Mais Problemas Não Decidíveis

Redução

Na aula anterior mostramos que $\text{Halt}_{\text{Java}}$, E_{TM} , EQ_{TM} e ALL_{TM} são todos não decidíveis, a partir da não decibilidade de A_{TM}.

De fato, pudemos extrair mais informação: em cada caso, pudemos dizer se instâncias positivas/negativas são reconhecíveis.

EX: Para A_{TM}, as instâncias negativas são mais difíceis porque pode ocorrer loop infinito nesse caso: não se pode eliminar sistematicamente todos os possíveis loops!

Podemos formalizar essas idéias por meio das noções de *Turing Redução* e *Redução por Mapeamento* :

Redução

DEF: Dadas linguagens[1] A, B.

Se pode ser construído um decisor para *A*, supondo que existe um decisor para B (chamando B como sub-rotina[2]), então A é ***Turing redutível*** (ou ***redutível***) a B.

Além disso, se existe uma função computável[3] f tal que $f(A) \subseteq B$ *e* $f(\overline{A}) \subseteq \overline{B}$, então A é ***redutível por mapeamento*** [4] a B .

Por outro lado, se existe uma função computável f tal que $f(A) \subseteq \overline{B}$ e $f(\overline{A}) \subseteq B$, então A é ***redutível por co-mapeamento***[5] a B .

Redução
Notação

- *A* ***Turing redutível*** a *B* :

$$A \leq_T B$$

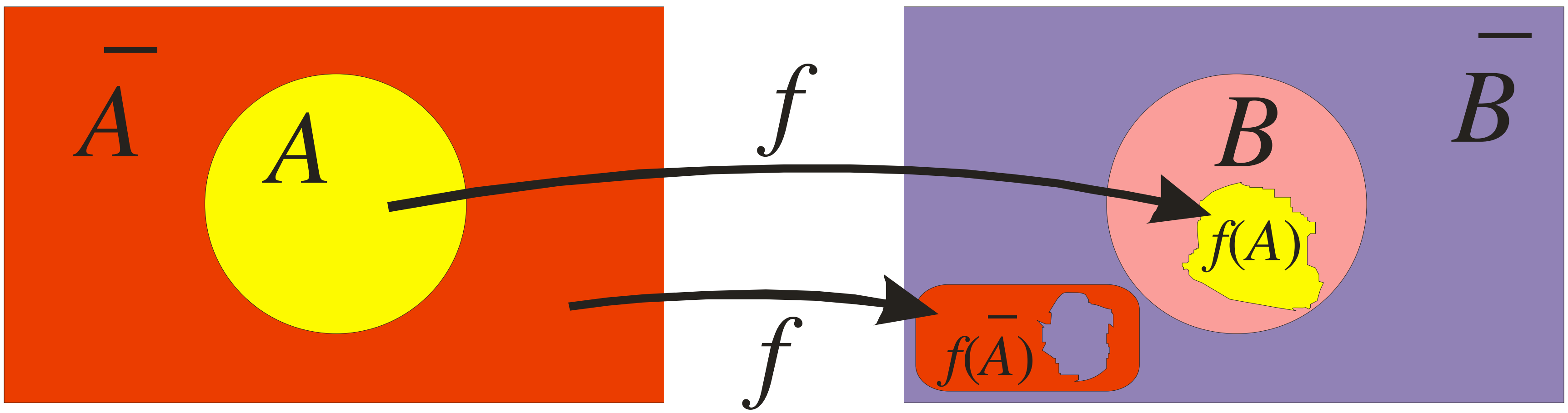
- *A* ***redutível por mapeamento*** a *B*:[2]

$$A \leq_m B \qquad \text{or} \qquad A \rightarrow_+ B$$

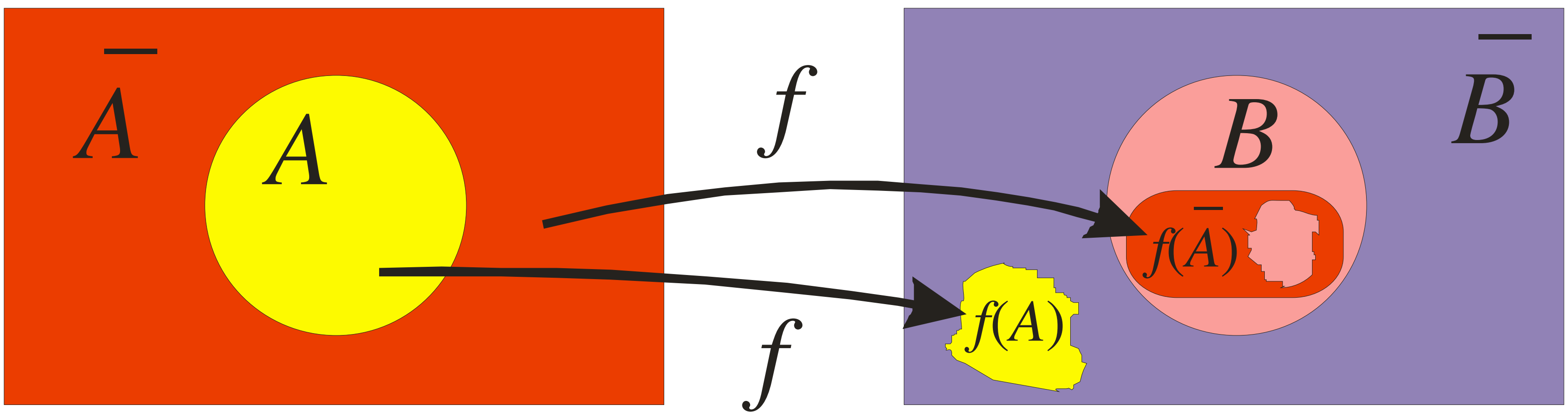
- *A* ***redutível por co-mapeamento*** a *B:*

$$A \leq_m \overline{B} \qquad \text{or} \qquad A \rightarrow_- B$$

Visualizando o Mapeamento de Redução $A \rightarrow_{+} B$



Visualizando Co-Mapeamento de Redução $A \rightarrow_{-} B$



Redução
Lemas Simples

LEMA: Sejam A e B linguagens:

- Se $A \rightarrow_{+} B$ então $A \leq_T B$
- Se $A \rightarrow_{-} B$ então $A \leq_T B$
- Se $A \rightarrow_{+} B$ e B é reconhecível, então A é reconhecível
- Se $A \rightarrow_{-} B$ e B is reconhecível, então A é co-reconhecível
- Se $A \leq_T B$ e A não é decidível, então B não é decidível

Redução
Lemas Simples

Lema de Transitividade. Sejam *A, B* e *C* linguagens:

- Se $A \rightarrow_{+} B$ e $B \rightarrow_{+} C$ então $A \rightarrow_{+} C$
- Se $A \rightarrow_{-} B$ e $B \rightarrow_{-} C$ então $A \rightarrow_{+} C$
- Se $A \leq_{T} B$ e $B \leq_{T} C$ então $A \leq_{T} C$

Resultados anteriores em termos de redução:$\text{Halt}_{\text{Java}}$

Vamos rever os resultados enteriores em termos da nova terminologia.

SUPONHA: A_{TM} não co-reconhecível.

$\text{Halt}_{\text{Java}}$: Para todo par TM-string (*M,w*), criamos um programa Java **`D.java`** que pára para a entrada *M,w* exatamente quando *M* aceita *w*. Portanto:

$$A_{\text{TM}} \rightarrow_{+} \text{Halt}_{\text{Java}}$$

Consequência: $\text{Halt}_{\text{Java}}$ não é co-reconhecível.

Pode-se mostrar que instâncias positivas são aceitas e, portanto, $\text{Halt}_{\text{Java}}$ é reconhecível, mas não co-reconhecível.

Resultados anteriores em termos de redução: E_{TM}

E_{TM} : Para cada par TM-string (*M,w*), criamos uma TM $K_{M,w}$ que é não vazia exatamente quando *M aceita* w:

$$A_{\text{TM}} \rightarrow_{\text{-}} E_{\text{TM}}$$

Consequência: E_{TM} não é reconhecível.

Podemos também mostrar que instâncias negativas são aceitas e, portanto, E_{TM} é co-reconhecível mas não reconhecível.

Resultados anteriores em termos de redução: ALL_{TM}

ALL_{TM} : Para todo par TM-string (M,w), criamos uma TM $K_{M,w}$ que aceita Σ^* exatamente quando M aceita w :

$$A_{\mathrm{TM}} \rightarrow_{+} ALL_{\mathrm{TM}}$$

Por outro lado: Todo par TM-string (M,w) dá origem a uma TM enumeradora $E_{M,w}$ que gera Σ^* exatamente quando M não aceita w:

$$A_{\mathrm{TM}} \rightarrow_{-} ALL_{\mathrm{TM}}$$

Consequência: ALL_{TM} não é reconhecível nem co-reconhecível.

Resultados anteriores em termos de redução: EQ_{TM}

EQ_{TM} : Cada instância positiva de E_{TM} é convertida em uma instância de EQ_{TM} , tomando a segunda linguagem vazia:

$$E_{\mathrm{TM}} \rightarrow_{+} EQ_{\mathrm{TM}}$$

De modo similar, ALL_{TM} reduz para EQ_{TM}:

$$ALL_{\mathrm{TM}} \rightarrow_{+} EQ_{\mathrm{TM}}$$

Consequência: EQ_{TM} não é reconhecível, nem co-reconhecível.

Redução: Problemas Não Decidíveis Equivalentes

Podemos usar a noção de redução para estudar a dificuldade relativa de problemas. Talvez alguns sejam "*menos decidíveis*" que outros!

EX: Parece a princípio que E_{TM} é mais difícil que A_{TM}, já que queremos advinhar se *algum* é aceito, vs. advinhar se um string particular é aceito. Entretanto:

E_{TM} tem a mesma dificuldade que A_{TM}.

Redução: Problemas Não Decidíveis Equivalentes

E_{TM} tem a mesma dificuldade que A_{TM}.

Prova. Já mostramos que $A_{TM} \leq_T E_{TM}$. Vamos agora mostrar o oposto, isto é, que $E_{TM} \leq_T A_{TM}$. Dada uma TM M construa a TM M' tal que:

Sobre a entrada w :

1. Simula M para todas as possíveis entradas, usando *multi-threading*
2. Se alguma *thread* aceita, aceita.

Q: Quando M' não aceita w ?

Redução: Problemas Não Decidíveis Equivalentes

R: M' não aceita w somente quando entra em loop infinito, o que acontece apenas quando M não aceita nada. I.e., M' aceita w sse $L(M)$ é não vazia.

Consequentemente, de pudermos decidir se M' aceita w podemos decidir se L(M) é vazia! Isso mostra que um algoritmo A_{TM} dá origem a um algoritmo para E_{TM} e, portanto, $E_{TM} \leq_T A_{TM}$.

Notação para ***equivalêcia algoritmica*** :

$$A \approx_T B \text{ (i.e., } A \leq_T B \text{ and } A \geq_T B \text{)}$$

Além da Não Decidibilidade

Existem problemas que são ainda mais difíceis que os problemas anteriores, i.e. não podem ser reduzidos a A_{TM}. Pelo argumento de contagem, eles devem existir. O exercício 6.21 em Sipser mostra que o seguinte problema é estritamente mais difícil que A_{TM}:

Dada uma "Máquina de Turing" M que usa uma suposta solução de A_{TM} como subrotina, decidir se M aceita w.

A prova é quase idêntica à prova de que A_{TM} não é decidível.

Hierarquia de Chomsky

Até agora vimos três tipos de gramáticas:

Tipo 1: Gramáticas Sensíveis ao Contexto

- Produções são da forma $u \rightarrow v$ com u contendo alguma variável e $|u| \leq |v|$

Tipo 2: Gramáticas Livres de Contexto

Tipo 3: Gramáticas Lineares à Direita

- Produções são da forma $A \rightarrow uB$ com u sendo um string terminal, e B uma variável ou vazio
- Gera as linguagens regulares

Q: Adivinhe qual é o último tipo.

Hierarquia de Chomsky

R: Deve ser tipo 4 ou tipo 0. Quanto maior o número maiores as restrições sobre a gramática, resultando em uma classe de linguagens menor.

TIPO 4 teria que ser mais restrito que linguagens regulares, podendo apenas consistir na classe de linguagens finitas, mas isso não tem utilidade.

Então o último tipo deve ser TIPO 0: gramáticas completamente irrestritas.

TIPO 0
Gramáticas Irrestritas

DEF: Uma ***gramática irrestra*** consiste de (V, Σ, R, S) onde:

- V –um conjunto finito de***variáveis***
- Σ – um conjunto finito de ***terminais***
- R – um conjunto finito de ***regras*** da forma $u \rightarrow v$ com $u \in (\Sigma \cup V)^+$ e $v \in (\Sigma \cup V)^*$
- $S \in V$ – o ***símbolo inicial***

A computação é como em qualquer gramática. Um string em Σ^* é gerado pela gramática se é derivável a partir de S usando as regras R.

Gramáticas Irrestritas
Exemplo

Note que o string derivado por uma gramática irrestrita pode tanto crescer como *encolher.*

EX: Considere a seguinte gramática irrestrita que gera números unários positivos que são potência de 2:

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad$ $A \rightarrow 1A1 \mid \$B$ $\quad$ $B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad$ $1C\$ \rightarrow D1\$$ $\quad$ $1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad$ $\$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

S

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad$ $A \rightarrow 1A1 \mid \$B$ $\quad$ $B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad$ $1C\$ \rightarrow D1\$$ $\quad$ $1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad$ $\$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $A \rightarrow 1A1 \mid \$B$ $B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $1C\$ \rightarrow D1\$$ $1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow 1^4A1111\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$
$1^4A1111\$ \Rightarrow 1^4\$B1111\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow 1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad$ $A \rightarrow 1A1 \mid \$B$ $\quad$ $B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad$ $1C\$ \rightarrow D1\$$ $\quad$ $1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad$ $\$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$
$1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$ \Rightarrow$
$1^4\$11C\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad$ $A \rightarrow 1A1 \mid \$B$ $\quad$ $B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad$ $1C\$ \rightarrow D1\$$ $\quad$ $1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad$ $\$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$
$1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$ \Rightarrow$
$1^4\$11C\$ \Rightarrow 1^4\$1D1\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $A \rightarrow 1A1 \mid \$B$ $B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $1C\$ \rightarrow D1\$$ $1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$
$1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$ \Rightarrow$
$1^4\$11C\$ \Rightarrow 1^4\$1D1\$ \Rightarrow 1^4\$D11\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$$
$$1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$ \Rightarrow$$
$$1^4\$11C\$ \Rightarrow 1^4\$1D1\$ \Rightarrow 1^4\$D11\$ \Rightarrow 1^4\$1C\$$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$
$1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$ \Rightarrow$
$1^4\$11C\$ \Rightarrow 1^4\$1D1\$ \Rightarrow 1^4\$D11\$ \Rightarrow 1^4\$1C\$ \Rightarrow$
$1^4\$D1\$$

Gramáticas Irrestritas
Exemplo

$S \rightarrow A\$$ $\quad A \rightarrow 1A1 \mid \$B$ $\quad B11 \rightarrow 1C$

$C11 \rightarrow 1C$ $\quad 1C\$ \rightarrow D1\$$ $\quad 1D \rightarrow D1$

$\$D11 \rightarrow \$1C$ $\quad \$D1\$ \rightarrow \varepsilon$

Por exemplo, 1^4 é gerado da seguinte forma:

$$S \Rightarrow A\$ \Rightarrow 1A1\$ \Rightarrow 1^2A11\$ \Rightarrow 1^3A111\$ \Rightarrow$$
$$1^4A1111\$ \Rightarrow 1^4\$B1111\$ \Rightarrow 1^4\$1C11\$ \Rightarrow$$
$$1^4\$11C\$ \Rightarrow 1^4\$1D1\$ \Rightarrow 1^4\$D11\$ \Rightarrow 1^4\$1C\$ \Rightarrow$$
$$1^4\$D1\$ \Rightarrow 1^4 \quad \text{ACEITO!}$$

Gramáticas Irrestritas
Exemplo

De modo geral:

$$S \Rightarrow A\$ \Rightarrow^* 1^{2^n} A1^{2^n}\$ \Rightarrow 1^{2^n}\$B1^{2^n}\$$$

$$\Rightarrow^* 1^{2^n}\$1^{2^{n-1}} C\$ \Rightarrow^* 1^{2^n}\$D1^{2^{n-1}}\$$$

$$\Rightarrow^* 1^{2^n}\$1^{2^{n-2}} C\$ \Rightarrow^* 1^{2^n}\$D1^{2^{n-2}}\$ \Rightarrow^* \ldots$$

$$\ldots \Rightarrow^* 1^{2^n}\$1C\$ \Rightarrow 1^{2^n}\$D1\$ \Rightarrow 1^{2^n}$$

Quando o segund passo deriva um número que não é potência de 2, obtemos o substring “*C*1$” em algum ponto adiante da derivação, chegando a um ponto *morto*. Portanto, apenas potências de 2 são geradas.

