

Gramáticas Regulares

1

EDUARDO FREIRE NAKAMURA

Instituto de Computação
Universidade Federal do Amazonas
nakamura@icomp.ufam.edu.br

¹Este material utiliza conteúdo das aulas fornecidas pelo Prof. Vilar da Câmara Neto (disponível em <http://http://prof.vilarneto.com>).

²Permissão de uso fornecida pelos autores.

³As figuras utilizadas neste material são de domínio público, disponíveis na Internet sem informações de direitos autorais.

Gramáticas

2

- Os métodos de formalização e as operações básicas (repetição, fecho de Kleene, união, etc.) são uma ferramenta poderosa para a representação de linguagens formais
- No entanto, essas ferramentas podem não ser muito adequadas para a construção de linguagens
 - Requerem experiência, intuição e inspiração
 - É fácil esquecer de certos detalhes (por exemplo, a presença ou ausência da palavra na linguagem)
 - Linguagens complexas são complicadas de se representar
 - É difícil de verificar se a descrição está 100% correta

Gramáticas

3

- **Gramáticas** são formalismos especialmente designados para facilitar a construção de linguagens
 - A descrição pode ser feita a partir dos elementos básicos da linguagem (ex.: comandos, expressões, literais, estruturas)
 - A descrição é quebrada em diversas regras, e não montada em uma única expressão difícil de se ler
 - Acrescentar novos elementos à linguagem em geral é simples
 - É bem mais fácil verificar se a formalização está correta

Como é a cara de uma gramática?

4

- Eis um exemplo de uma gramática simples, capaz de gerar números binários inteiros (positivos, negativos e zero):

$$\begin{array}{ll} P & \rightarrow SB \\ S & \rightarrow \lambda | - \\ B & \rightarrow 0B | 1B | 0 | 1 \end{array}$$

- Uma **sequência de derivação** começa com a sequência “ P ”
- Pode-se transformar qualquer pedaço da sequência atual em outros pedaços de acordo com as regras descritas
- Pode-se parar a qualquer momento quando a sequência só possui símbolos do alfabeto

Elementos de uma gramática

5

- Uma gramática constitui-se de três elementos principais:
 - Um conjunto de **terminais**, que são os símbolos que formam a linguagem (isto é, o alfabeto)
 - Um conjunto de **variáveis** ou **não terminais**, que são símbolos que não fazem parte da linguagem
 - Um conjunto de **regras**, que definem as transformações possíveis entre uma sequência de terminais e variáveis em outra sequência

Formas sentenciais

6

- Convenção adotada
 - As variáveis são letras maiúsculas (A , C , etc.)
 - Os demais símbolos são os terminais (a , b , etc.)
- Uma sequência de terminais e variáveis é chamada de forma sentencial
 - $ACabCb$
 - P
 - $aabbcc$
 - $AAcB$

Regras

7

- Cada **regra** estabelece possíveis transformações para trechos de formas sentenciais. São escritas na forma $seq_1 \rightarrow seq_2$
- Por exemplo, dada seguinte regra

$$A \rightarrow aB$$

- Isto é, ocorrências da variável A podem ser substituídas por aB
- Exemplos
 - $abAc$ pode ser transformada em $abaBc$
 - $AccA$ pode ser transformada em $aBccA$ ou em $AccaB$
 - $aBccA$, por sua vez, pode ser transformada em $aBccaB$
 - aB não pode ser transformada por esta regra

Regras

8

- Cada passo de transformação é chamado de **derivação** e é designado por $\textit{forma}_1 \Rightarrow \textit{forma}_2$
- Mantendo o exemplo anterior

$$A \rightarrow aB$$

- as seguintes derivações são válidas

$$abAc \Rightarrow abaBc$$

$$AccA \Rightarrow aBccA$$

$$AccA \Rightarrow AccaB$$

$$aBccA \Rightarrow aBcc aB$$

Regras

9

- Outros exemplos
 - $A \rightarrow \lambda$ (sequência nula)
 - $A \rightarrow aA$ (recursiva)
 - $A \rightarrow 0A \mid 1A \mid \lambda$ (múltiplas regras para a mesma sequência)
- O objetivo de uma gramática é partir de uma forma sentencial contendo apenas uma variável, chamada de **variável de partida**, e utilizar as regras em sequência para gerar **sentenças**
- Cada sentença que pode ser gerada por uma gramática é uma **palavra**
- A **linguagem gerada por uma gramática** é o conjunto de todas as palavras que podem ser geradas pela gramática

