

Aspectos Teóricos da Computação

Aula 03 - Conceitos iniciais sobre linguagens
formais, máquinas e automação. Gramáticas
Formais

Professor Luís Carlos Pompeu



Conjunto de todas as palavras

- Se Σ é um alfabeto
 - Σ^* conjunto de todas palavras possíveis sobre Σ
 - Ou **fechamento de um alfabeto**
 - Fecho transitivo e reflexivo
$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots \cup \Sigma^n$$
(união de todas as possíveis palavras)
 - Considere que meu alfabeto é $\Sigma = \{a,b\}$ então quantas palavras consigo formar com ele:
 - $\Sigma^0 \rightarrow \epsilon$; $\Sigma^1 \rightarrow a \ b$; $\Sigma^2 \rightarrow aa \ ab \ ba \ bb$ e assim por diante



Conjunto de todas as palavras

- Se Σ é um alfabeto

$$\Sigma^+ = \Sigma^* - \{\epsilon\}$$

- Ou **fechamento positivo** de um alfabeto
- Fecho transitivo
- É exatamente o mesmo conceito, mas não considera a palavra vazia, ou seja, é o conjunto de todas as palavras de um alfabeto exceto a palavra vazia
- Provavelmente vocês perceberam que esse conceito foi visto em matemática discreta, que é uma disciplina pré requisito para entender as linguagens formais



Conjunto de todas as palavras

- Se Σ é um alfabeto (continuando)
 - Σ^* é indutivamente definido
 - Base de Indução
 - $\varepsilon \in \Sigma^*$; a palavra vazia pertence ao alfabeto
 - Para qualquer $x \in \Sigma$, entende-se que $x \in \Sigma^*$.
(sendo x uma palavra, pois x está no fim do nosso alfabeto)



Conjunto de todas as palavras

- Se Σ é um alfabeto (continuando)
 - Passo de Indução
 - Se u e v são palavras de Σ^*
 - Então a concatenação $u.v$ é uma palavra de Σ^*
 - Considere $\Sigma = \{a,b\}$ se $u = aa$ e $v = ba$ e $x = u.v$, então $x = aaba$ e, portanto, $x \in \Sigma^*$
 - Assim como: seja z uma palavra de Σ^* , se $z = a$, e $y = z.v$, então $x = aba$ e, portanto, $y \in \Sigma^*$



Palavra sobre um alfabeto

(mais exemplos)

- Uma palavra sobre o alfabeto Σ é:
 - Qualquer elemento w de Σ^*
 - $w \in \Sigma^*$
- Exemplo: conjunto de todas as palavras:
 - Se $\Sigma = \{a, b\}$, então:
 - $\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, \dots\}$
 - $\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, bbb, \dots\}$

Σ^1

Σ^2

Σ^3

indefinidamente



Comprimento (tamanho) de uma palavra

- O comprimento de uma palavra w :
 - Representado por $|w|$;
 - Número de símbolos que compõe a palavra;
 - Função com domínio em Σ^* e codomínio em \mathbb{N} .
- Exemplo: comprimento de uma palavra:
 - $|abcb| = 4$
 - $|\epsilon| = 0$
 - Até aqui vimos vários conceitos que serão utilizados nos assuntos posteriores

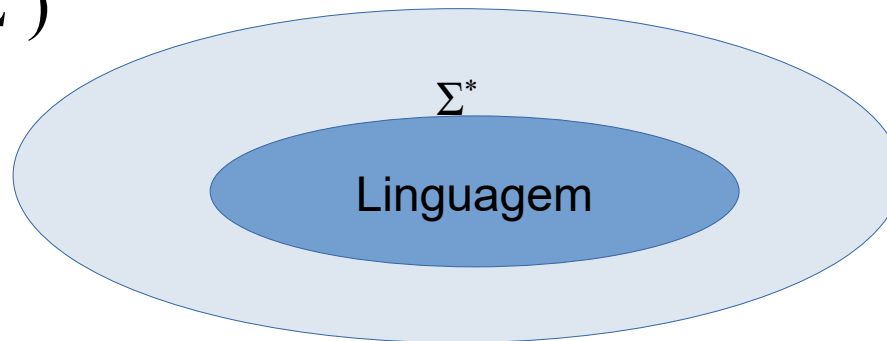


Linguagem Formal



Linguagem Formal

- Dado o alfabeto Σ , o conjunto de palavras L é uma linguagem sobre Σ , se e somente se $L \subseteq \Sigma^*$
- L é um **subconjunto** do conjunto de todas as palavras sobre Σ (Σ^*)



Nem todas as palavras formadas sobre o alfabeto pertencem a uma determinada linguagem, ou seja, gramáticas diferentes geram linguagens (subconjuntos) diferentes.