Gramáticas Irrestritas Exemplo

A sequência de derivação anterior é bastante semelhante a um algoritmo, no sentido de que existem diferentes estágios na derivação:

1) Gera $1^k\$B1^k\$$
2) Divide a sequencia de 1's por by 2. Se a sequência tem tamanho 1, vai para o passo #3

   Se não é divisível por 2, a derivação pára sem saída

   Senão, repete #2
3) Vai para $1^k\$D1\$$ e então para 1^k sendo k uma potência de 2.

Gramáticas Irrestritas

A_{UG}

O problema de aceitação para gramáticas irrestritas é denotado por A_{UG}. A semelhança entre derivações nessa gramática e algoritmos nos indica que é verdade o seguinte:

FATO: A_{UG} não é decidível.

Sistemas Semi-Thue

Um sistema Semi-Thue é uma gramática em sua forma mais essencial:

DEF: Um ***sistema semi-Thue*** consiste em (Σ, R) onde Σ é uma alfabeto e R um conjunto de regras da forma $u \rightarrow v$ onde $u \in \Sigma^+$ e $v \in \Sigma^*$.

Computações são agora relativas: representam que strings podem ser derivados a partir de que outros.

O Problema Semi-Thue

O problema de decisão que nos concerne é:

STP = Problema Semi-Thue =

“Dados $u,v \in \Sigma^*$.

É possível a derivação $u \Rightarrow^* v$?”

I.e., dados dois strings arbitrários sobre o mesmo alfabeto, pode um derivar o outro?

De Gramáticas Irrestritas para Sistemas Semi-Thue

Sistemas Semi-Thue e gramáticas irrestritas são essencialmente a mesma coisa, exceto por diferenças artificiais. Para o nosso propósito, precisaremos do seguinte lema, embora se possa provar propriedade mais forte:

LEMA: STP $\rightarrow_{+} A_{\mathrm{UG}}$.

Prova. Vamos mostrar que sistemas semi-Thue podem ser naturalmente convertidos para gramáticas irrestritas.

De Gramáticas Irrestritas para Sistemas Semi-Thue

Dada uma instância de STP com alfabeto Σ, regras R, e strings u,v, construímos a gramática G de mesmo alfabeto, com conjunto de variáveis $V = \{S\}$ tal que $S \Rightarrow^* v$ sse $u \Rightarrow^* v$ no STP original. Se isso pode ser feito, então a instância do STP é positiva sse a instância $\langle G,v \rangle$ de A_{UG} é positiva ($v \in L(G)$).

Q: Que produções precisam ser adicionadas a G para que isso ocorra?

De Gramáticas Irrestritas para Sistemas Semi-Thue

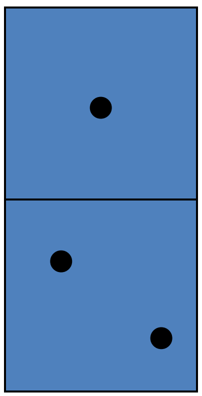
A: Simples, adicione $S \rightarrow u$!

Problema de Correspondência de Post (Dominó)

Usualmente dominós são jogados assim:

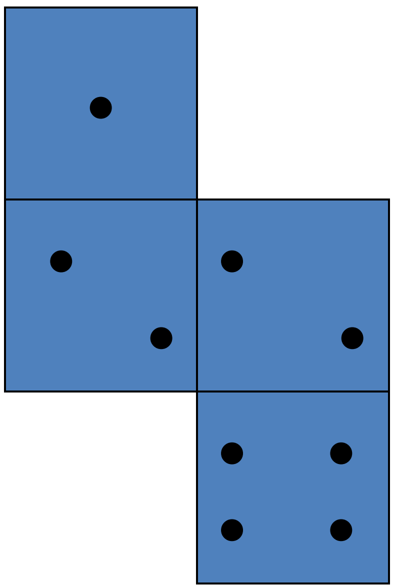
Problema de Correspondência de Post (Dominó)

Usualmente dominós são jogados assim:



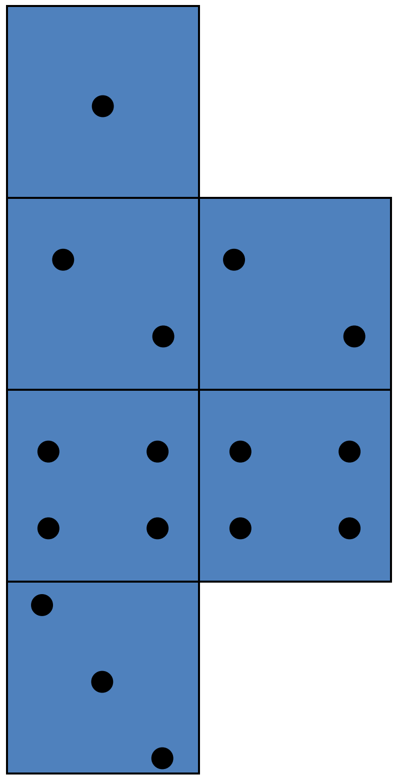
Problema de Correspondência de Post (Dominó)

Usualmente dominós são jogados assim:



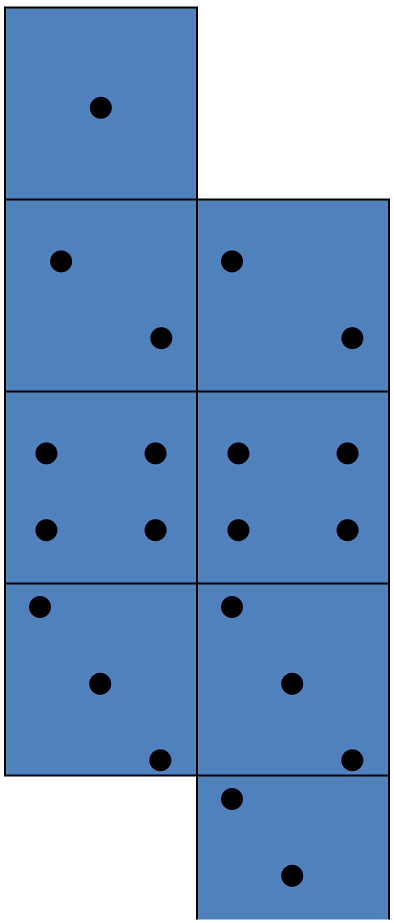
Problema de Correspondência de Post (Dominó)

Usualmente dominós são jogados assim:



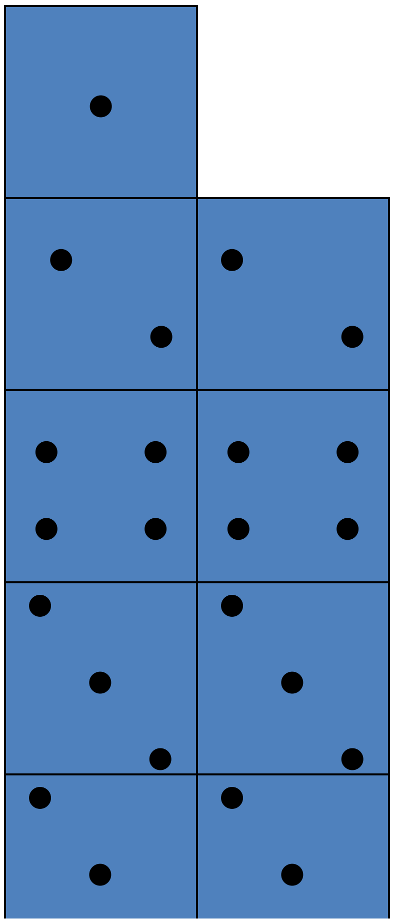
Problema de Correspondência de Post (Dominó)

Usualmente dominós são jogados assim:



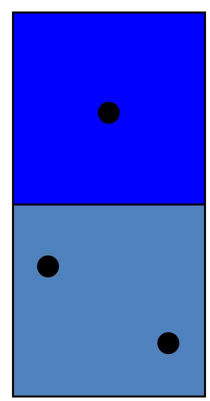
Problema de Correspondência de Post (Dominó)

Usualmente dominós são jogados assim:



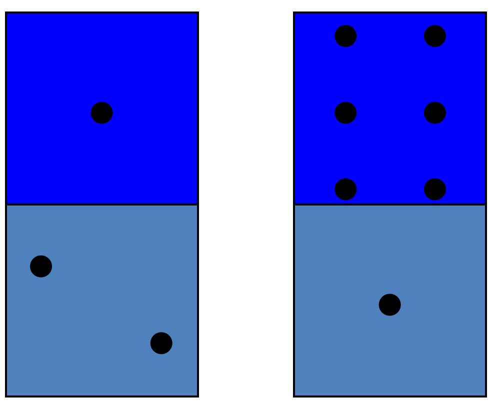
Problema de Correspondência de Post (Dominó)

Aqui vamos jogar horizontalmente. Além disso, os dominós não podem ser virados, e por isso as metades têm cores distintas:



Problema de Correspondência de Post (Dominó)

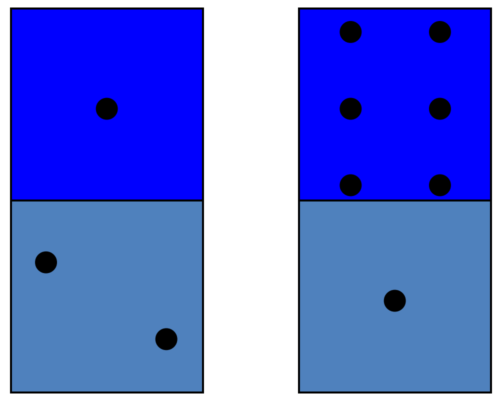
O objetivo do jogo será ter o mesmo número de pontos na parte de cima e na de baixo. O jogador recebe um conjunto de *protótipos* de dominós, podendo escolher quantos precisar de cada um dos protótipos

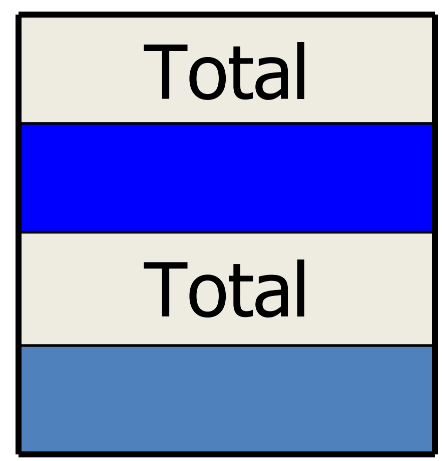


Vamos jogar com os 2 protótipos acima.

Problema de Correspondência de Post (Dominó)

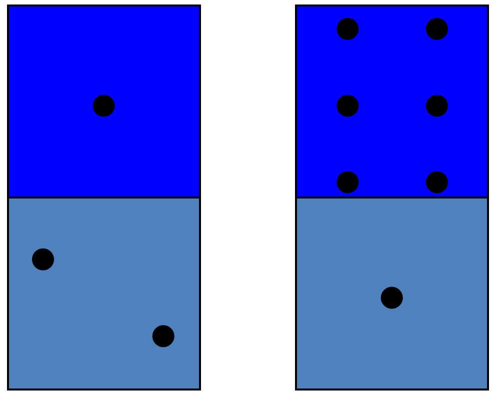
Vamos jogar com os 2 protótipos:

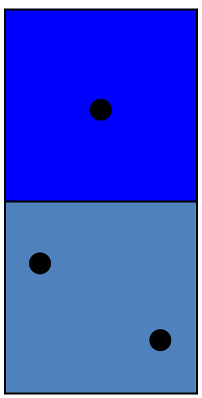




Problema de Correspondência de Post (Dominó)

Vamos jogar com os 2 protótipos:

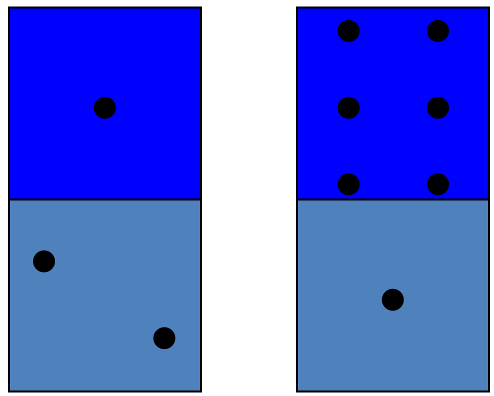


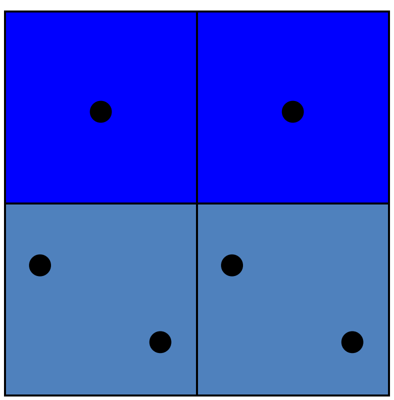


Total
1
Total
2

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 2 protótipos:

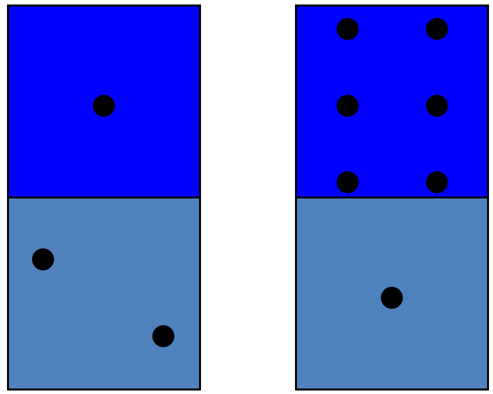


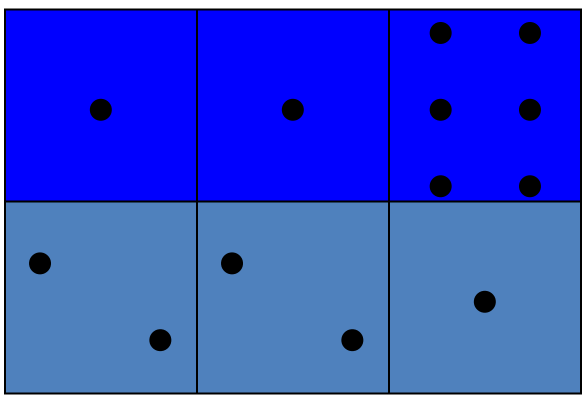


Total
2
Total
4

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 2 protótipos:

