



# Análise Sintáctica

Compiladores, Aula N<sup>o</sup> 18  
João M. P. Cardoso

1

Aula 18



## Construção do *Parser*

---

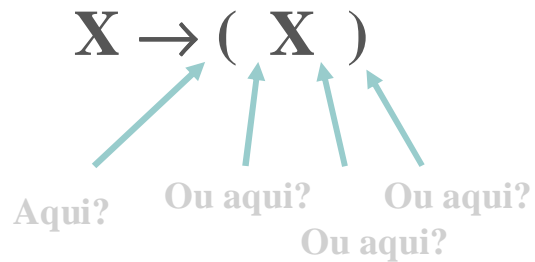
- Sintetizar um DFA
  - Captura todos os estados possíveis em que o *parser* pode estar
  - Transições de estados para terminais e não-terminais
- Utilizar o DFA para criar a tabela do *parser*

2

Aula 18

## Exemplo

- Estados do DFA baseados nos itens
  - Temos de capturar o que foi já percorrido numa produção



- Gramática
  - $S \rightarrow X \$$  (1)
  - $X \rightarrow (X)$  (2)
  - $X \rightarrow ( )$  (3)

3

Aula 18

## Exemplo

- Estados do DFA baseados nos itens
  - Temos de capturar o que foi já percorrido numa produção

$$X \rightarrow ( X )$$

- Gramática
  - $S \rightarrow X \$$  (1)
  - $X \rightarrow (X)$  (2)
  - $X \rightarrow ( )$  (3)
- Produção  $X \rightarrow (X)$  gera 4 itens:
  - $X \rightarrow \bullet (X)$
  - $X \rightarrow ( \bullet X)$
  - $X \rightarrow (X \bullet)$
  - $X \rightarrow (X) \bullet$

4

Aula 18



## Exemplo

---

- o Estados do DFA baseados nos itens
  - Temos de capturar o que foi já percorrido numa produção

Itens para todas as produções da Gramática:

o Gramática	$S \rightarrow X \$$	(1)	$S \rightarrow \bullet X \$$
	$X \rightarrow (X)$	(2)	$S \rightarrow X \bullet \$$
	$X \rightarrow ( )$	(3)	$X \rightarrow \bullet (X)$
			$X \rightarrow ( \bullet X )$
			$X \rightarrow (X \bullet )$
			$X \rightarrow ( \bullet )$
			$X \rightarrow ( \bullet )$
			$X \rightarrow ( ) \bullet$

5

Aula 18



## Ideia por trás dos itens

---

- o Estados correspondem a conjuntos de itens
- o Se um estado contém um item:  $A \rightarrow \alpha \bullet c \beta$ 
  - O *parser* espera uma eventual redução utilizando a produção:  $A \rightarrow \alpha c \beta$
  - O *parser* já analisou  $\alpha$
  - Espera que a entrada possa conter  $c$ , seguido de  $\beta$
- o Se um estado contém um item:  $A \rightarrow \alpha \bullet$ 
  - O *parser* já analisou  $\alpha$
  - Reduzirá utilizando:  $A \rightarrow \alpha$
- o Se um estado contém um item:  $S \rightarrow \alpha \bullet \$$  e a entrada está vazia
  - O *parser* aceita a entrada

Aula 18



## Relação entre itens e acções

---

- Se o estado corrente contém o item:  $A \rightarrow \alpha \bullet c \beta$  e o símbolo corrente na entrada é  $c$ 
  - O *parser* desloca  $c$  para a pilha de símbolos
  - O próximo estado conterá  $A \rightarrow \alpha c \bullet \beta$
- Se o estado corrente contém o item:  $A \rightarrow \alpha \bullet$ 
  - Reduzirá utilizando:  $A \rightarrow \alpha$
- Se o estado corrente contém o item:  $S \rightarrow \alpha \bullet \$$  e a entrada está vazia
  - O *parser* aceita a entrada

7

Aula 18



## Closure() de um conjunto de itens

---

- Closure (fecho) encontra todos os itens no mesmo estado
- Algoritmo de ponto-fixo para Closure(**I**)
  - Cada item em **I** é também um item em Closure(**I**)
  - Se  $A \rightarrow \alpha \bullet B \beta$  está em Closure(**I**) e  $B \rightarrow \bullet \gamma$  é um item, então adicionar  $B \rightarrow \bullet \gamma$  a Closure(**I**)
  - Repetir até que não haja adição de novos itens a Closure(**I**)

8

Aula 18

## Exemplos de Closure()

### o Closure( $\{X \rightarrow (\cdot X)\}$ )

$$\left\{ \begin{array}{l} X \rightarrow (\cdot X) \\ X \rightarrow \cdot (X) \\ X \rightarrow \cdot ( ) \end{array} \right\}$$

### o Closure( $\{S \rightarrow \cdot X \$\}$ )

$$\left\{ \begin{array}{l} S \rightarrow \cdot X \$ \\ X \rightarrow \cdot (X) \\ X \rightarrow \cdot ( ) \end{array} \right\}$$

### o Itens

$S \rightarrow \cdot X \$$

$S \rightarrow X \cdot \$$

$X \rightarrow \cdot (X)$

$X \rightarrow (\cdot X)$

$X \rightarrow (X \cdot)$

$X \rightarrow (X) \cdot$

$X \rightarrow \cdot ( )$

$X \rightarrow (\cdot )$

$X \rightarrow ( ) \cdot$

9

Aula 18

## Goto() de um conjunto de itens

- o Goto encontra o novo estado depois de consumir um símbolo da gramática enquanto no presente estado
- o Algoritmo para  $Goto(I, X)$ 
  - em que  $I$  é um conjunto de itens
  - e  $X$  um símbolo terminal ou não-terminal da gramática

$$Goto(I, X) = \text{Closure}(\{ A \rightarrow \alpha X \cdot \beta \mid A \rightarrow \alpha \cdot X \beta \text{ em } I \})$$