Portanto posso ter vários subconjuntos dentro do mesmo alfabeto, cada um com suas próprias regras ou gramáticas

Linguagem Formal

Voltando ao exemplo utilizado anteriormente:

Alfabeto da linguagem Pascal

{program, var, integer, real, char, begin, end, if, then, else, for,..., ;, “”, :, :=, ., ...}

O código fonte de um programa corresponde à uma cadeia formada a partir de símbolos do alfabeto.

```
Program Teste;
Var
  i: integer;
Begin
  i:=0;
End.
```

LINGUAGEM

Conjunto de todas as cadeias descritas a partir do alfabeto que **respeitam um conjunto de regras sintáticas.**

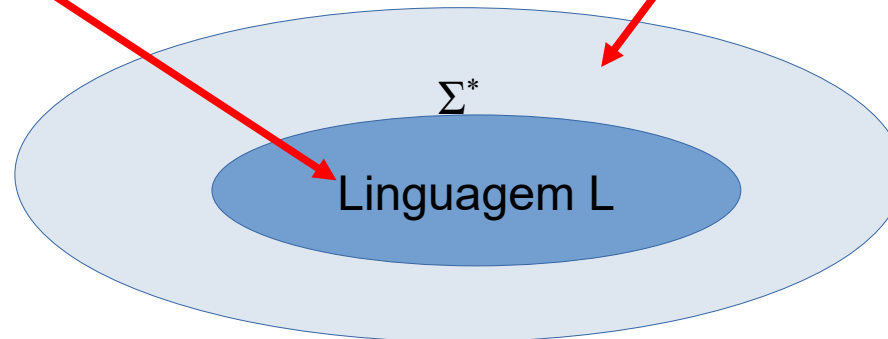
Se eu decidir criar um programa nessa linguagem de programação e sair colocando as palavras dessa linguagem de forma aleatória o programa não irá compilar, preciso seguir as regras ou gramática dessa linguagem.

Cada programa será uma das possíveis palavras que eu consigo formar com o alfabeto

Linguagem Formal

```
program esfera;
  const pi = 3.1415927;
  var raio, area, volume: real;
begin
  read(raio);
  area := 4 * pi * sqr(raio);
  volume := raio * area / 3;
  writeln('Medidas de uma
esfera');
  writeln('Raio = ', raio);
  writeln('Area = ', area);
  writeln('Voluma = ', volume);
end.
```

```
program
var real: o2, m2
end
  o2 = m2
  else for
begin
```



O primeiro código é uma palavra do alfabeto e também é uma palavra da linguagem, porque segue as regras da linguagem.

O segundo também é uma palavra do alfabeto, mas não pertence ao subconjunto da linguagem L.

Exemplo: Linguagem Formal

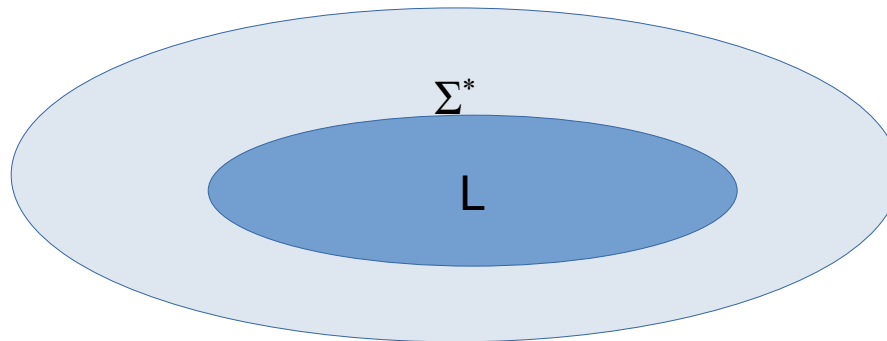
- \emptyset e $\{ \epsilon \}$ são linguagens sobre qualquer alfabeto:
 - $\emptyset \neq \{ \epsilon \}$ Conjunto vazio é um conceito diferente de um conjunto que possui uma palavra vazia
- Σ^* e Σ^+ são linguagens sobre um Σ qualquer:
 - $\Sigma^* \neq \Sigma^+$
- Conjunto de palíndromos sobre $\Sigma = \{a, b\}$
 - $\epsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, \dots$