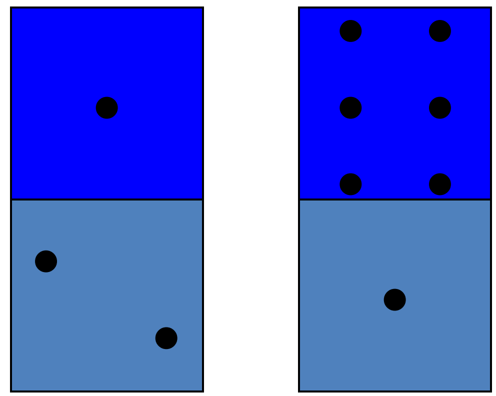


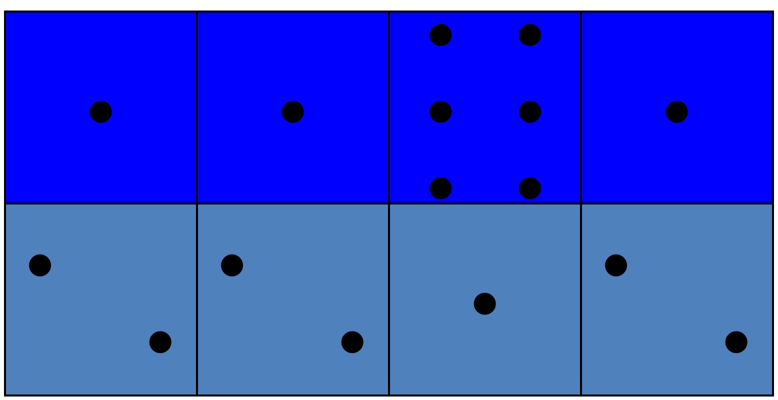


Total
8
Total
5

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 2 protótipos:

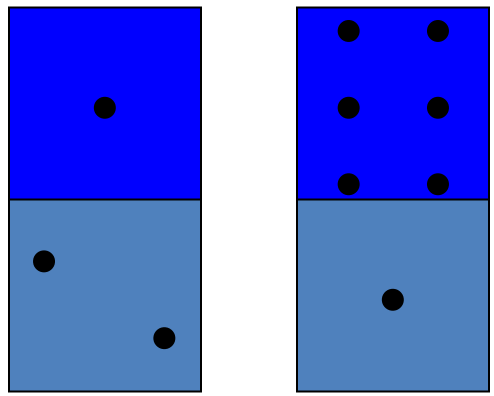


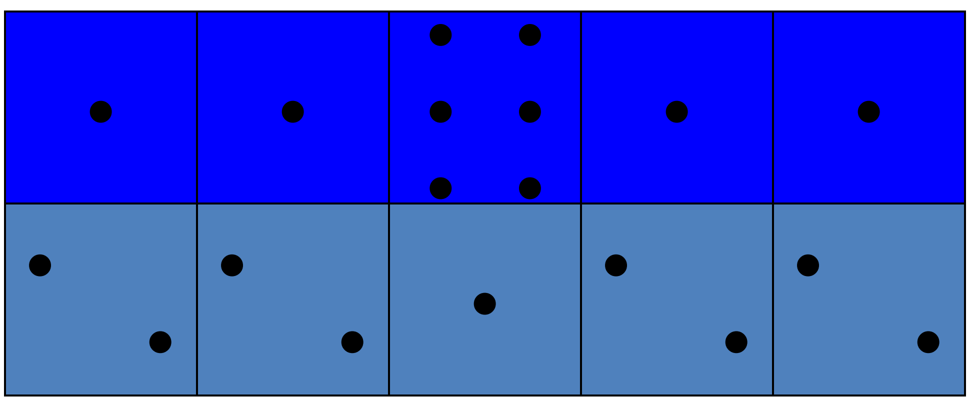


Total
9
Total
7

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 2 protótipos:

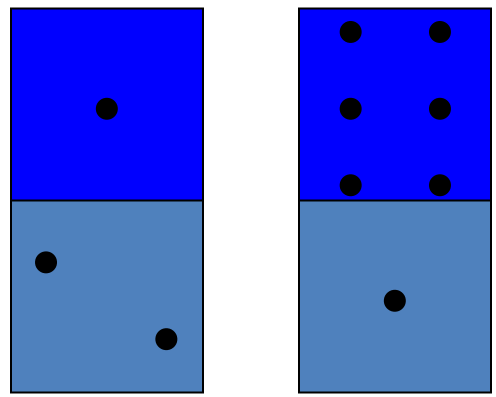




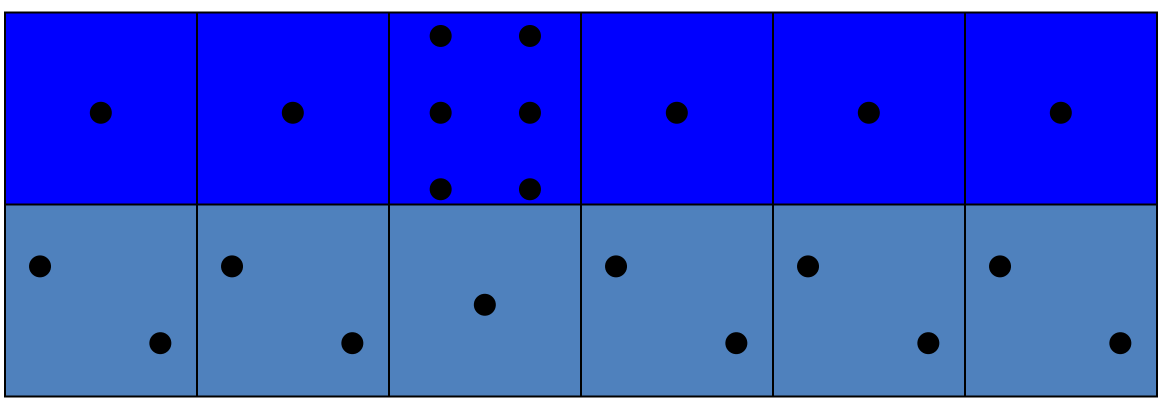
Total
10
Total
9

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 2 protótipos:



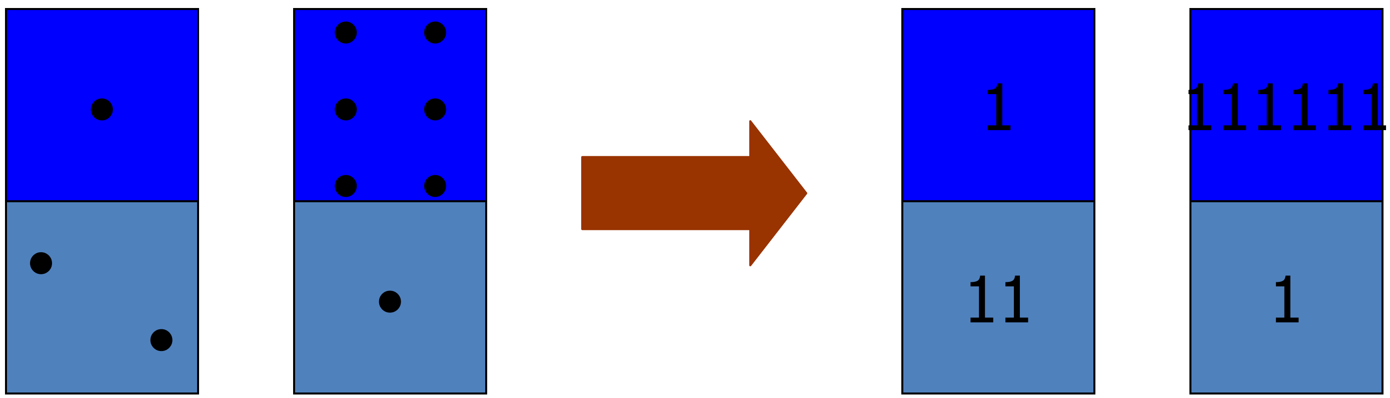
GANHOU!



Total
11
Total
11

Problema de Correspondência de Post (Dominó)

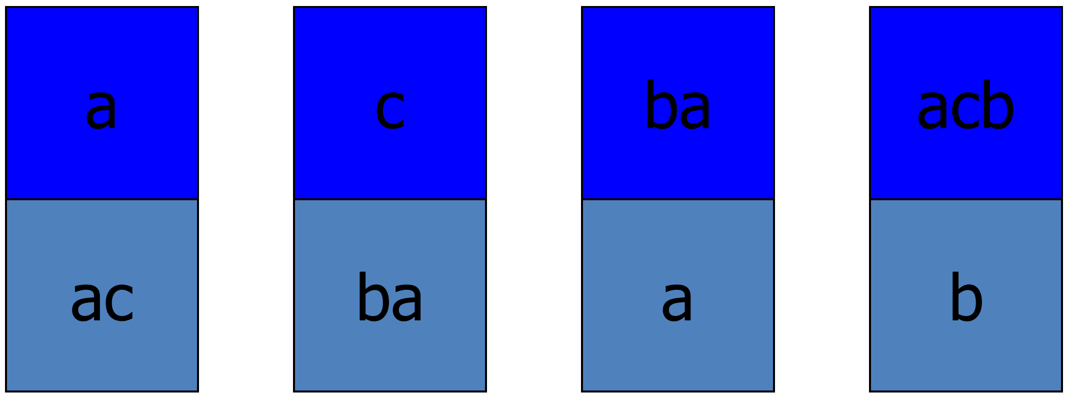
Podemos representar dominós como strings unários:



O objetivo do jogo é então obter o mesmo string na parte de cima e na parte de baixo.

Problema de Correspondência de Post (Dominó)

Ou podemos usar strings arbitrários. EX:



Obj: Obter mesmo string em cima e embaixo

PCP: Dado um alfabeto Σ e um conjunto finito de pares de strings $(u_1,v_1), (u_2,v_2), \ldots , (u_n,v_n)$ onde $u_i ,v_i \in \Sigma^*$, existe uma sequência não vazia de indices $i_1, i_2, i_3, \ldots , i_t$ tal que $u_{i1}u_{i2}u_{i3}\ldots u_{it} = v_{i1}v_{i2}v_{i3}\ldots v_{it}$?

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 4 protótipos:

1:		, 2:		, 3:		, 4:	
	a		c		ba		acb
	ac		ba		a		b

Total
Total
Indices

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 4 protótipos:

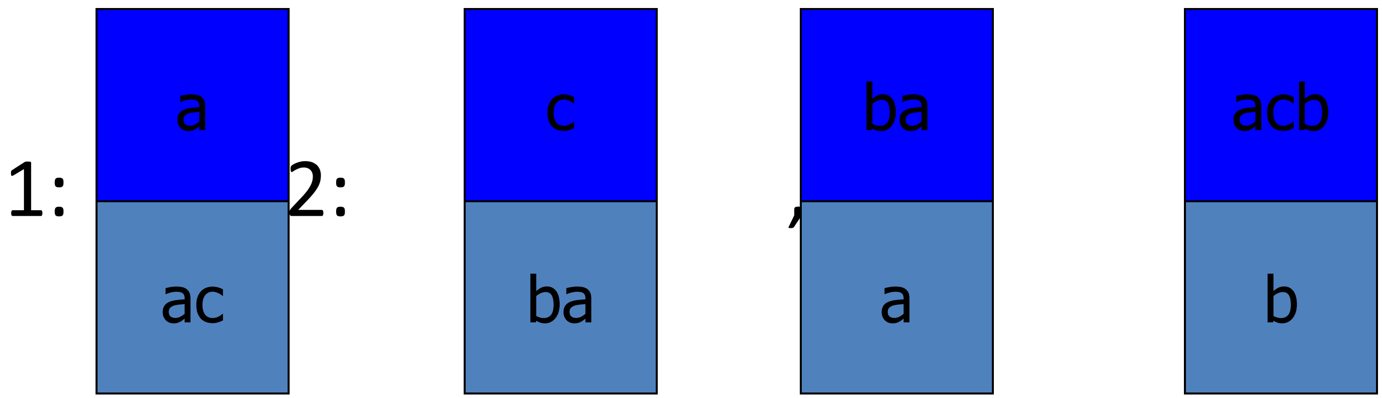
1:	2:	,	
a	c	ba	acb
ac	ba	a	b

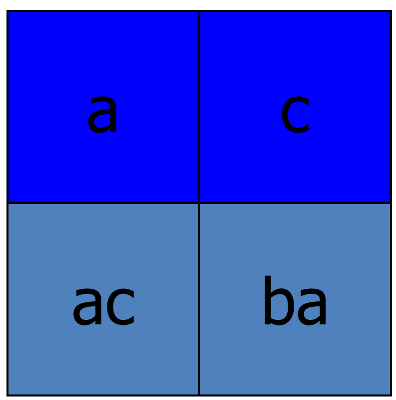
a
ac

Total
a
Total
ac
Indices
1

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 4 protótipos:





Total
ac
Total
acba
Indices
12

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 4 protótipos:

1:		2:		,		
	a		c		ba	acb
	ac		ba		a	b

a	c	ba
ac	ba	a

Total
acba
Total
acbaa
Indices
123

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 4 protótipos:

1: 2: ,

a	c	ba	acb
ac	ba	a	b

a	c	ba	a
ac	ba	a	ac

Total
acbaa
Total
acbaaac
Indices
1231

Problema de Correspondência de Post (Dominó)

Vamos jogar com os 4 protótipos:

1:	2:	,	
a	c	ba	acb
ac	ba	a	b

a	c	ba	a	acb
ac	ba	a	ac	b

Total
acbaaacb
Total
acbaaacb
Indices
12314

Resposta: SIM! (solução: 12314)

Redução: Resultados

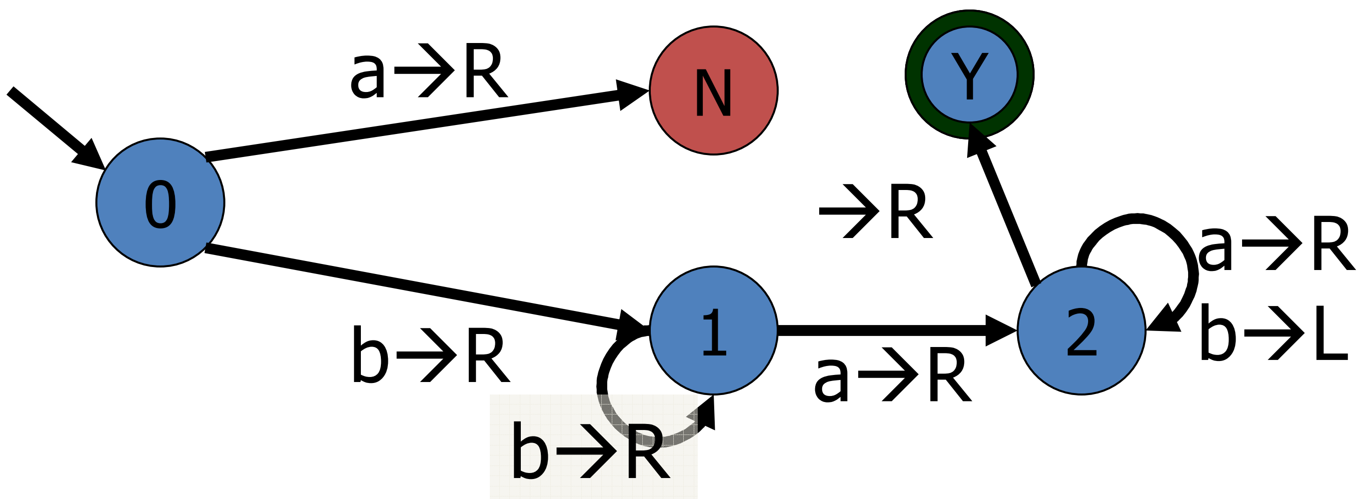
THM: $A_{\mathrm{TM}} \rightarrow_{+}$ STP $\rightarrow_{+}$ PCP. Portanto, STP e PCP não são decidíveis (de fato, não co-reconhecíveis). Além disso, como STP $\rightarrow_{+} A_{\mathrm{UG}}$, segue que A_{UG} não é decidível (e não reconhecível).

NOTA: PCP unário é decidível!