Formalização

10

- Uma gramática G é uma quádrupla (V, Σ, R, P) , onde
 - V é o conjunto de variáveis;
 - Σ é o conjunto de terminais (isto é, o alfabeto das palavras geradas)
 - R é o conjunto de regras
 - P *pertencente a* V é a variável de partida

Gramática regular

11

- Uma **Gramática Regular**, ou **GR**, é uma gramática que possui certas restrições na forma de suas regras
- Em uma GR, as regras podem ter apenas uma das formas
 - $A \rightarrow aB$
 - $A \rightarrow a$
 - $A \rightarrow \lambda$
- onde A e B são variáveis da gramática e $a \in \Sigma$
- Uma característica importante das GRs é que a qualquer momento **a forma sentencial possui no máximo uma variável**, e **esta se encontra sempre no fim** da forma sentencial

Exemplo

12

- Números binários inteiros não negativos

○ $P \rightarrow 0P \mid 1P \mid 0 \mid 1$

 ↑ ↑ ↑ ↑

 r1 r2 r3 r4

- Como gerar a palavra 0110?

Exemplo

13

- Números binários inteiros não negativos

○ $P \rightarrow 0P \mid 1P \mid 0 \mid 1$

↑ ↑ ↑ ↑
r1 r2 r3 r4

- Como gerar a palavra 0110?

$P \Rightarrow 0P \quad (r1)$
 $\Rightarrow 01P \quad (r2)$
 $\Rightarrow 011P \quad (r2)$
 $\Rightarrow 0110 \quad (r3)$

Exemplo

14

- Números binários inteiros não negativos

○ $P \rightarrow 0P \mid 1P \mid 0 \mid 1$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow$
 $r1 \quad r2 \quad r3 \quad r4$

- Como gerar a palavra 0110?

$P \Rightarrow 0P \quad (r1)$
 $\Rightarrow 01P \quad (r2)$
 $\Rightarrow 011P \quad (r2)$
 $\Rightarrow 0110 \quad (r3)$

Exemplo

15

- Números binários inteiros não negativos

○ $P \rightarrow 0P \mid 1P \mid 0 \mid 1$

 ↑ ↑ ↑ ↑
 r1 r2 r3 r4

- Como gerar a palavra 0110?

$P \Rightarrow 0P$ (r1)
 $\Rightarrow 01P$ (r2)
 $\Rightarrow 01\downarrow P$ (r2)
 $\Rightarrow 0110$ (r3)

Exemplo

16

- Números binários inteiros não negativos

○ $P \rightarrow 0P \mid 1P \mid 0 \mid 1$

 ↑ ↑ ↑ ↑
 r1 r2 r3 r4

- Como gerar a palavra 0110?

$P \Rightarrow 0P \quad (r1)$
 $\Rightarrow 01P \quad (r2)$
 $\Rightarrow 011P \quad (r2)$
 $\Rightarrow 0110 \quad (r3)$

 ↓
 r4

Exemplo

17

- Números binários inteiros não negativos

○ $P \rightarrow 0P \mid 1P \mid 0 \mid 1$

↑ ↑ ↑ ↑
r1 r2 r3 r4

- Como gerar a palavra 0110?
- Como gerar a palavra 1011?