*Um palíndromo é um número inteiro positivo, sem zeros à esquerda, que é o mesmo se lido da esquerda para a direita ou da direita para a esquerda



Exemplo: Linguagem Formal

- Considerando que L é um subconjunto dos palíndromos de Σ^* , quais são as palavras que pertencem ao subconjunto?
- ϵ , a , b , aa , bb , ab , ba , aaa , bbb , bab , aba , abb , baa ,



$aaaa$, $bbbb$, $aaab$, $baaa$, $baab$, $abba$, $aabb$

Operações sobre linguagens

- Considere $L1$ e $L2$ linguagens definidas sobre Σ :
 - União:
 - $L1 \cup L2 = \{x \mid x \in L1 \text{ ou } x \in L2\}$
 - Intersecção:
 - $L1 \cap L2 = \{x \mid x \in L1 \text{ e } x \in L2\}$
 - Diferença:
 - $L1 - L2 = \{x \mid x \in L1 \text{ e } x \notin L2\}$
 - Concatenação:
 - $L1.L2 = \{x \mid x = yz, y \in L1 \text{ e } z \in L2\}$
 - Complemento:
 - $L1' = \{x \mid x \in \Sigma^* \text{ e } x \notin L1\}$



Exemplo de operações sobre linguagens

- Sejam $L1$ e $L2$ definidas sobre $\{0, 1\}$:
 - $L1 = \{0, 11\}$
 - $L2 = \{0, 1, 00\}$
- $L1 \cup L2 = \{0, 1, 00, 11\}$
- $L1 \cap L2 = \{0\}$
- $L1 - L2 = \{11\}$
- $L2 - L1 = \{1, 00\}$
- $L1.L2 = \{00, 01, 000, 110, 111, 1100\}$ obs. a ordem importa