$A_{TM} \rightarrow_{+}$ STP

Toda TM pode ser transformada em um sistema semi-Thue que simula a TM. A idéia é codificar os movimentos da TM por regras. As regras transformam strings que representam configurações da TM. Assim, uma instância de A_{TM} é convertida em uma instância de STP se perguntarmos se uma dada configuração inicial deriva uma configuração de aceitação.

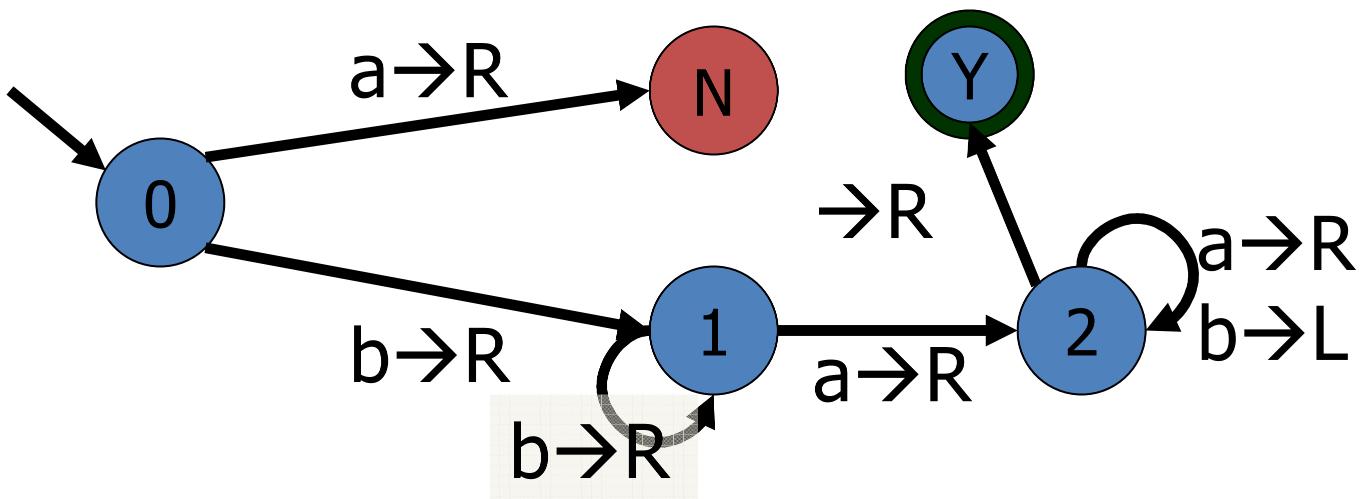
$A_{TM} \rightarrow_{+}$ STP



EX: Considere a TM acima.

O alfabeto Σ do sistema semi-Thue consiste do alfabeto de fita Γ, juntamente com colchetes '[' e ']' para marcar "início" e "fim" (i.e. início de tudo branco) da fita, e dos estados da TM – de modo a guardar a configuração.

$A_{TM} \rightarrow_{+} STP$



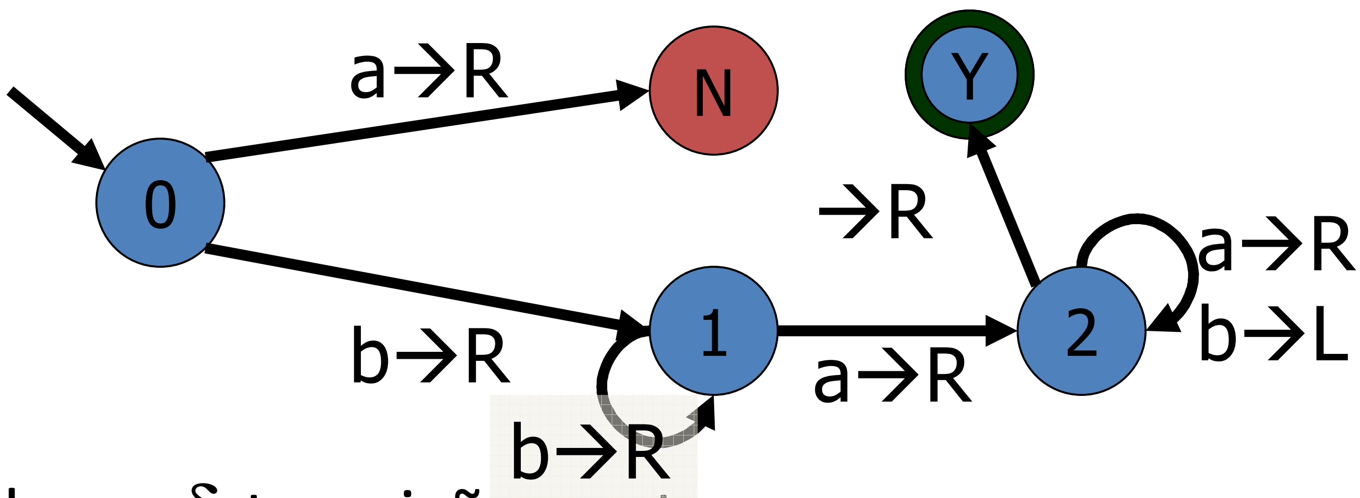
Alfabeto Σ = {a,b,,[,],0,1,2,Y,N}

Regras simulam transformações de configuração da TM. E.X. veja como $\delta(0,a)=(N,a,R)$ transforma a configuração:

$$u0av \rightarrow uaNv$$

Portanto devemos ter uma regra 0a → aN

$A_{TM} \rightarrow_{+} STP$



Todas as δ-transições se tornam regras:

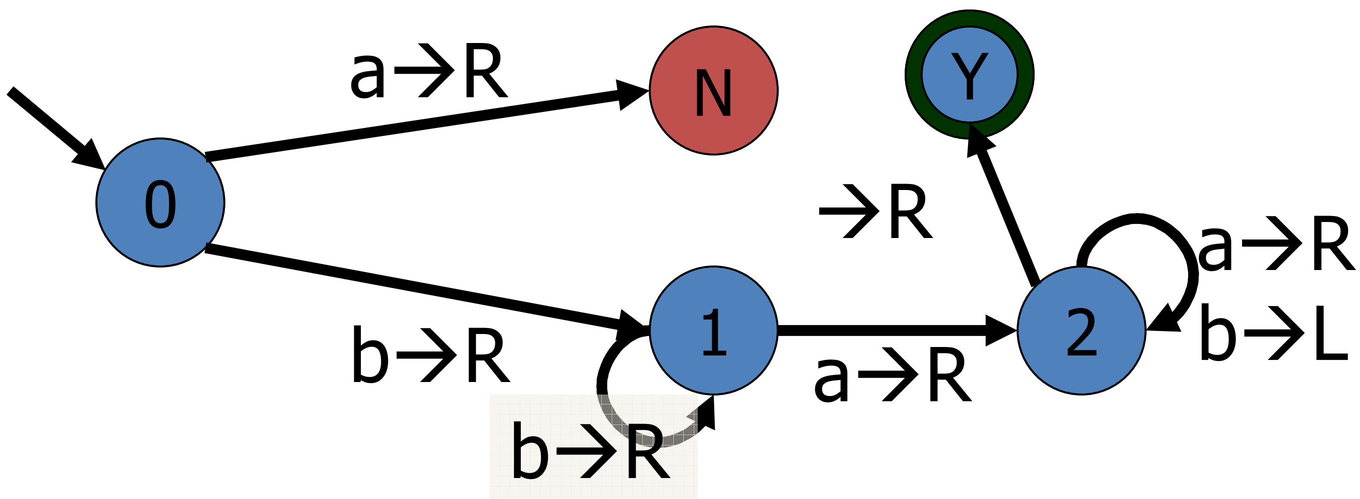
0a → aN, 0b → b1, 1a → a2, 1b → b1

2a → a2, a2b→2ab,b2b→2bb, 2b→2b

2→Y

Nota: movimentos p/ esq. são mais complexos do que p/ dir., já que precisamos escrever o símbolo à esquerda da cabeça da fita.

$A_{TM} \rightarrow_{+}$ STP

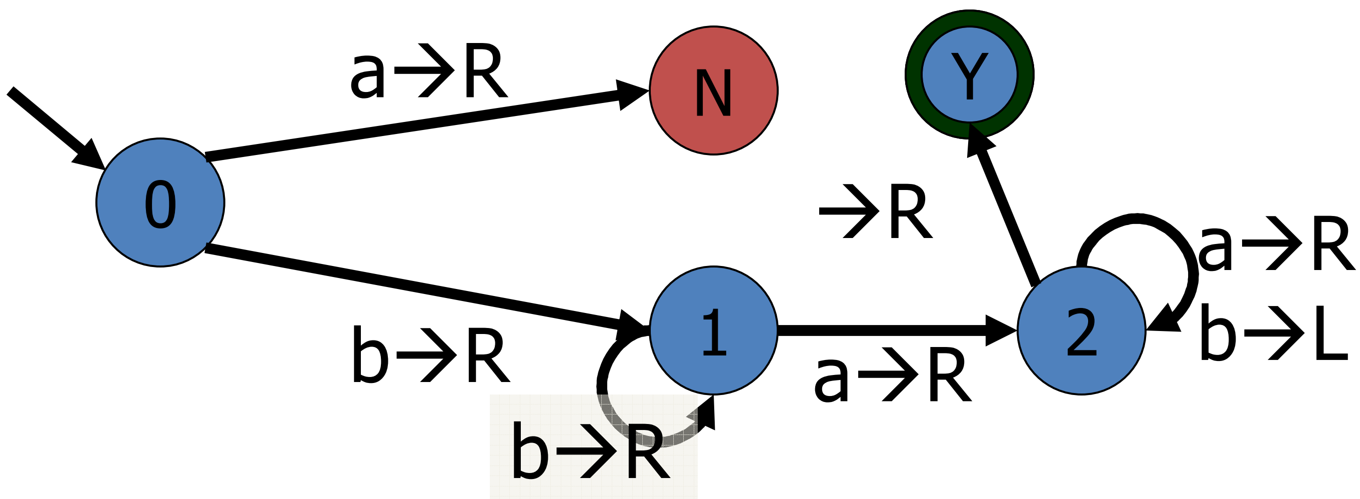


Temos também que criar brancos quando a máquina atinge o limite à direita:

0]→ 0],   1]→ 1],   2]→ 2],

E manter a cabeça parada ao atingir o limite à esquerda: [2b → [2b

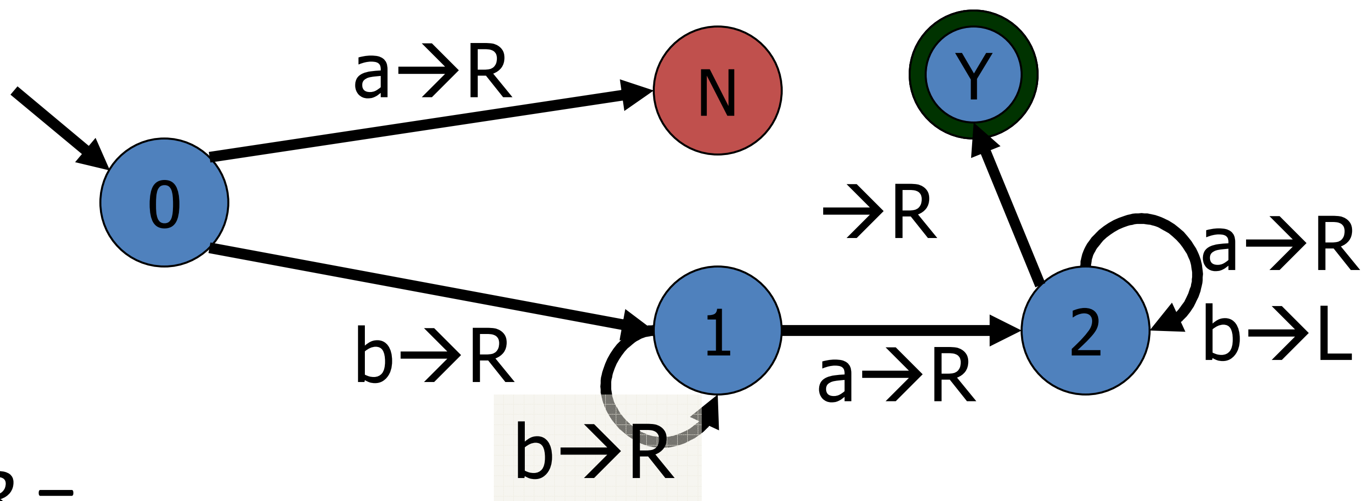
$A_{TM} \rightarrow_{+}$ STP



Finalmente, se "[Y]" é uma configuração de aceitação, então apagamos o conteúdo antes de aceitar:

Ya → Y, Yb → Y, Y→ Y,
aY → Y, bY → Y, Y→ Y

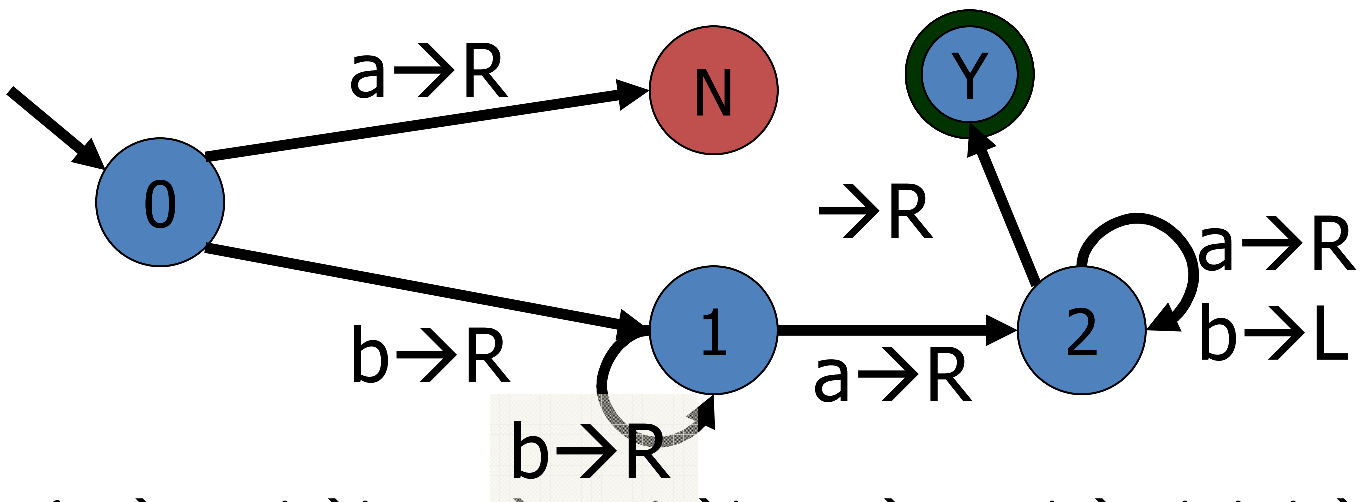
$A_{TM} \rightarrow_{+} \mathrm{STP}$



R =

{0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2,
a2b→2ab, b2b→2bb, 2b→2b,2→Y, 0]→0],
1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y,
aY→Y, bY→Y, Y→ Y}

