
Verificação de Sistemas de Tempo Real com Autômatos Temporizados: Teoria e Prática

Prof. Guilherme A. Pinto

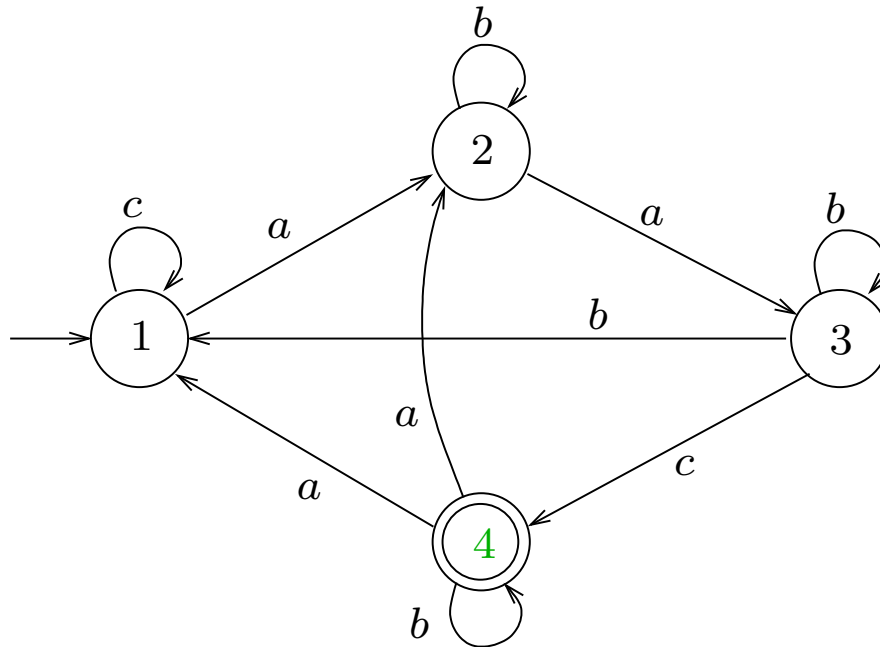
07 de dezembro de 2006

Departamento de Ciência da Computação – UnB

Roteiro

- Autômatos Temporizados e Verificação:
 - Autômato Finito \longrightarrow Autômato Temporizado
 - Composição e Verificação usando Linguagens
 - Alcançabilidade
- UPPAAL e Exemplo
 - O que dizer para o Engenheiro?
 - Interface e Exemplo
- Grau de Indecidibilidade:
 - Universalidade
 - Hierarquias de indecidibilidade

Autômato Finito



Alfabeto: $\Sigma = \{a, b, c\}$

Lugares: $Q = \{1, 2, 3, 4\}$

Lugares Iniciais: $Q_0 = \{1\}$

Lugares Finais: $F = \{4\}$

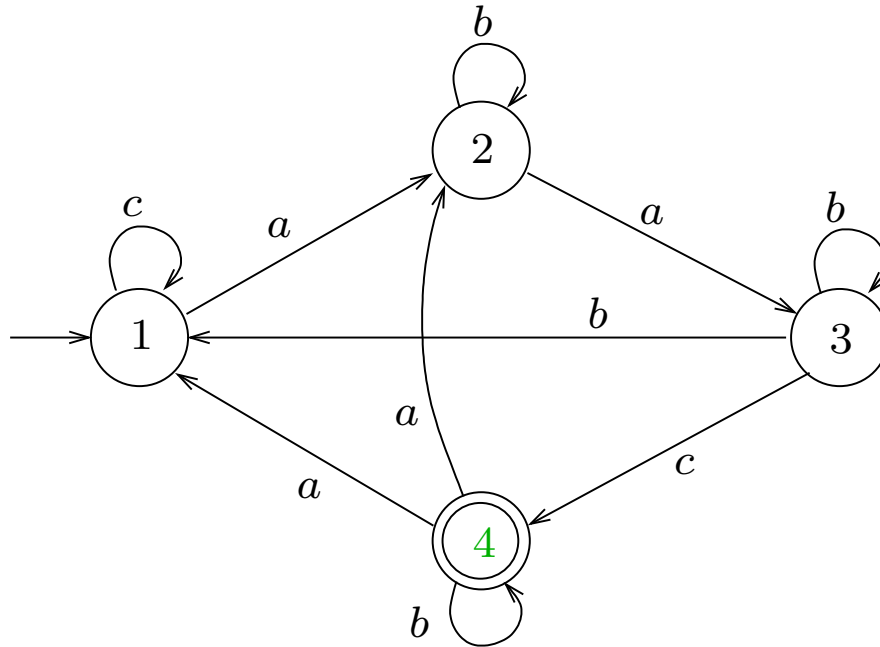
Transições: $T \subseteq Q \times \Sigma \times Q$