$P \Rightarrow 0P$ (r1)
 $\Rightarrow 01P$ (r2)
 $\Rightarrow 011P$ (r2)
 $\Rightarrow 0110$ (r3)

$P \Rightarrow 1P$ (r2)
 $\Rightarrow 10P$ (r1)
 $\Rightarrow 101P$ (r2)
 $\Rightarrow 1011$ (r4)

Transformando GRs em AFNs

18

- Antes de mais nada, vejamos a intuição por trás da transformação de GRs em AFNs, revendo as formas possíveis de regras

- $A \rightarrow aB$

Uma transição de um estado A para um estado B consumindo o símbolo a

- $A \rightarrow a$

Uma transição de um estado A para um **estado final** Z sem saídas, consumindo o símbolo a

- $A \rightarrow \lambda$

O **estado** A é **final**

- A **variável de partida** é o **estado inicial**

Exemplo

19

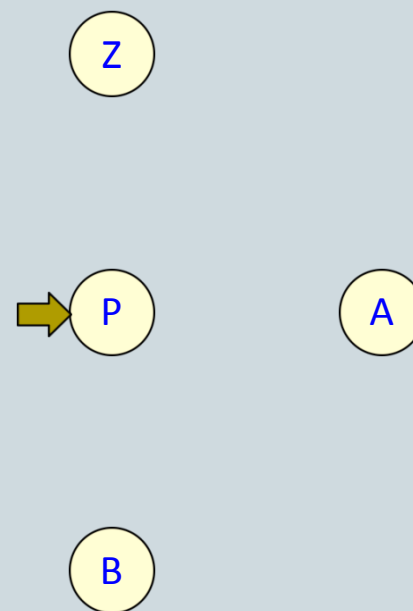
- Dada a seguinte GR, gerar um AFN correspondente

$P \rightarrow 0A \mid 0B \mid 1$

$A \rightarrow 0B \mid \lambda$

$B \rightarrow 0B \mid 1P$

- Conjunto de estados
 - Há uma regra que gera apenas um terminal ($P \rightarrow 1$),
 - Então temos que criar um estado final especial Z



Exemplo

20

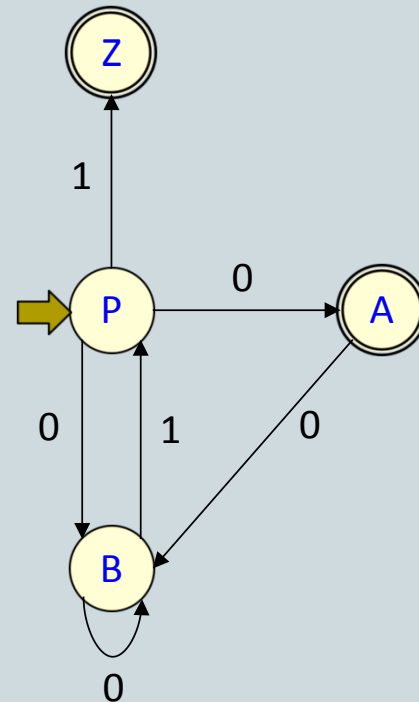
- Dada a seguinte GR, gerar um AFN correspondente

$P \rightarrow 0A \mid 0B \mid 1$

$A \rightarrow 0B \mid \lambda$

$B \rightarrow 0B \mid 1P$

- Conjunto de estados
 - Há uma regra que gera apenas um terminal ($P \rightarrow 1$),
 - Então temos que criar um estado final especial Z



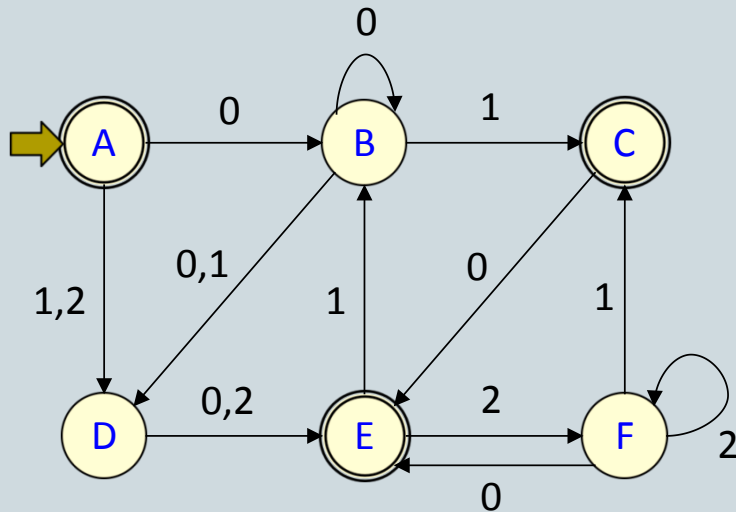
Transformando AFNs em GRs

21

- O processo contrário — transformar AFNs em GRs — é mais simples
- Os passos são
 - Os estados do AFN original representam as variáveis da GR gerada
 - A variável de partida será o estado inicial
 - Cada estado final A gera uma regra do tipo $A \rightarrow \lambda$
 - Cada transição de um estado A para um estado B consumindo um símbolo a gera uma regra do tipo $A \rightarrow aB$