$A_{TM} \rightarrow_{+}$ STP

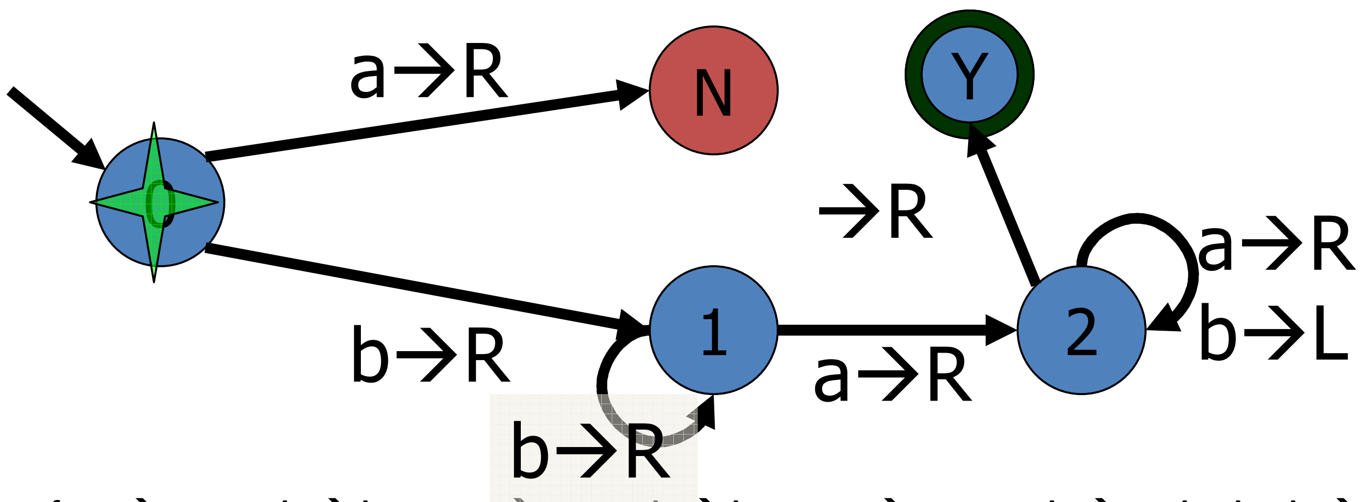


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

Perguntar se x é aceito pela TM equivale a perguntar se $[0x] \Rightarrow^* [Y]$

$A_{TM} \rightarrow_+ STP$

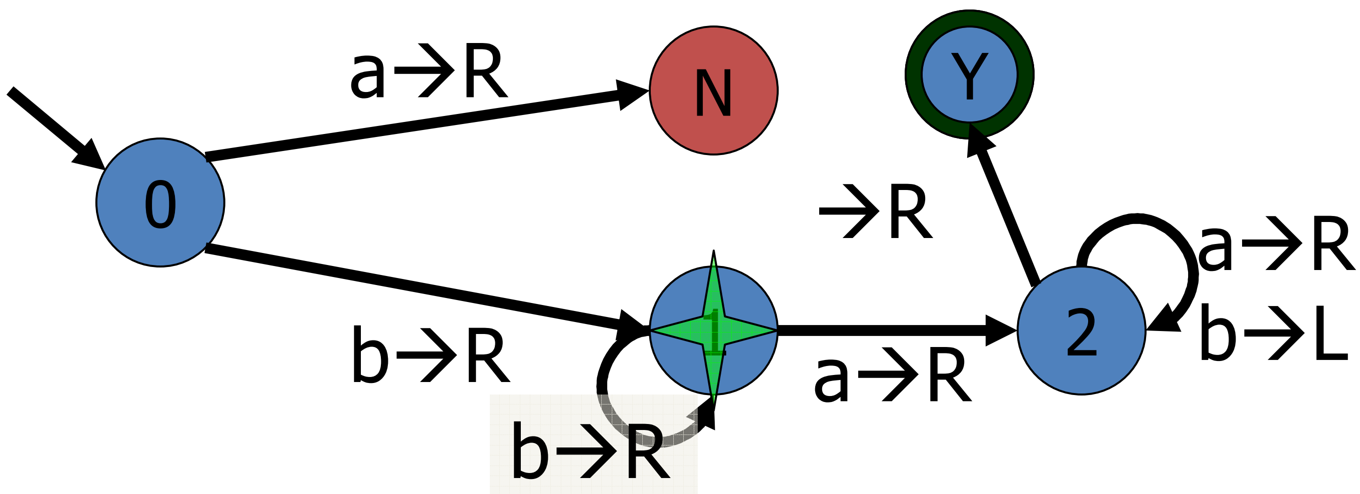


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada *x* = bbaaa

[0bbaaa]

$A_{TM} \rightarrow_{+}$ STP

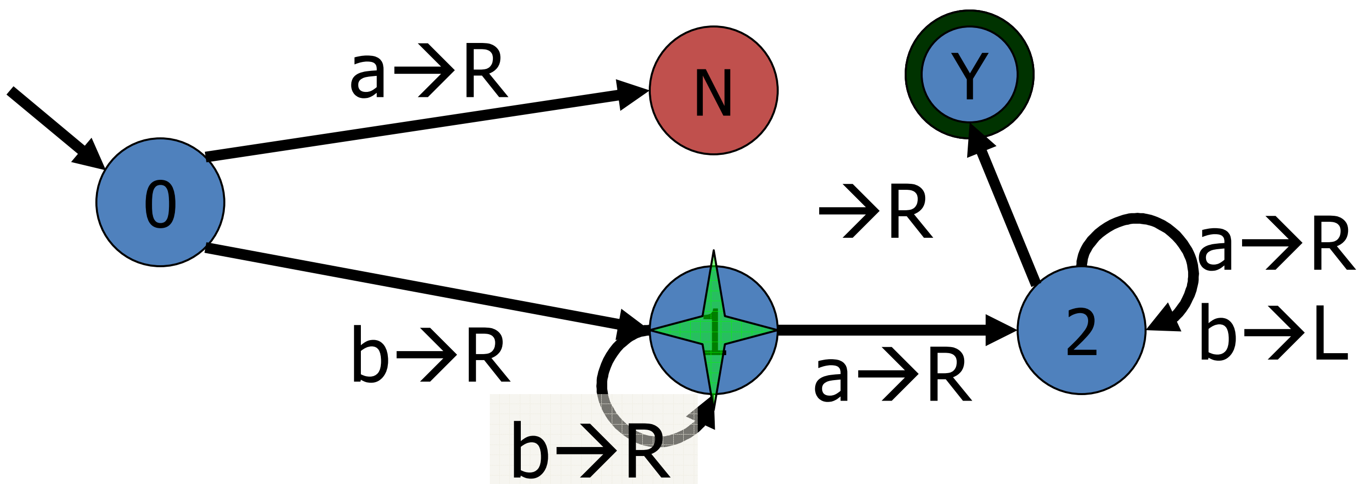


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada *x* = bbaaa

[0bbaaa] ⇒ [b1baaa]

$A_{TM} \rightarrow_{+} STP$

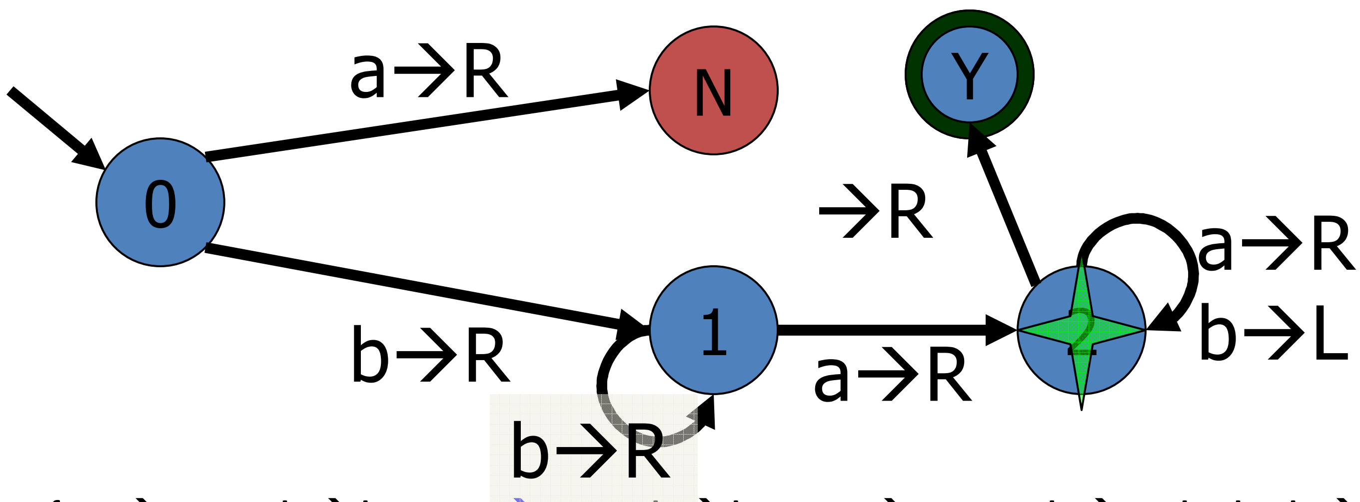


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] $\Rightarrow$ [b1baaa] $\Rightarrow$ [bb1aaa]

$A_{TM} \rightarrow_{+}$ STP

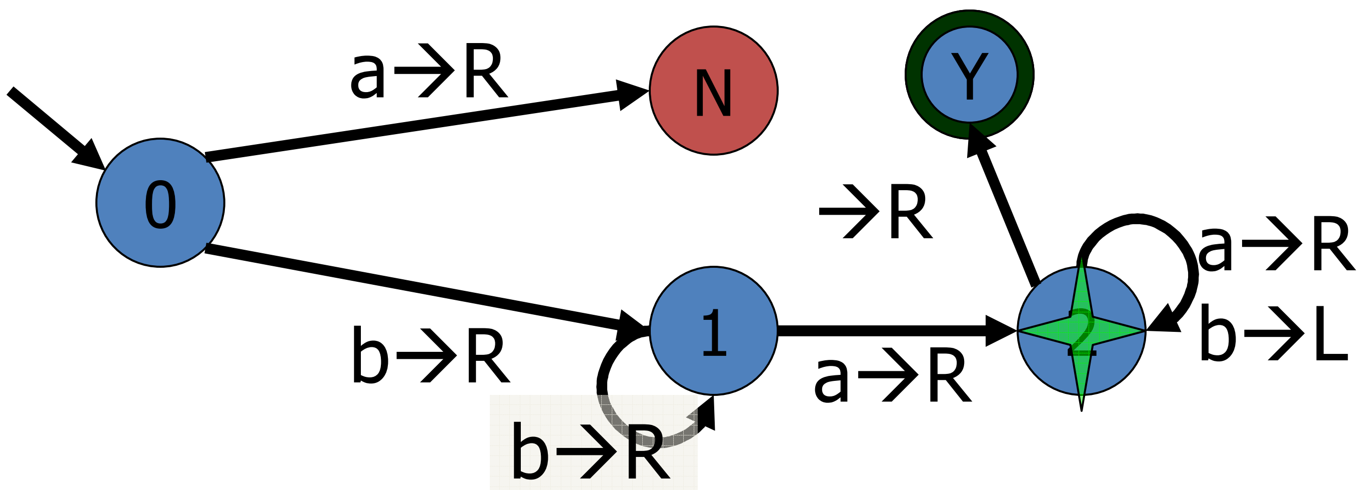


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa]

$A_{TM} \rightarrow_+ STP$

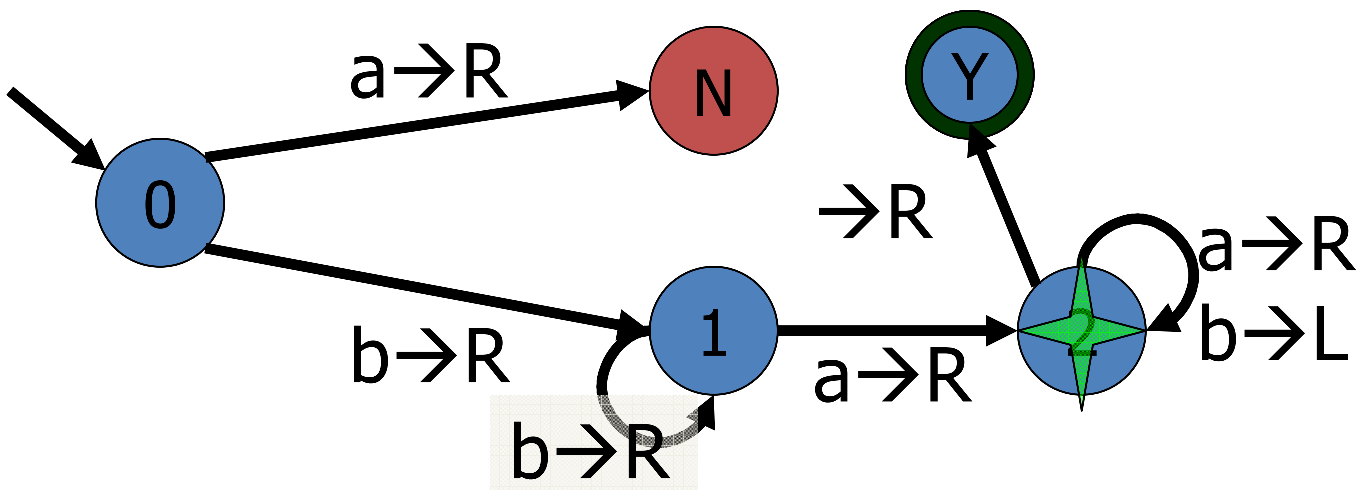


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada *x* = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a]

$A_{TM} \rightarrow_{+}$ STP

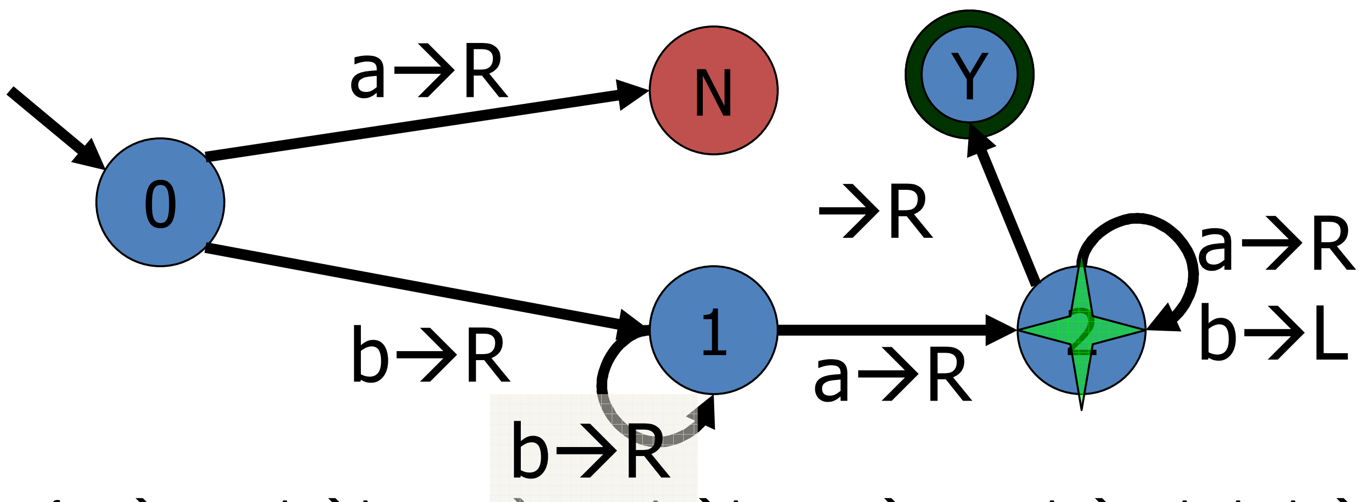


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2]