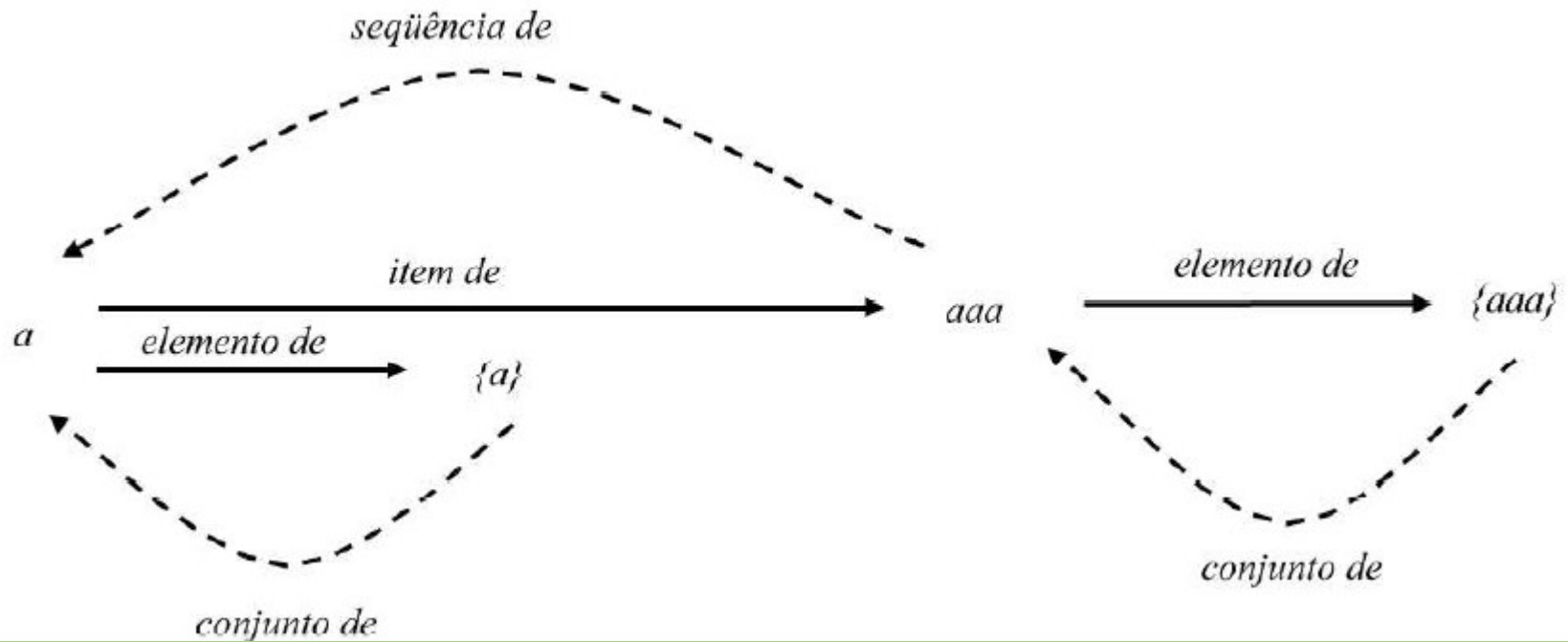


Conjunto de todas as linguagens sobre um alfabeto

- É o **conjunto das partes** de Σ^*
 2^{Σ^*}
- Ou seja, é o conjuntos de todos os subconjuntos de Σ^*

Resumo

- O símbolo a é elemento do alfabeto $\{a\}$...
- E também um item da cadeia (ou palavra) aaa ...
- Que por sua vez é elemento da linguagem $\{aaa\}$...
- Por outro lado, a linguagem $\{aaa\}$ é um conjunto que contém a cadeia aaa ...
- A cadeia aaa é uma sequência de símbolos a
- E o alfabeto $\{a\}$ contém o símbolo a ...



Gramática de Linguagens Formais

Gramática

- Linguagem de programação
 - Definida pelo conjunto de todos os programas (palavras).
- Linguagem de propósitos gerais (Pascal, C, Java, ...):
 - Conjunto de todos os programas é infinito!
 - Uma definição **inadequada** para implementação em computador. Não é possível listar infinitos programas, exatamente por serem infinitos, portanto é necessário ter uma definição finita que possa ser armazenada.

Gramática

- Formalismo Gramática
 - Uma maneira de especificar, de forma finita, linguagens (eventualmente) infinitas.
 - A gramática pode ser vista como um conjunto finito de regras e seguindo esse conjunto de regras eu consigo construir um conjunto infinito de palavras (programas).
 - Dessa forma eu tenho uma definição **adequada** para implementação em computador.

Gramática

- Gramática é, basicamente:
 - Conjunto finito de regras;
 - Quando aplicadas sucessivamente, geram palavras;
 - Conjunto de todas as palavras geradas por uma gramática define a linguagem.
- Gramáticas para linguagens naturais como Português:
 - As mesmas que as usadas para linguagens artificiais como Pascal, C ou Java



Gramática

- Gramática, Gramática Irrestrita ou Gramática de Chomsky

$$\mathbf{G = (V, T, P, S)}$$

- **V**: conjunto finito de símbolos variáveis ou não terminais;
- **T**: conjunto finito de símbolos terminais disjunto de V;
- **P**: Produções. Relação finita: $(V \cup T)^+ \rightarrow (V \cup T)^*$;
 - Regra de produção, composto pela união transitiva e reflexiva de V e T, sendo que a união transitiva faz parte da reflexiva e pode ser substituída por esta.
- **S**: elemento distinguido de V: símbolo inicial ou variável inicial;



Gramática

- Representação de uma regra de produção (α , β):
 - $\alpha \rightarrow \beta$ (Alfa gera beta, significa que alfa pode ser substituída por beta em algum momento da aplicação da regra de produção)
 - Quando utilizamos letras gregas, como no α e β , significa que é uma mistura (união de variáveis e terminais)
- Representação abreviada para $\alpha \rightarrow \beta_1$, $\alpha \rightarrow \beta_2$, ..., $\alpha \rightarrow \beta_n$