Exemplo

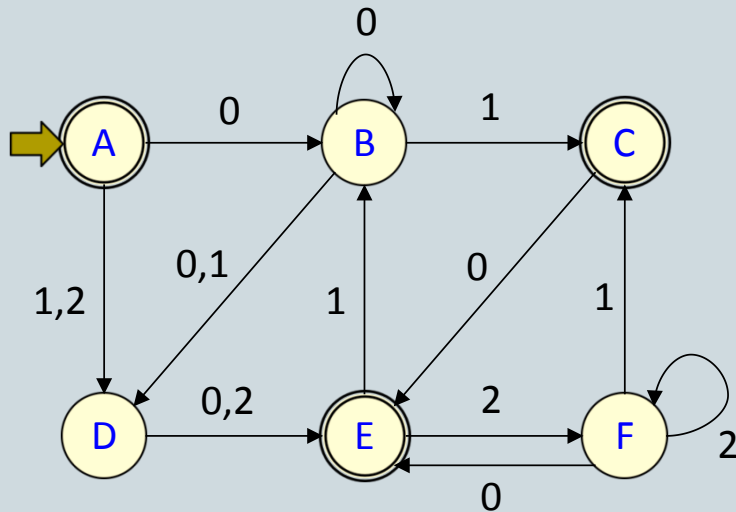
22



- Variável de partida: **A**
- Conjunto de regras:

Exemplo

23



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow$

$B \rightarrow$

$C \rightarrow$

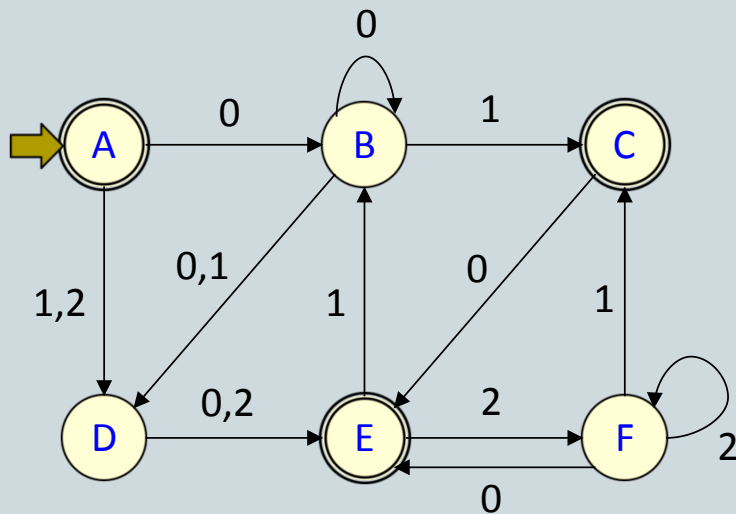
$D \rightarrow$

$E \rightarrow$

$F \rightarrow$

Exemplo

24



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow \lambda \mid 0B \mid 1D \mid 2D$