$A_{TM} \rightarrow_{+}$ STP

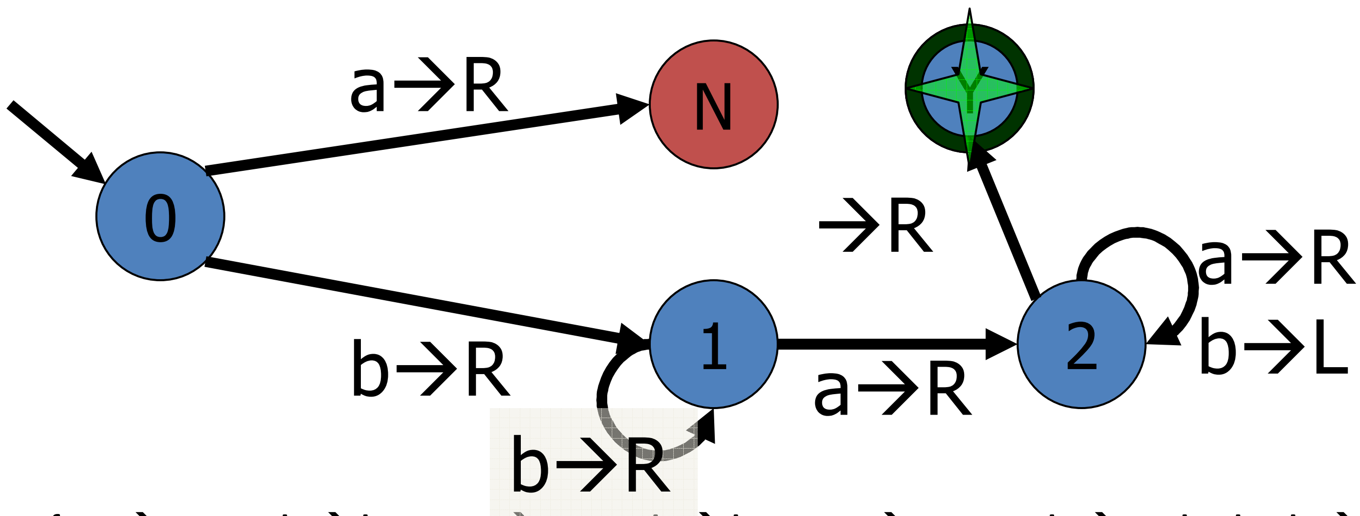


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]

$A_{TM} \rightarrow_+ STP$

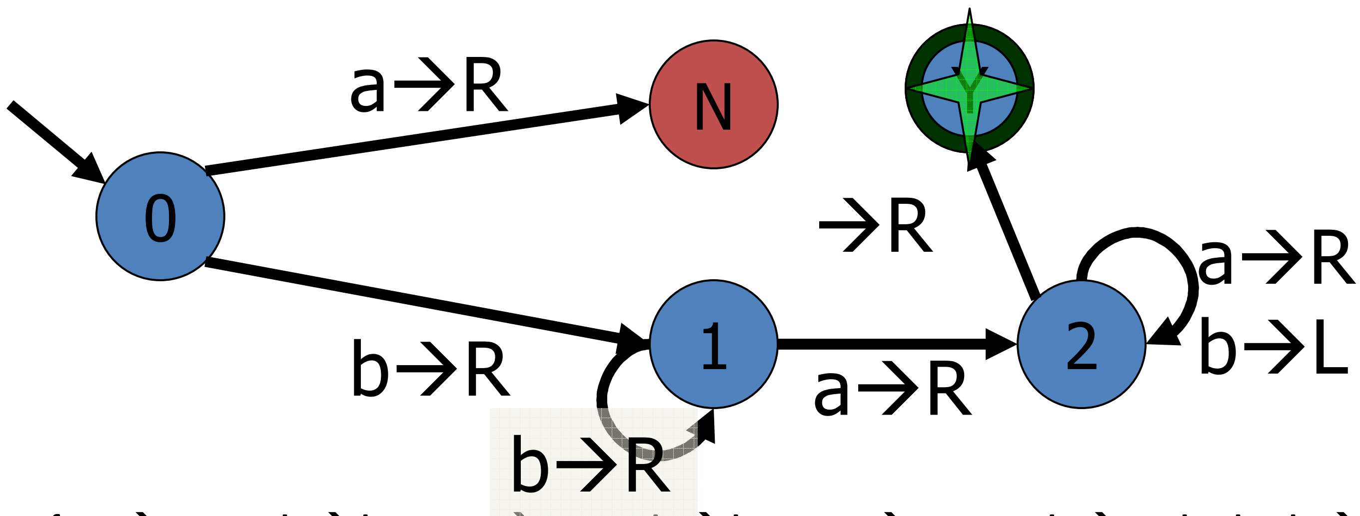


R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada *x* = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

$A_{TM} \rightarrow_{+}$ STP



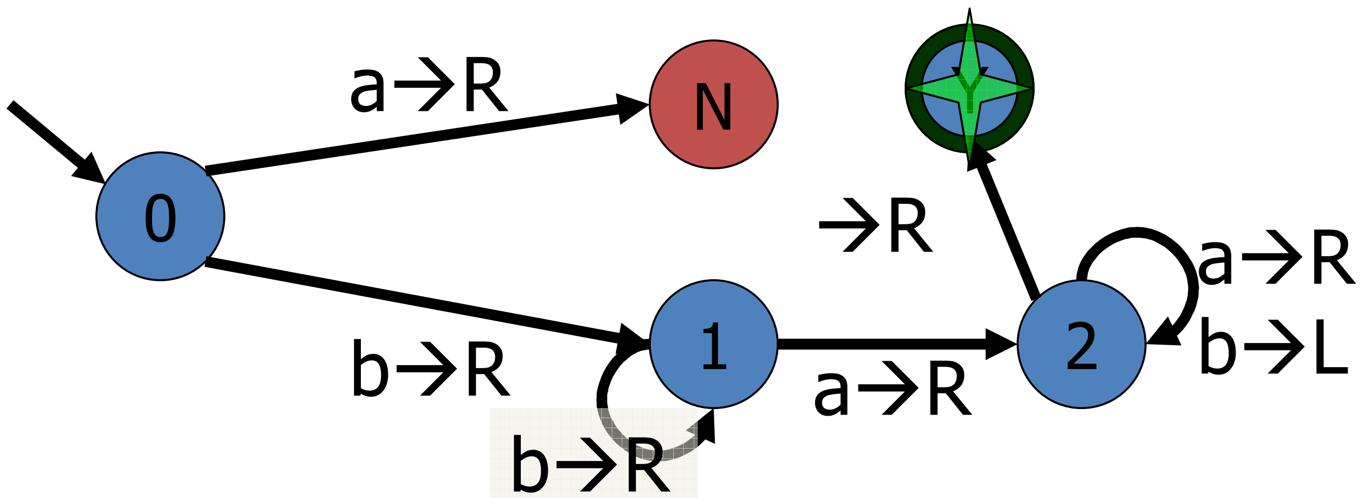
R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada *x* = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

⇒[bbaaaY]

$A_{TM} \rightarrow_{+}$ STP



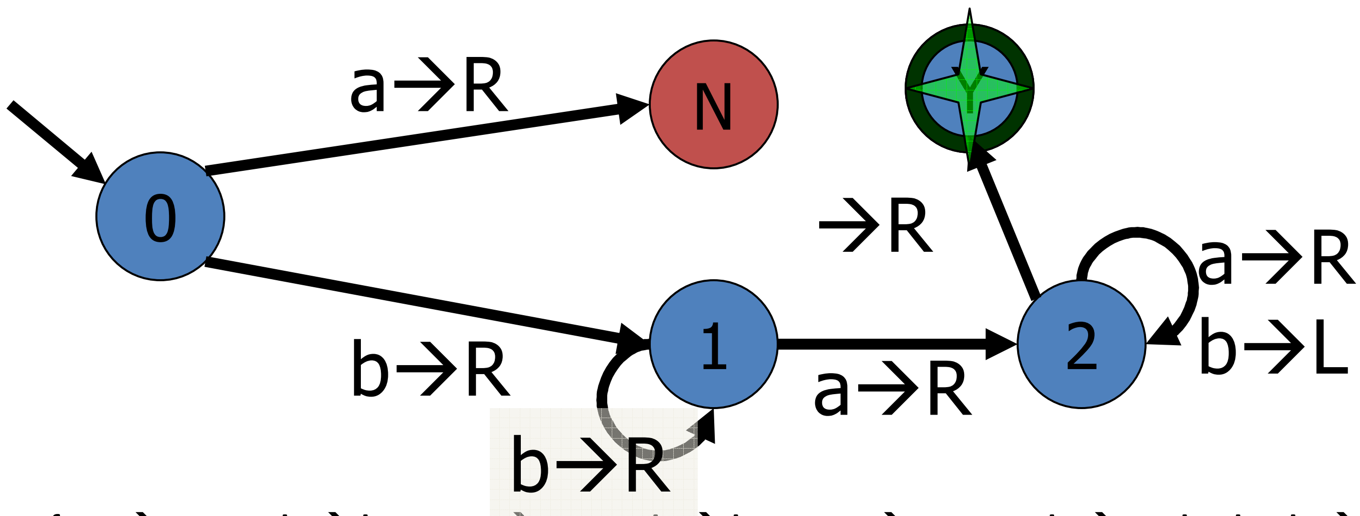
R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

⇒[bbaaaY]⇒[bbaaY]

$A_{TM} \rightarrow_+ \text{STP}$



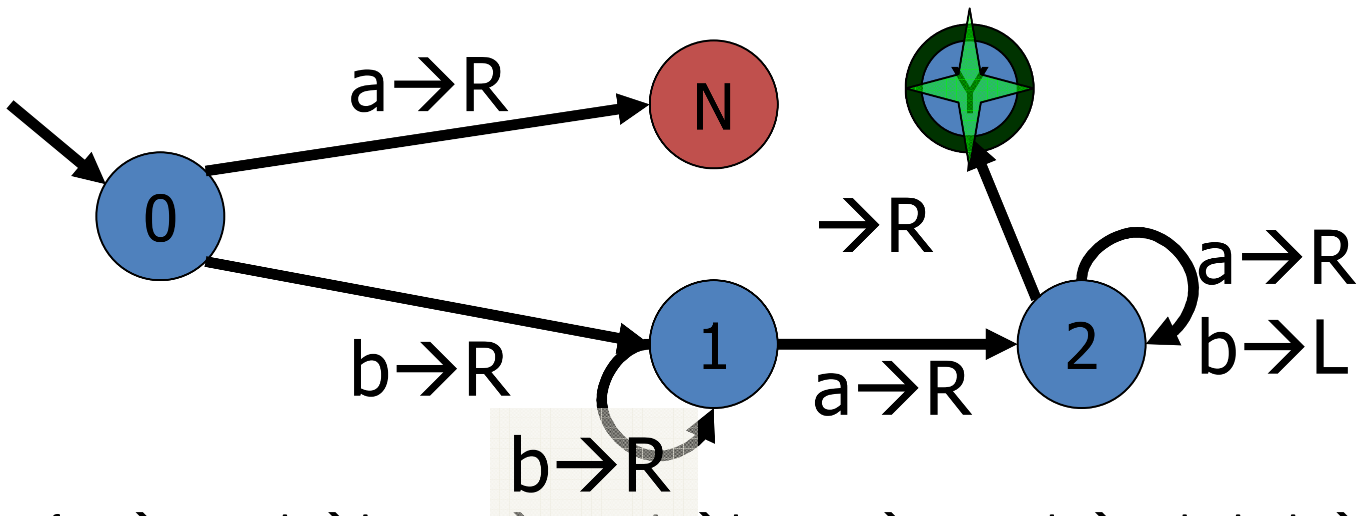
R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

⇒[bbaaaY]⇒[bbaaY]⇒[bbaY]

$A_{TM} \rightarrow_{+}$ STP



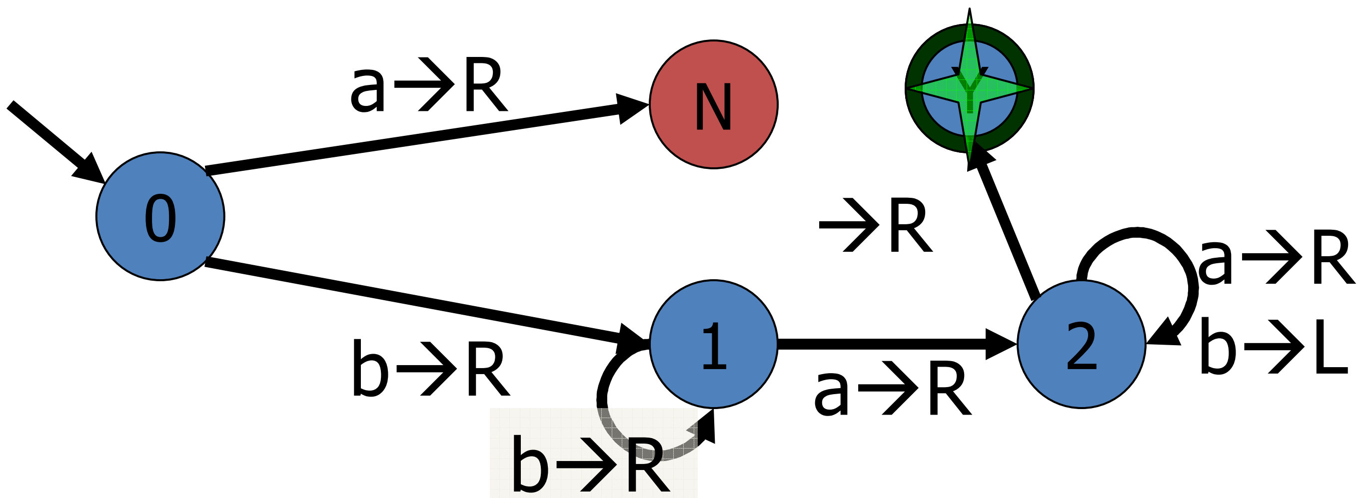
R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

⇒[bbaaaY]⇒[bbaaY]⇒[bbaY]⇒[bbY]

$A_{TM} \rightarrow_+ STP$



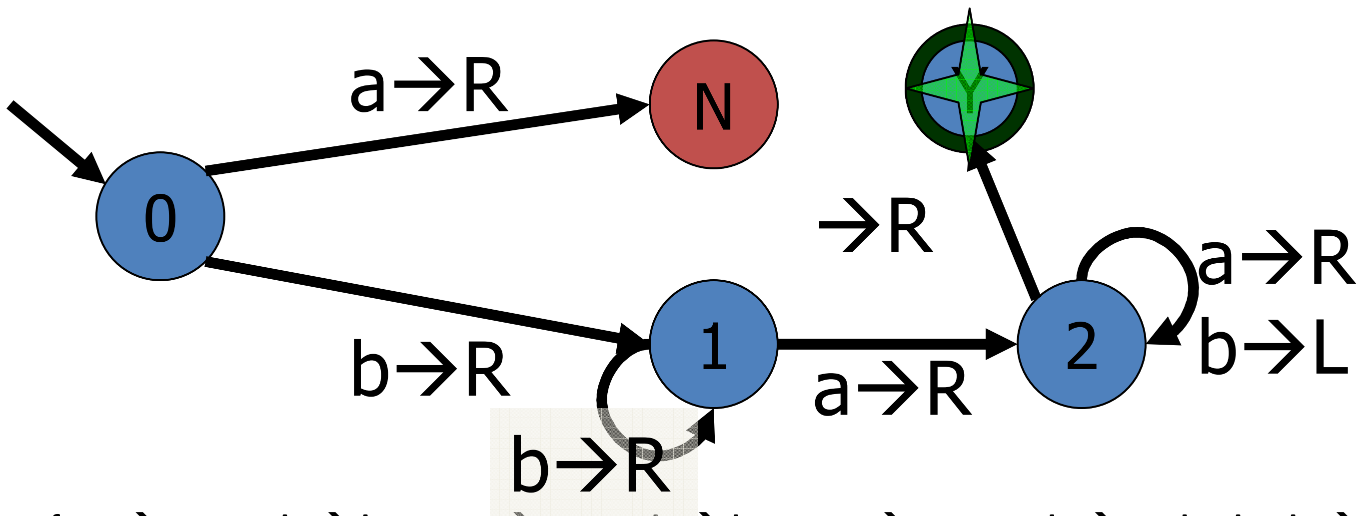
R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada x = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

⇒[bbaaaY]⇒[bbaaY]⇒[bbaY]⇒[bbY]⇒[bY]

$A_{TM} \rightarrow_{+}$ STP



R = {0a→aN, 0b→b1, 1a→a2, 1b→b1, 2a→a2, a2b→2ab, b2b→2bb, 2b→2b, 2→Y, 0]→0], 1]→1], 2]→2], [2b→[2b, Ya→Y, Yb→ Y, Y→Y, aY→Y, bY→Y, Y→ Y}

EX: Considere a entrada *x* = bbaaa

[0bbaaa] ⇒ [b1baaa] ⇒ [bb1aaa]⇒ [bba2aa] ⇒ [bbaa2a] ⇒ [bbaaa2] ⇒ [bbaaa2]⇒ [bbaaaY]

⇒[bbaaaY]⇒[bbaaY]⇒[bbaY]⇒[bbY]⇒[bY]⇒[Y]

$A_{TM} \rightarrow_{+}$ STP
Redução – caso geral

Perguntar se TM $(Q,S,G,d,q_0,q_{rej},q_{acc})$ aceita x é o mesmo que perguntar se $[0x] \Rightarrow^* [q_{acc}]$ no STP com as regras R: (q,q_1,q_2 são estados arbitrários)

$$q_1 a \rightarrow b q_2 \text{ when } \delta(q_1, a) = (q_2, b, \mathrm{R})$$

$$x q_1 a \rightarrow q_2 x b \ \ \forall x \in \Gamma, \text{ when } \delta(q_1, a) = (q_2, b, \mathrm{L})$$

$$[q_1 a \rightarrow [q_2 b \ \text{ whenever } \delta(q_1, a) = (q_2, b, \mathrm{L})$$

$$q] \rightarrow q\text{O}]$$

$$q_{\mathrm{acc}} x \rightarrow q_{\mathrm{acc}} \quad \text{and} \quad x q_{\mathrm{acc}} \rightarrow q_{\mathrm{acc}} \quad \forall x \in \Gamma$$

STP $\rightarrow_{+}$ PCP

A última parte da prova do teorema é mostrar que qualquer instância de sistema semi-Thue pode ser convertida em um problema de correspondência de Post.