Convenções

- A, B, C, ..., S, T para símbolos variáveis
- a, b, c, ..., s, t para símbolos terminais
- u, v, w, x, y, z para palavras de símbolos terminais
- α , β , ... para palavras de símbolos variáveis ou terminais
- Por exemplo: Seja o $\Sigma \{A, a, b, c\}$
- Para **abc = w**, ou seja uma palavra composta apenas por símbolos terminais é representada por um símbolo do final do nosso alfabeto.
- Para **abcAb = β** palavra composta por terminais e variáveis deve ser representada símbolo do alfabeto grego



Derivação

- É a aplicação de uma regra de produção;
- Aplicação sucessiva de regras de produção:
 - Permite derivar palavras da linguagem;
 - Fecho transitivo da relação de derivação.

Derivação e relação de derivação

- Derivação:
 - Substituição de uma subpalavra de acordo com uma regra de produção.
- Sucessivos passos de derivação
- \Rightarrow^* Fecho transitivo e reflexivo da relação
 - Zero ou mais passos de derivações sucessivos.
- \Rightarrow^+ Fecho transitivo da relação
 - Um ou mais passos de derivações sucessivos.
- \Rightarrow^i
 - Exatos i passos de derivações sucessivos (i é natural).



Derivação e relação de derivação

- Considere a gramática G :

$$G = (V, T, P, S)$$

- Linguagem Gerada por G :

$$L(G) \text{ ou } \text{GERA}(G)$$

- Palavras de símbolos terminais deriváveis a partir de S :

- $L(G) = \{ w \in T^* \mid S \Rightarrow^+ w \}$ (| desde que ou tal que)

- A linguagem gerada pela gramática G é formada por toda palavra (w) pertencente ao conjunto de todas as palavras possíveis sobre o alfabeto T (ou fechamento do alfabeto) tal que w tenha sido gerada por pelo menos 1 passo de sucessivas derivações da variável inicial S .



Exemplo: Números naturais

- Gramática $G = (V, T, P, N)$, nesse caso o símbolo inicial é N e não S
 - $V = \{ N, D \}$
 - $T = \{0, 1, 2, \dots, 9\}$
 - $P = \{N \rightarrow D, N \rightarrow DN, D \rightarrow 0 \mid 1 \mid \dots \mid 9\}$
 - Nossas regras de produção, $N \rightarrow D$, ou N gera D, significa que N pode substituir D, DN pode substituir N e D pode substituir qualquer dos símbolos de 0 até 9
- Gera, sintaticamente, o conjunto dos números naturais:
 - Se distinguem os zeros à esquerda.
 - Exemplo: 123 diferente de 0123.



Exemplo: Números naturais

- $G = (V, T, P, N)$ **Observe que toda derivação deve iniciar com N**
 - $V = \{ N, D \}$
 - $T = \{0, 1, 2, \dots, 9\}$
 - $P = \{N \rightarrow D, N \rightarrow DN, D \rightarrow 0 \mid 1 \mid \dots \mid 9\}$
- Uma derivação do número 243
- $N \Rightarrow N \rightarrow DN$
- $DN \Rightarrow D \rightarrow 2$
- $2N \Rightarrow N \rightarrow DN$
- $2DN \Rightarrow D \rightarrow 4$
- $24N \Rightarrow N \rightarrow D$
- $24D \Rightarrow D \rightarrow 3$
- 243

Exemplo: Números naturais

- **Vamos entender**
 - $P = \{N \rightarrow D, N \rightarrow DN, D \rightarrow 0 \mid 1 \mid \dots \mid 9\}$
- Uma derivação do número 243
 - Se eu derivasse $N \Rightarrow N \rightarrow D$
 - $D \Rightarrow D \rightarrow 2$
 - 2 Acabaria aqui e não conseguiria gerar 243 que é o valor desejado
 - Então, usamos a regra $N \Rightarrow N \rightarrow DN$
 - $DN \Rightarrow D \rightarrow 2$ dessa forma o D derivou (gerou) o 2 e ainda sobrou 2N
 - $2N \Rightarrow N \rightarrow DN$ novamente o N derivou DN
 - $2DN \Rightarrow D \rightarrow 4$ e o D derivou o 4 e novamente sobrou o N (24N)
 - $24N \Rightarrow N \rightarrow D$ agora só preciso de mais um símbolo, portanto, aplicamos a regra N deriva D
 - $24D \Rightarrow D \rightarrow 3$
 - 243
 - seguindo esse conjunto de regras finitas de produção, consigo gerar qualquer número natural



