e	every ster	)	
	We need the	ne state only w	hile reducing	to <b>dete</b>	rmine the fina	l state
	■ The β-table	e is potentiona	ally large and	is used o	only after a red	uce
	Let's dismi	ss both table	and trace and	d determ	ine state from	stack
	[Δ b] → SA· [Δ	$al \rightarrow S2^{\circ}$				
I	[/ b] -> 54, [/	a] → 52,				
	$\frac{Clo(S' \to \cdotS)}{S \to \cdotASB}$	S Clo	$(S' \rightarrow S^{\bullet})$ S	1	Clo(B → b•)	S4
	$S \rightarrow AB$	a ► Clo	$(A \rightarrow a \bullet)$ S	2 💊	Clo(S → AS•B)	_
	SO	A	8	5	B→•b	S5
		Clo S –	(S → A•SB   A•B) • •ASB		~	_
		S –	•AB		$Clo(S \rightarrow ASB \cdot)$	S6
		D —	-0	в.		67

A → •a

А

・ロト ・日本 ・モト ・モト

► Clo(S → AB•)

В

S3

**S**7

æ

Left conte	xt DR-automat	a Summa		Right Context	LR-regular
DR-	automata				
	Gist				
	We keep states-t	rack on the stack	c in every step	)	
	We need the state	te only while redu	icing to <b>dete</b> i	rmine the fina	al state
	• The $\beta$ -table is p	otentionally large	and is used o	only after a rec	luce
	Let's <b>dismiss</b> bo	th table and trace	e and determ	ine state from	stack
[	$[A b] \Rightarrow S4; [A a] \Rightarrow$	→ S2; [A B]			
	$\begin{array}{c} Clo(S' \to \cdotS) \\ S \to \cdotASB \end{array} \qquad $	► Clo(S' → S•)	S1	Clo(B → b•)	S4
	$S \rightarrow \bullet AB$ $A \rightarrow \bullet a$	$  Clo(A \rightarrow a) $	S2 v	$\begin{array}{c} Clo(S \to AS \bullet B) \\ B \to \bullet b \end{array}$	S5
	SO	Clo(S → A•SB	(A•B) S	B	
		$S \rightarrow \cdot ASB$ $S \rightarrow \cdot AB$		$Clo(S \rightarrow ASB^{\bullet})$	<b>S6</b>
		$ \begin{array}{c} B \to \bullet b \\ A \to \bullet a \end{array} $	В	Clo(S → AB•)	S7

★ロト ★御 と ★ 注 と ★ 注 と … 注 990 19 / 32

S3

Α

		DR-automata		Right Context	LR-regular
DR	l-automata				
	Gist				
	We keep s	states-track or	the stack in eve	ery step	
	We need t	the state only	while reducing to	determine the fina	state
	■ The β-tab	le is potention	nally large and is	used only after a redu	JCe
	Let's disn	<b>niss</b> both table	e and trace and o	letermine state from s	tack
	$[A b] \Rightarrow S4; [$	A a] $\Rightarrow$ S2; [A	A B] $\Rightarrow$ S5;		_
	$\frac{Clo(S' \to \bulletS)}{S \to \bulletASB}$	S CI	<b>o(S'</b> → <b>S•</b> ) S1	Clo(B → b•)	34



E ∽ Q (~ 20 / 32

#### 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  $S_A \rightarrow S_B$  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



	DR-automata	Summary	Right Context	LR-regular
Summarv				

#### 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 performance) 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

Left context	DR-automata	Summary	Right Context	LR-regular
<b>Right</b> Cor	itext			

#### 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  $D_1{I}$
- $S \rightarrow aBb$   $F_x\{B \rightarrow \circ\} = \{b\}$
- S 
  ightarrow aCc
- $\blacksquare B, C \to \epsilon$

$$F_{X}\{D \to 0\} = \{D\}$$

• 
$$F_x \{ C \rightarrow \circ \} = \{ c \}$$

#### 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

 $\square D_x \{ A \to \alpha \circ \beta \} \to \beta F_x \{ A \to \alpha \circ \beta \}$ 

う Q (や 25 / 32

・ロト ・回ト ・ヨト ・ヨト

æ

26 / 32

### Right context - example



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

## Right context - example



• 
$$F_{50}{S' \to \circ S} = \epsilon$$
  
•  $F_{50}{S \to \circ ASB} = F_{50}{S' \to \circ S}$   
•  $F_{50}{S \to \circ AB} = F_{50}{S' \to \circ S}$   
•  $F_{50}{A \to \circ a} = B F_{50}{S \to \circ AB}$   
•  $F_{50}{A \to \circ a} = SB F_{50}{S \to \circ ASB}$ 

◆ロ → < □ → < Ξ → < Ξ → < Ξ → ○ Q (~ 27/32

イロン 不同と 不同と 不同と

æ

28 / 32

## Right context - example



■ 
$$F_{50}{S' \to \circ S} = \epsilon$$
  
■  $F_{50}{S \to \circ ASB} = F_{50}{S' \to \circ S}$   
■  $F_{50}{S \to \circ AB} = F_{50}{S' \to \circ S}$   
■  $F_{50}{A \to \circ a} = B F_{50}{S \to \circ AB}$   
■  $F_{50}{A \to \circ a} = SB F_{50}{S \to \circ ASB}$   
■  $F_{51}{S' \to S\circ} = F_{50}{S' \to \circ S}$   
■  $F_{52}{A \to a\circ} = F_{50}{A \to \circ A}$ 

## Right context - example



■ 
$$F_{S0}{S' \to \circ S} = \epsilon$$
  
■  $F_{S0}{S \to \circ ASB} = F_{S0}{S' \to \circ S}$   
■  $F_{S0}{S \to \circ AB} = F_{S0}{S' \to \circ S}$   
■  $F_{S0}{A \to \circ a} = B F_{S0}{S \to \circ AB}$   
■  $F_{S0}{A \to \circ a} = SB F_{S0}{S \to \circ ASB}$   
■  $F_{S1}{S' \to S_0} = F_{S0}{S' \to \circ S}$   
■  $F_{S2}{A \to a_0} = F_{S0}{A \to \circ A}$   
■  $D_{S0}{S' \to \circ S} = S F_{S0}{S' \to \circ S}$   
■  $D_{S0}{S \to \circ ASB} = ASB F_{S0}{S \to \circ ASB}$ 

◆ロ → < □ → < Ξ → < Ξ → < Ξ → ○ Q (~ 29/32

# 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

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへで

32 / 32

# Thank you for your attention. Questions?