Suponha que estamos tentando determinar se $x \Rightarrow^* y$ no sistema semi-Thue. Então, no PCP temos um dominó para cada regra do SST; começamos com *x* na parte de cima e vamos “colar” dominós de modo a simular uma derivação no SST que gere y na parte de baixo. Vejamos um exemplo a seguir.

STP $\rightarrow_+$ PCP

EX: Considere o sistema semi-Thue:

$S \rightarrow aSbc$ $S \rightarrow \varepsilon$ $cb \rightarrow bc$

STP $\rightarrow_+$ PCP

EX: Considere o sistema semi-Thue:

S $\rightarrow$ aSbc   S $\rightarrow$ ε cb $\rightarrow$ bc

As produções são imitadas por:

aSbc	aSbc			bc	bc
S	S	S	S	cb	cb

Q: Esse PCP tem solução?

STP $\rightarrow_{+}$ PCP

EX: Considere o sistema semi-Thue:

$S \rightarrow aSbc \quad S \rightarrow \varepsilon \qquad\qquad cb \rightarrow bc$

As produções são imitadas por:

aSbc	aSbc			bc	bc
S	S	S	S	cb	cb

R: Esse PCP não tem solução, já que nenhum dominó começa com a mesma letra na parte de cima e na de baixo. Mas…

STP $\rightarrow_{+}$ PCP

EX: Considere o sistema semi-Thue:

S $\rightarrow$ aSbc   S $\rightarrow$ ε cb $\rightarrow$ bc

E queremos verificar se S$\Rightarrow$*aabbcc.
Adicionamos dominós para início, fim e bordas:

[S#	]	$\underline{\#}$	#
[	$\underline{\#}$aabbcc]	#	$\underline{\#}$

STP $\rightarrow_{+}$ PCP

EX: Considere o sistema semi-Thue:

S $\rightarrow$ aSbc   S $\rightarrow$ ε   cb $\rightarrow$ bc

Para que a derivação possa terminar com *y* sublinhado em baixo, incluimos inversores

$\underline{S}$	S	$\underline{a}$	a	$\underline{b}$	b	$\underline{c}$	c
S	$\underline{S}$	a	$\underline{a}$	b	$\underline{b}$	c	$\underline{c}$

STP $\rightarrow_{+}$ PCP

Portanto, se a instância de STP é

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

É transformado na instância de PCP instance:

aSbc	aSbc			bc	bc
S	S	S	S	cb	cb

[S#	]	#	#
[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

STP $\rightarrow_{+}$ PCP

$R=\{S\rightarrow aSbc, S\rightarrow \varepsilon, cb\rightarrow bc\}$, $x=S$, $y=aabbcc$

$\underline{aSbc}$	$aSbc$			$\underline{bc}$	bc	$[S\#$	$]$	$\underline{\#}$	$\#$
S	$\underline{S}$	S	$\underline{S}$	cb	$\underline{cb}$	$[$	$\underline{\#}aabbcc]$	$\#$	$\underline{\#}$

$\underline{S}$	S	$\underline{a}$	a	$\underline{b}$	b	$\underline{c}$	c
S	$\underline{S}$	a	$\underline{a}$	b	$\underline{b}$	c	$\underline{c}$

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

Total da execução

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S

[S#
[

Total da execução
[S#
[S

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S $\Rightarrow$ aSbc

[S#	aSbc
[	S

Total da execução
[S#aSbc
[S

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

$\underline{\text{aSbc}}$	aSbc			$\underline{\text{bc}}$	bc	[S#	]	$\underline{\#}$	#
S	$\underline{\text{S}}$	S	$\underline{\text{S}}$	cb	$\underline{\text{cb}}$	[	$\underline{\#}$aabbcc]	#	$\underline{\#}$

$\underline{\text{S}}$	S	$\underline{\text{a}}$	a	$\underline{\text{b}}$	b	$\underline{\text{c}}$	c
S	$\underline{\text{S}}$	a	$\underline{\text{a}}$	b	$\underline{\text{b}}$	c	$\underline{\text{c}}$

S $\Rightarrow$ aSbc

[S#	$\underline{\text{aSbc}}$	$\underline{\#}$
[	S	#

Total da execução
[S#$\underline{\text{aSbc\#}}$
[S#

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S $\Rightarrow$ aSbc

[S#	aSbc	#	a
[	S	#	a

Total da execução
[S#aSbc#a
[S#a

STP $\rightarrow_{+}$ PCP

R={S$\rightarrow$aSbc,S$\rightarrow\varepsilon$,cb$\rightarrow$bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S $\Rightarrow$ aSbc $\Rightarrow$ aaSbcbc

[S#	aSbc	#	a	aSbc
[	S	#	a	S

Total da execução
[S#aSbc#aaSbc
[S#aS

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S $\Rightarrow$ aSbc $\Rightarrow$ aaSbcbc

[S#	aSbc	#	a	aSbc	b
[	S	#	a	S	b

Total da execução
[S#aSbc#aaSbcb
[S#aSb

STP $\rightarrow_+$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc$

[S#	aSbc	#	a	aSbc	b	c	
[	S	#	a	S	b	c	

Total da execução
[S#aSbc#aaSbcbc
[S#aSbc

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc$

[S#	aSbc	#	a	aSbc	b	c	#
[	S	#	a	S	b	c	#

Total da execução
[S#aSbc#aaSbcbc#
[S#aSbc#

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S ⇒ aSbc ⇒ aaSbcbc

[S#	aSbc	#	a	aSbc	b	c	#	a
[	S	#	a	S	b	c	#	a

Total da execução
[S#aSbc#aaSbcbc#a
[S#aSbc#a

STP $\rightarrow_{+}$ PCP

$R=\{S\rightarrow aSbc, S\rightarrow \varepsilon, cb\rightarrow bc\}$, $x=S$, $y=aabbcc$

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc$

[S#	aSbc	#	a	aSbc	b	c	#	a	a		
[	S	#	a	S	b	c	#	a	a		

Total da execução
[S#aSbc#aaSbcbc#aa
[S#aSbc#aa

STP $\rightarrow_+$ PCP

$R=\{S\rightarrow aSbc, S\rightarrow \varepsilon, cb\rightarrow bc\}$, $x=S$, $y=aabbcc$

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc \Rightarrow aabcbc$

[S#	aSbc	#	a	aSbc	b	c	#	a	a
[	S	#	a	S	b	c	#	a	a S

Total da execução
[S#aSbc#aaSbcbc#aa
[S#aSbc#aaS

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S $\Rightarrow$ aSbc $\Rightarrow$ aaSbcbc $\Rightarrow$ aabcbc

[S#	aSbc	#	a	aSbc	b	c	#	a	a		b
[	S	#	a	S		b	c	#	a	a S	b

Total da execução
[S#aSbc#aaSbcbc#aab
[S#aSbc#aaSb

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S $\Rightarrow$ aSbc $\Rightarrow$ aaSbcbc $\Rightarrow$ aabcbc

[S#	aSbc	#	a	aSbc	b	c	#	a	a		b	c
[	S	#	a	S	b	c	#	a	a	S	b	c

Total da execução
[S#aSbc#aaSbcbc#aabc
[S#aSbc#aaSbc

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc \Rightarrow aabcbc$

[S# aSbc # a aSbc b c # a a b c b
[S # a S b c # a a S b c b

Total da execução
[S#aSbc#aaSbcbc#aabcb
[S#aSbc#aaSbcb

STP $\rightarrow_+$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

$\underline{\text{aSbc}}$	aSbc			$\underline{\text{bc}}$	bc	[S#	]	$\underline{\text{\#}}$	#
S	$\underline{\text{S}}$	S	$\underline{\text{S}}$	cb	$\underline{\text{cb}}$	[	$\underline{\text{\#}}$aabbcc]	#	$\underline{\text{\#}}$

$\underline{\text{S}}$	S	$\underline{\text{a}}$	a	$\underline{\text{b}}$	b	$\underline{\text{c}}$	c
S	$\underline{\text{S}}$	a	$\underline{\text{a}}$	b	$\underline{\text{b}}$	c	$\underline{\text{c}}$

S $\Rightarrow$ aSbc $\Rightarrow$ aaSbcbc $\Rightarrow$ aabcbc

[S# $\underline{\text{aSbc}}$ $\underline{\text{\#}}$ a aSbc b c # $\underline{\text{a}}$ $\underline{\text{a}}$ $\underline{\text{b}}$ $\underline{\text{c}}$ $\underline{\text{b}}$ $\underline{\text{c}}$
[S # $\underline{\text{a}}$ $\underline{\text{S}}$ $\underline{\text{b}}$ $\underline{\text{c}}$ $\underline{\text{\#}}$ a a S b c b c

Total da execução
[S#$\underline{\text{aSbc\#}}$aaSbcbc#$\underline{\text{aabcbc}}$
[S#$\underline{\text{aSbc\#}}$aaSbcbc

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

$\underline{\text{aSbc}}$	aSbc			$\underline{\text{bc}}$	bc	[S#	]	$\underline{\#}$	#
S	$\underline{\text{S}}$	S	$\underline{\text{S}}$	cb	$\underline{\text{cb}}$	[	$\underline{\#}$aabbcc]	#	$\underline{\#}$

$\underline{\text{S}}$	S	$\underline{\text{a}}$	a	$\underline{\text{b}}$	b	$\underline{\text{c}}$	c
S	$\underline{\text{S}}$	a	$\underline{\text{a}}$	b	$\underline{\text{b}}$	c	$\underline{\text{c}}$

S ⇒ aSbc ⇒ aaSbcbc ⇒ aabcbc

[S#	$\underline{\text{aSbc}}$	$\underline{\#}$	a	aSbc	b	c	#	$\underline{\text{a}}$	$\underline{\text{a}}$		$\underline{\text{b}}$	$\underline{\text{c}}$	$\underline{\text{b}}$	$\underline{\text{c}}$	$\underline{\#}$
[	S	#	$\underline{\text{a}}$	$\underline{\text{S}}$	$\underline{\text{b}}$	$\underline{\text{c}}$	$\underline{\#}$	a	a	S	b	c	b	c	#

Total da execução
[S#$\underline{\text{aSbc\#}}$aaSbcbc#$\underline{\text{aabcbc\#}}$
[S#$\underline{\text{aSbc\#}}$aaSbcbc#

STP $\rightarrow_+$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

$\underline{aSbc}$	aSbc			$\underline{bc}$	bc	[S#	]	$\underline{\#}$	#
S	$\underline{S}$	S	$\underline{S}$	cb	$\underline{cb}$	[	$\underline{\#}$aabbcc]	#	$\underline{\#}$

$\underline{S}$	S	$\underline{a}$	a	$\underline{b}$	b	$\underline{c}$	c
S	$\underline{S}$	a	$\underline{a}$	b	$\underline{b}$	c	$\underline{c}$

$$S \Rightarrow aSbc \Rightarrow aaSbcbc \Rightarrow aabcbc$$

[S# $\underline{aSbc}$ $\underline{\#}$ a aSbc b c # $\underline{a}$ $\underline{a}$ $\underline{b}$ $\underline{c}$ $\underline{b}$ $\underline{c}$ $\underline{\#}$ a

[S # $\underline{a}$ $\underline{S}$ $\underline{b}$ $\underline{c}$ $\underline{\#}$ a a S b c b c # $\underline{a}$

Total da execução
[S#$\underline{aSbc\#}$aaSbcbc#$\underline{aabcbc\#}$a
[S#$\underline{aSbc\#}$aaSbcbc#$\underline{a}$

STP $\rightarrow_+$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S ⇒ aSbc ⇒ aaSbcbc ⇒ aabcbc

[S#	aSbc	#	a	aSbc	b	c	#	a	a		b	c	b	c	#	a	a
[	S	#	a	S	b	c	#	a	a	S	b	c	b	c	#	a	a

Total da execução
[S#aSbc#aaSbcbc#aabcbc#aa
[S#aSbc#aaSbcbc#aa

STP $\rightarrow_{+}$ PCP

$R=\{S\rightarrow aSbc, S\rightarrow \varepsilon, cb\rightarrow bc\}$, $x=S$, $y=aabbcc$

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc \Rightarrow aabcbc$

[S#	aSbc	#	a	aSbc	b	c	#	a	a		b	c	b	c	#	a	a	b
[	S	#	a	S	b	c	#	a	a	S	b	c	b	c	#	a	a	b

Total da execução
[S#aSbc#aaSbcbc#aabcbc#aab
[S#aSbc#aaSbcbc#aab

STP $\to_+$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S ⇒ aSbc ⇒ aaSbcbc ⇒ aabcbc ⇒ aabbcc

[S# aSbc # a aSbc b c # a a b c b c # a a b bc
[S # a S b c # a a S b c b c # a a b cb

Total da execução
[S#aSbc#aaSbcbc#aabcbc#aabbc
[S#aSbc#aaSbcbc#aabcb

STP $\rightarrow_{+}$ PCP

R={S→aSbc,S→ε,cb→bc}, x=S, y=aabbcc

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

S ⇒ aSbc ⇒ aaSbcbc ⇒ aabcbc ⇒ aabbcc

[S#	aSbc	#	a	aSbc	b	c	#	a	a		b	c	b	c	#	a	a	b	bc	c
[	S	#	a	S	b	c	#	a	a	S	b	c	b	c	#	a	a	b	cb	c

Total da execução
[S#aSbc#aaSbcbc#aabcbc#aabbcc
[S#aSbc#aaSbcbc#aabcbc

STP $\rightarrow_{+}$ PCP

$R=\{S \rightarrow aSbc, S \rightarrow \varepsilon, cb \rightarrow bc\}$, $x=S$, $y=aabbcc$

aSbc	aSbc			bc	bc	[S#	]	#	#
S	S	S	S	cb	cb	[	#aabbcc]	#	#

S	S	a	a	b	b	c	c
S	S	a	a	b	b	c	c

$S \Rightarrow aSbc \Rightarrow aaSbcbc \Rightarrow aabcbc \Rightarrow aabbcc$

[S# aSbc # a aSbc b c # a a b c b c # a a b bc c]
[S # a S b c # a a S b c b c # a a b cb c #aabbcc]

Total da execução
[S#aSbc#aaSbcbc#aabcbc#aabbcc]
[S#aSbc#aaSbcbc#aabcbc#aabbcc]

STP $\rightarrow_{+}$ PCP
Redução Geral

De modo geral, um STP (Σ,R,x,y) [perguntando $x \Rightarrow^* y$?] se torna o PCP:

$\dfrac{\underline{v}}{u}$ $\dfrac{v}{\underline{u}}$ para toda regra $u \rightarrow v$ em R

$\dfrac{\underline{a}}{a}$ $\dfrac{a}{\underline{a}}$ para todo símbolo $a \in \Sigma \cup \{\#\}$

$\dfrac{[x\#}{[}$ –regra inicial, e $\dfrac{]}{\underline{\#}y]}$ –regra final.