$B \rightarrow$

$C \rightarrow$

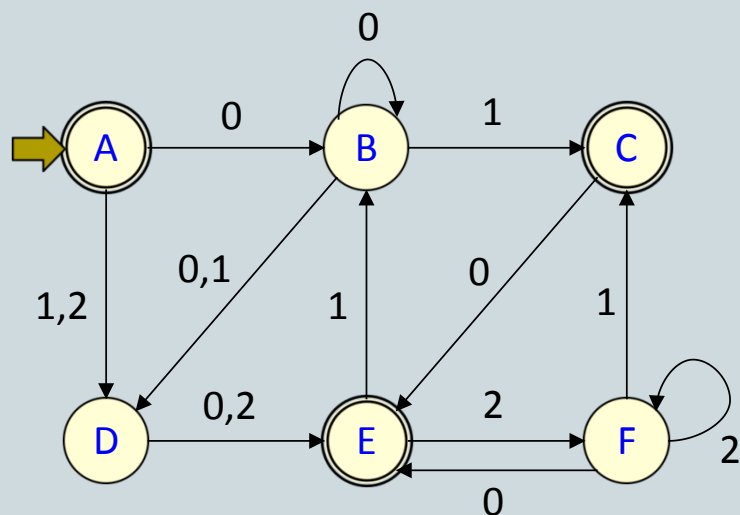
$D \rightarrow$

$E \rightarrow$

$F \rightarrow$

Exemplo

25



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow \lambda \mid 0B \mid 1D \mid 2D$

$B \rightarrow 0B \mid 1C \mid 0D \mid 1D$

$C \rightarrow$

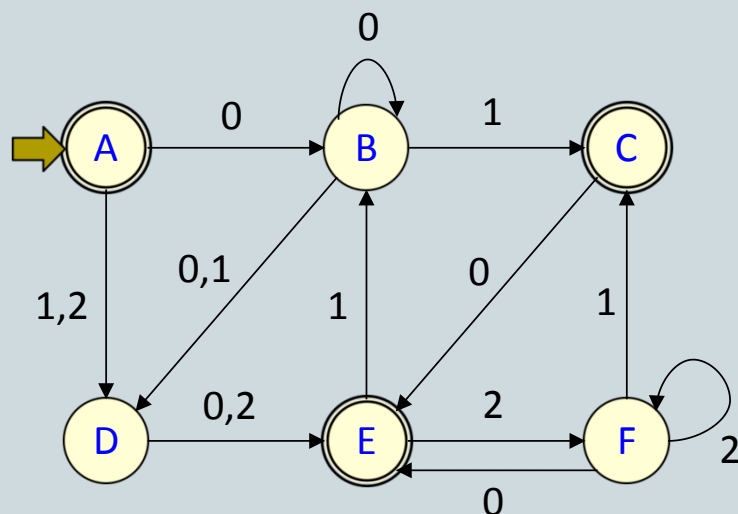
$D \rightarrow$

$E \rightarrow$

$F \rightarrow$

Exemplo

26



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow \lambda \mid 0B \mid 1D \mid 2D$

$B \rightarrow 0B \mid 1C \mid 0D \mid 1D$

$C \rightarrow \lambda \mid 0E$

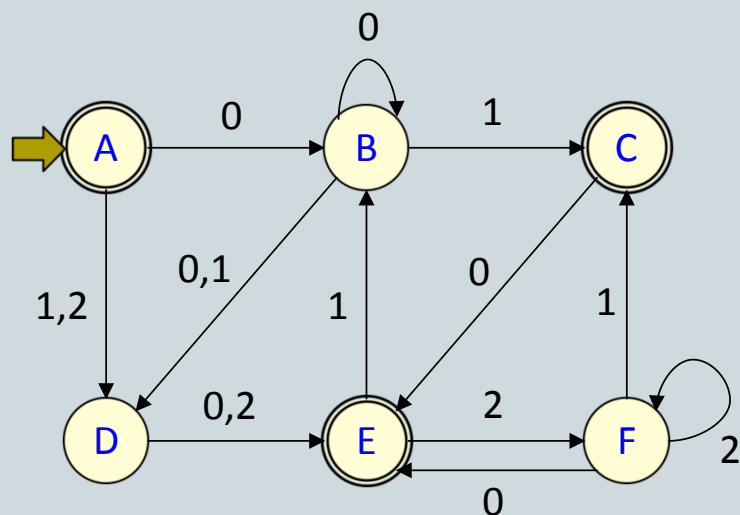
$D \rightarrow$

$E \rightarrow$

$F \rightarrow$

Exemplo

27



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow \lambda \mid 0B \mid 1D \mid 2D$