- Aceita palavras finitas: $\rho \in \Sigma^*$

$$\rho = ababc$$

$$\{1\} \xrightarrow{a} \{2\} \xrightarrow{b} \{2\} \xrightarrow{a} \{3\} \xrightarrow{b} \{3\} \xrightarrow{c} \{4\}$$

ω -Autômato



Alfabeto: $\Sigma = \{a, b, c\}$

Lugares: $Q = \{1, 2, 3, 4\}$

Lugares Iniciais: $Q_0 = \{1\}$

Lugares Finais: $F = \{4\}$

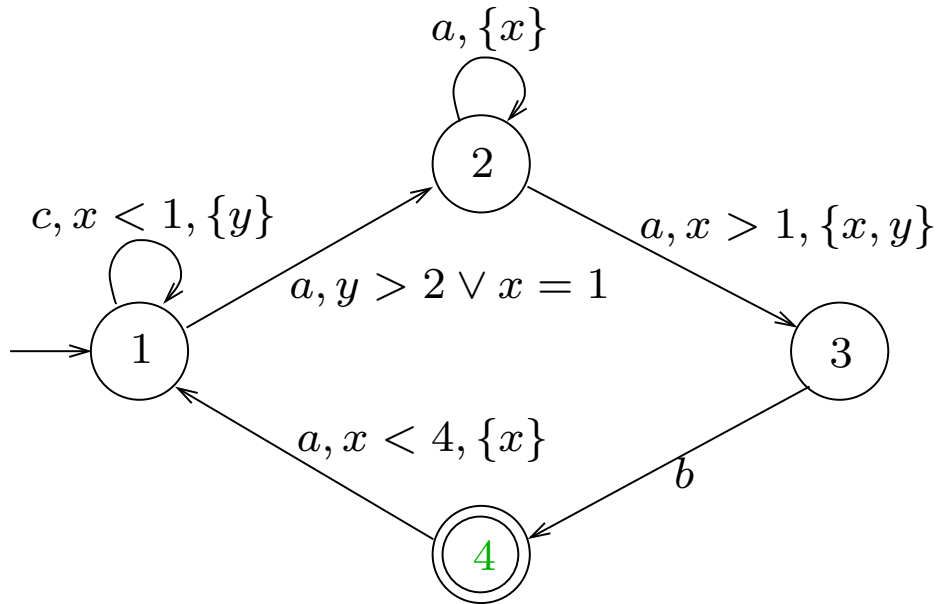
Transições: $T \subseteq Q \times \Sigma \times Q$

- Aceita palavras infinitas: $\rho \in \Sigma^\omega$

$$\rho = aacbaacbaac \dots$$

$$\{1\} \xrightarrow{a} \{2\} \xrightarrow{a} \{3\} \xrightarrow{c} \{4\} \xrightarrow{b} \{4\} \xrightarrow{a} \{2\} \xrightarrow{a} \{3\} \xrightarrow{c} \{4\} \dots$$

Autômato Temporizado [Alur, Dill, 90–94]