Teorema Concluído!

Isso completa a prova de

THM: $A_{TM} \rightarrow_{+}$ STP $\rightarrow_{+}$ PCP. Portanto, STP e PCP não são decidíveis (de fato, não co-reconhecíveis). Além disso, como STP $\rightarrow_{+} A_{UG}$, segue que A_{UG} não é decidível (não reconhecível).

Problemas Não decidíveis sobre CFG's

De fato, qualqer PCP pode ser simulado por *duas* CFG's trabalhando em conjunto de um certo modo. Isso permite provar alguns resultados de não decidibilidade de problemas relativos a linguagens livres de contexto

Representando PCP por Duas CFG's

O PCP:

	a	c	ba	acb
1:	ac	ba	a	b

é representado por duas gramáticas. Ambas têm terminais {1,2,3,4,a,b,c} obtidos dos *possíveis indices* junto com o alfabeto do PCP, e conj. de variáveis {S}

As duas gramáticas são chamadas TOP e BOTTOM sugestivamente:

Representando PCP por Duas CFG's

O PCP:

	1:	2:	3:	4:
	a	c	ba	acb
	ac	ba	a	b

Regras TOP:

$$S \rightarrow 1Sa \mid 2Sc \mid 3Sab \mid 4Sbca \mid \varepsilon$$

Regras BOTTOM:

$$S \rightarrow 1Sca \mid 2Sab \mid 3Sa \mid 4Sb \mid \varepsilon$$

O que está à esq. de S mantém informação sobre a sequência de dominós. O que está à dir. é o reverso do total computado.

Representando PCP por Duas CFG's

The PCP:

	a	c	ba	acb
1:	ac	ba	a	b

EX: A solução 12314 significa que o string 12314bcaaabca é gerado por ambas as gramáticas TOP e BOTTOM.

Representando PCP por Duas CFG's

De modo geral. Considere um PCP dado por dominós: (u_1,v_1), (u_2,v_2), … , (u_n,v_n).

TOP é definida por:

$S \rightarrow \varepsilon \mid 1Su_1^R \mid 2Su_2^R \mid \ldots \mid nSu_n^R$

BOTTOM é definida por:

$S \rightarrow \varepsilon \mid 1Sv_1^R \mid 2Sv_2^R \mid \ldots \mid nSv_n^R$

THM: O PCP tem uma solução sse TOP e BOTTOM geram algum string em comum. I.e:

$L(\text{TOP}) \cap L(\text{BOTTOM}) = \emptyset \Leftrightarrow$ PCP não tem solução

Leis de DeMorgan implicam que:

$\overline{L(\text{TOP})} \cup \overline{L(\text{BOTTOM})} = \Sigma^* \Leftrightarrow$ o PCP não tem solução

Não decidibilidade de ALL_{CFG}

Acontece que não apenas L(TOP) e L(BOTTOM) são livres de contexto: também o são os complementos dessas linguagens. Isso implica:

THM: ALL_{CFG} não é decidível.

Prova. Supondo o que se disse acima, e como CFL's são fechadas em relação a união, segue que $\overline{L(\text{TOP})} \cup \overline{L(\text{BOTTOM})}$ é livre de contexto. Mas essa linguagen é Σ^* sse o PCP não tem solução. Portanto, um algoritmo para ALL_{CFG} forneceria um algoritmo para PCP, o que é impossível.

Q: Existe uma prova mais simples? (rec./corec.?)

Não decidibilidade de ALL_{CFG}

R: Uma análise mais simples. Podemos obter soluções "sim" para PCP, mas soluções "não" são mais difíceis. Portanto PCP não é co-reconhecível. Solutions "sim" para ALL_{CFG} dariam soluções "não" para PCP, portanto ALL_{CFG} não é reconhecível.

Finalmente, o seguinte algoritmo mostra que ALL_{CFG} é co-reconhecível:

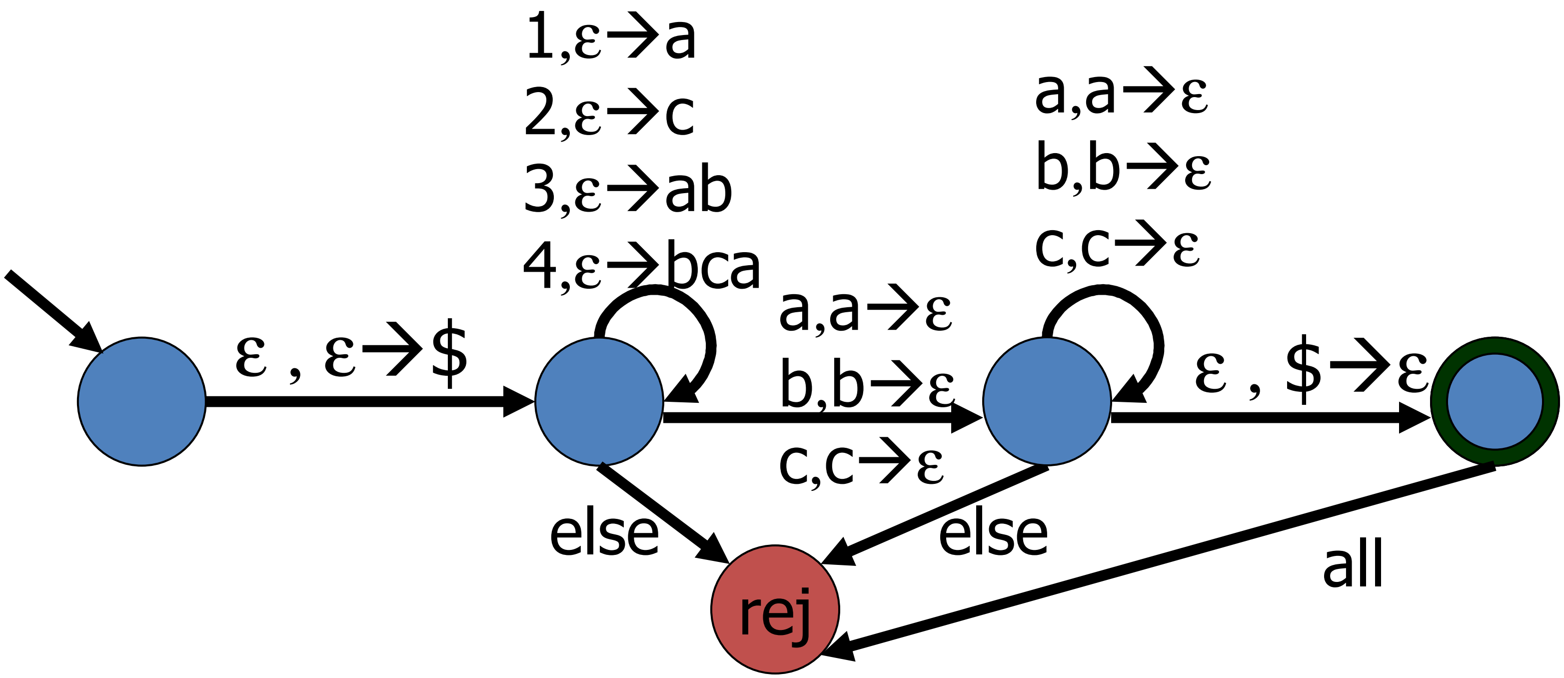
Sobre a entrada <*G*>

1. Para cada string, veja se é gerado por *G*.
2. Se algum não é gerado, "L(*G*) $\neq \Sigma^*$"

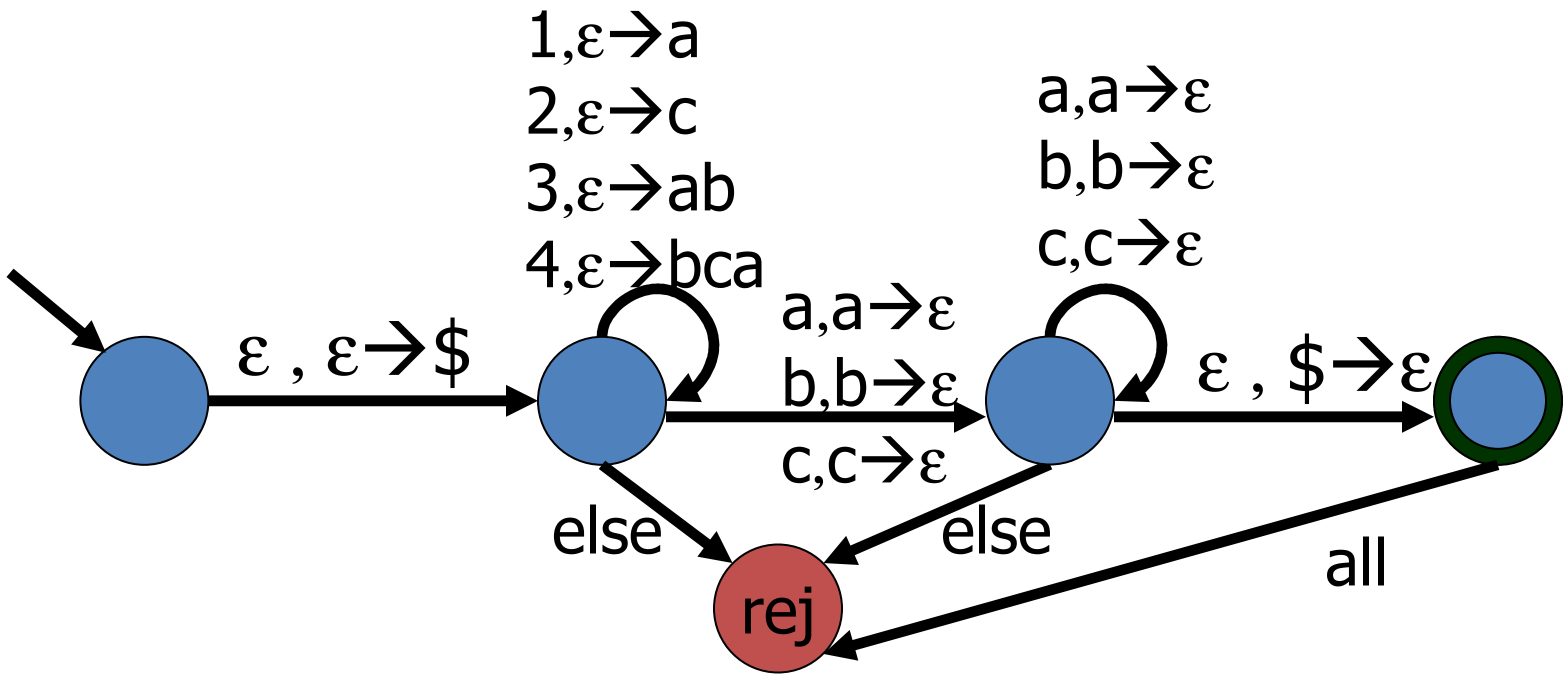
PDA para *L*(TOP)

Parte final: Precisamos mostrar que os complementos de *L*(TOP) e *L*(BOTTOM) são livres de contexto. É fácil ver isso pelo PDA:

EX: TOP = S→ 1Sa|2Sc|3Sab|4Sbca|ε



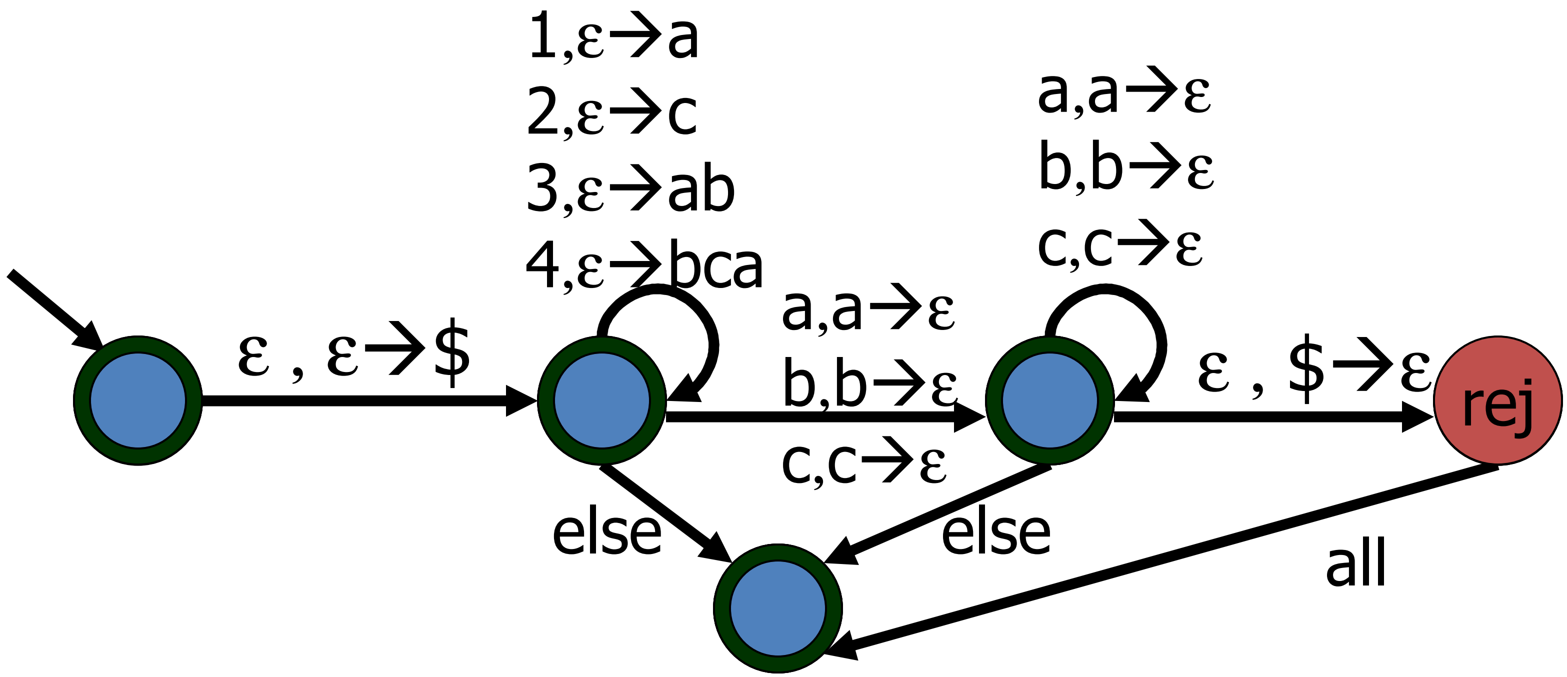
PDA para *L*(TOP)



Note que o PDA é determinista no sentido de que ele não permite falhas e toda entrada símbolo leva a um único estado final (exceto ε – um caso que é fácil tratar).

Q: Como podemos obter o complemento?

PDA p/ *L*(TOP): Complemento



R: Apenas inverta estados finais e não finais (e garanta que funciona no caso de ε). Portanto:

THM: *L*(TOP) e *L*(BOTTOM) são aceitas por PDA's deterministas. Consequentemente, seus complementos são livres de contexto.