$B \rightarrow 0B \mid 1C \mid 0D \mid 1D$

$C \rightarrow \lambda \mid 0E$

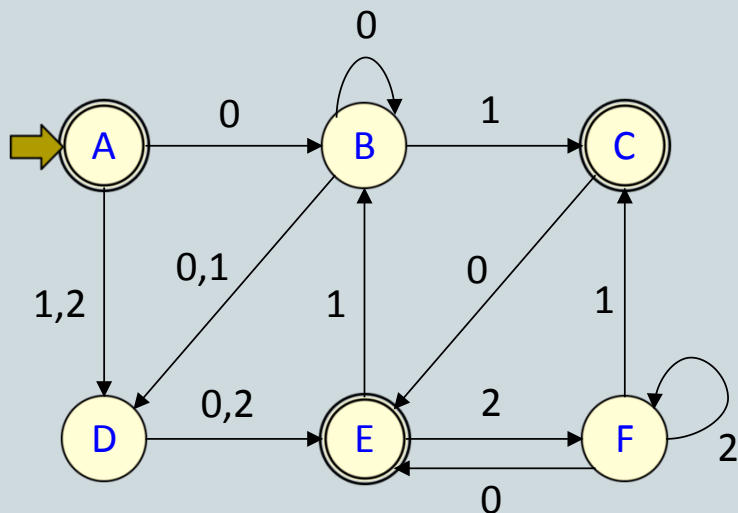
$D \rightarrow 0E \mid 2E$

$E \rightarrow$

$F \rightarrow$

Exemplo

28



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow \lambda \mid 0B \mid 1D \mid 2D$

$B \rightarrow 0B \mid 1C \mid 0D \mid 1D$

$C \rightarrow \lambda \mid 0E$

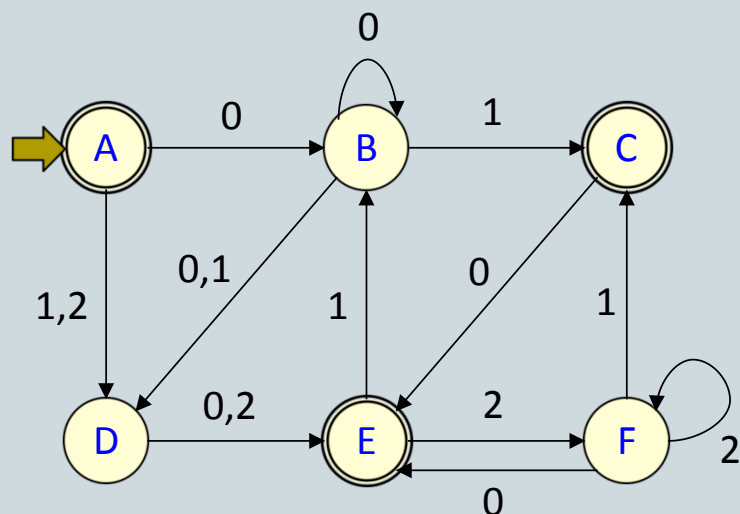
$D \rightarrow 0E \mid 2E$

$E \rightarrow \lambda \mid 1B \mid 2F$

$F \rightarrow$

Exemplo

29



- Variável de partida: A

- Conjunto de regras:

$A \rightarrow \lambda \mid 0B \mid 1D \mid 2D$

$B \rightarrow 0B \mid 1C \mid 0D \mid 1D$

$C \rightarrow \lambda \mid 0E$

$D \rightarrow 0E \mid 2E$

$E \rightarrow \lambda \mid 1B \mid 2F$

$F \rightarrow 0E \mid 1C \mid 2F$

Exercícios

30

- Faça uma GR que gere
 1. Palavras sobre $\{0,1\}$ que terminem em 101
 2. Palavras sobre $\{0,1\}$ que começam com 0 e terminam com 1
 3. Palavras sobre $\{0,1\}$ que terminam em 0 ou 11
 4. Palavras sobre $\{0,1\}$ que tenham um número par de 0s
 5. Palavras sobre $\{0,1\}$ que tenham um número par de 0s e ímpar de 1s
- Converta as GRs acima em AFNs