- o dá-nos o novo conjunto obtido pelo movimento do ponto sobre  $X$

10

Aula 18

## Exemplos de Goto()

o Goto( $\{X \rightarrow (\bullet X)\}, X$ )

$\{X \rightarrow (X \bullet)\}$

o Goto ( $\{X \rightarrow \bullet(X)\}, ()$ )

$\left\{ \begin{array}{l} X \rightarrow (\bullet X) \\ X \rightarrow \bullet (X) \\ X \rightarrow \bullet ( ) \end{array} \right\}$

o Itens

$S \rightarrow \bullet X \$$

$S \rightarrow X \bullet \$$

$X \rightarrow \bullet (X)$

$X \rightarrow (\bullet X)$

$X \rightarrow (X \bullet)$

$X \rightarrow (X) \bullet$

$X \rightarrow \bullet ( )$

$X \rightarrow ( \bullet )$

$X \rightarrow ( ) \bullet$

11

Aula 18

## Construir os estados do DFA

o Começar com o item:  $S \rightarrow \bullet \beta \$$

- Caso não exista adicionar a primeira produção com término \$
- Criar o primeiro estado como sendo  $\text{Closure}(\{ \text{Goal} \rightarrow \bullet S \$ \})$

o Escolher um estado  $I$

- Para cada item  $A \rightarrow \alpha \bullet X \beta$  em  $I$ 
  - determinar  $\text{Goto}(I, X)$
  - se  $\text{Goto}(I, X)$  não está no estado, criar um novo estado
  - Adicionar um laço  $X$  do estado  $I$  ao estado  $\text{Goto}(I, X)$

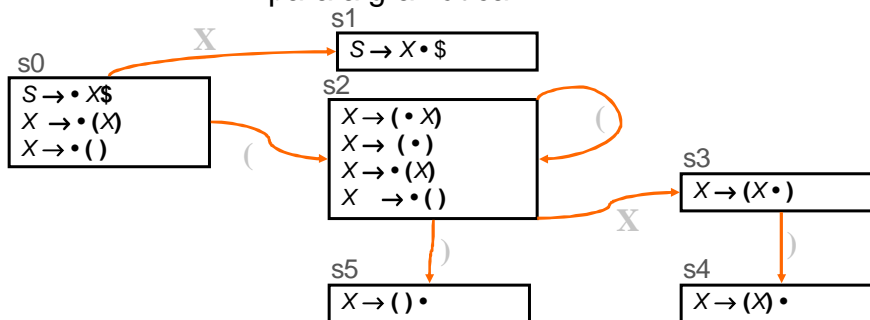
o Repetir até que não haja mais modificações possíveis

12

Aula 18

## Construir o *Parser*

- o Construir o DFA
  - DFA para a gramática:



- o Construir a tabela do *parser* utilizando o DFA

13

Aula 18

## Criação da Tabela do *Parser*

- o Para cada estado
  - Transição para outro estado utilizando um símbolo **terminal** é um **deslocamento** para esse estado (*shift to sn*)
  - Transição para outro estado utilizando um símbolo **não-terminal** é um **goto** para esse estado (*goto sn*)
  - Se existir um item  $A \rightarrow \alpha \cdot$  no estado fazer uma **redução** com essa produção para todos os terminais (*reduce k*)
  - Se existir um item  $S \rightarrow X \cdot \$$  no estado então colocar acção de aceitação para o terminal  $\$$

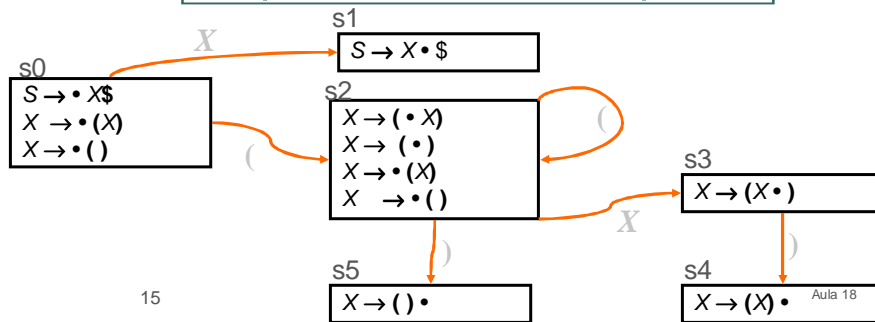
14

Aula 18



## Criação da Tabela do *Parser*

State	ACTION			Goto
	(	)	\$	X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	



15

Aula 18



## Problema Potencial

- o Nenhum *lookahead*
- o Vulnerável a conflitos desnecessários
  - Conflitos Shift/Reduce (pode reduzir demasiado cedo em alguns casos)
  - Conflitos Reduce/Reduce
- o Solução: *Lookahead*
  - Apenas para reduções – reduzir apenas quando o próximo símbolo pode ocorrer depois de não-terminal da produção
  - *Lookahead* sistemático, divisão de estados baseada no próximo símbolo, acção é sempre uma função do próximo símbolo
  - Pode ser generalizado para ver à frente múltiplos símbolos

16

Aula 18