Exemplo: Números naturais

- Portanto:
 - $S \Rightarrow^* 243$
 - $S \Rightarrow^+ 243$
 - $S \Rightarrow^6 243$
- Interpretação indutiva da gramática
 - Base de indução
 - Todo dígito é natural
 - Passo de Indução
 - Se n é natural, então a concatenação com qualquer dígito também é natural



Exemplo: Palavra duplicada

- $G = (\{ S, X, Y, A, B, F \}, \{ a, b \}, P, S)$
- Na qual:
- $P = \{ S \rightarrow XY,$
 $X \rightarrow XaA \mid XbB \mid F$
 $Aa \rightarrow aA, Ab \rightarrow bA, AY \rightarrow Ya,$
 $Ba \rightarrow aB, Bb \rightarrow bB, BY \rightarrow Yb,$
 $Fa \rightarrow aF, Fb \rightarrow bF, FY \rightarrow \epsilon \}$
- Faça a derivação da palavra: **baba**



Exemplo: Palavra duplicada

- Derivação de baba:

- $S \Rightarrow S \rightarrow XY$
- $XY \Rightarrow X \rightarrow XaA$
- $XaAY \Rightarrow AY \rightarrow Ya$
- $XaYa \Rightarrow X \rightarrow XbB$
- $XbBaYa \Rightarrow Ba \rightarrow aB$
- $XbaBYa \Rightarrow BY \rightarrow Yb$
- $XbaYba \Rightarrow X \rightarrow F$
- $FbaYba \Rightarrow Fb \rightarrow bF$
- $bFaYba \Rightarrow Fa \rightarrow aF$
- $baFYba \Rightarrow FY \rightarrow \epsilon$
- $baba$

$G = (\{S, X, Y, A, B, F\}, \{a, b\}, P, S)$

$P = \{$

- $S \rightarrow XY,$
- $X \rightarrow XaA \mid XbB \mid F,$
- $Aa \rightarrow aA,$
- $Ab \rightarrow bA,$
- $AY \rightarrow Ya,$
- $Ba \rightarrow aB,$
- $Bb \rightarrow bB,$
- $BY \rightarrow Yb,$
- $Fa \rightarrow aF,$
- $Fb \rightarrow bF,$
- $FY \rightarrow \epsilon$

$\}$

Exemplo: Palavra duplicada

• Derivação de baba:

- S \Rightarrow S \rightarrow XY % Única transição possível.
- XY \Rightarrow X \rightarrow XaA % Gera um 'a' da primeira parte de b(a)ba.
- XaAY \Rightarrow AY \rightarrow Ya % Gera o outro 'a' bab(a).
- XaYa \Rightarrow X \rightarrow XbB % Gera um 'b' da primeira parte de (b)aba.
- XbBaYa \Rightarrow Ba \rightarrow aB % Caminha com 'B' para o inicio da outra metade.
- XbaBYa \Rightarrow BY \rightarrow Yb % Gera o outro 'b' de ba(b)a.
- XbaYba \Rightarrow X \rightarrow F % Transforma X em F para poder ser removido.
- FbaYba \Rightarrow Fb \rightarrow bF % Caminha com F até o inicio da segunda metade.
- bFaYba \Rightarrow Fa \rightarrow aF % Caminha com F até o inicio da segunda metade.
- baFYba \Rightarrow FY \rightarrow ϵ % Remove FY.
- baba

$G = (\{ S, X, Y, A, B, F \}, \{ a, b \}, P, S)$

$P = \{$

S	\rightarrow XY,
X	\rightarrow XaA XbB F,
Aa	\rightarrow aA,
Ab	\rightarrow bA,
AY	\rightarrow Ya,
Ba	\rightarrow aB,
Bb	\rightarrow bB,
BY	\rightarrow Yb,
Fa	\rightarrow aF,
Fb	\rightarrow bF,
FY	$\rightarrow \epsilon$ }