Alfabeto: $\Sigma = \{a, b, c\}$

Lugares: $Q = \{1, 2, 3, 4\}$

Lugares Iniciais: $Q_0 = \{1\}$

Lugares Finais: $F = \{4\}$

Relógios: $\{x, y\}$

Transições

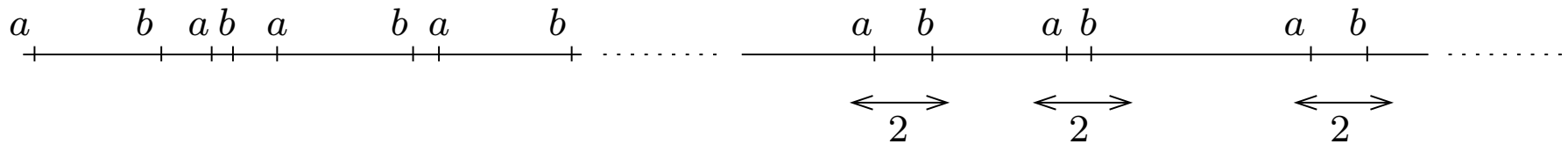
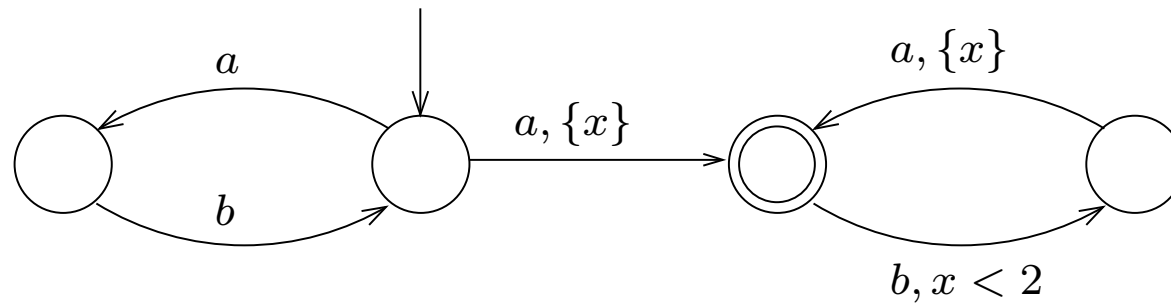
- Aceita palavras infinitas temporizadas: $\rho \in \Sigma^t$

$$\rho = (c, 0.6)(a, 2.8)(a, 2.9)(b, 7) \dots$$

$$\{1, \{0, 0\}\} \xrightarrow{c} \{1, \{0.6, 0\}\} \xrightarrow{a} \{2, \{3.4, 2.8\}\} \xrightarrow{a} \{3, \{0, 0\}\} \xrightarrow{b} \{4, \{7, 7\}\} \dots$$

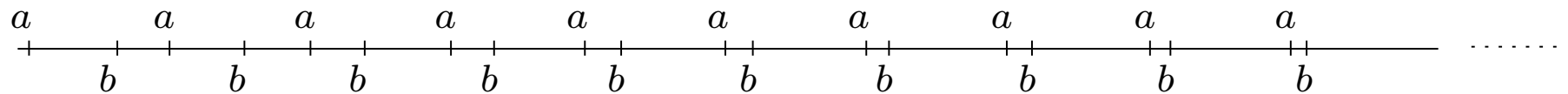
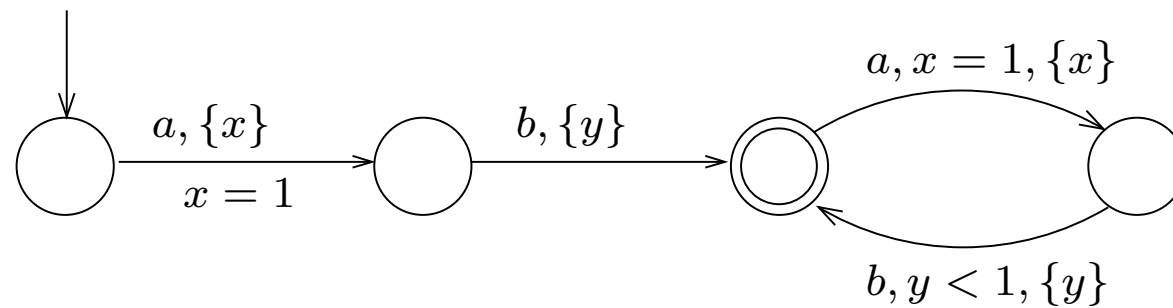
O que dá para fazer...

Tempo de resposta convergente

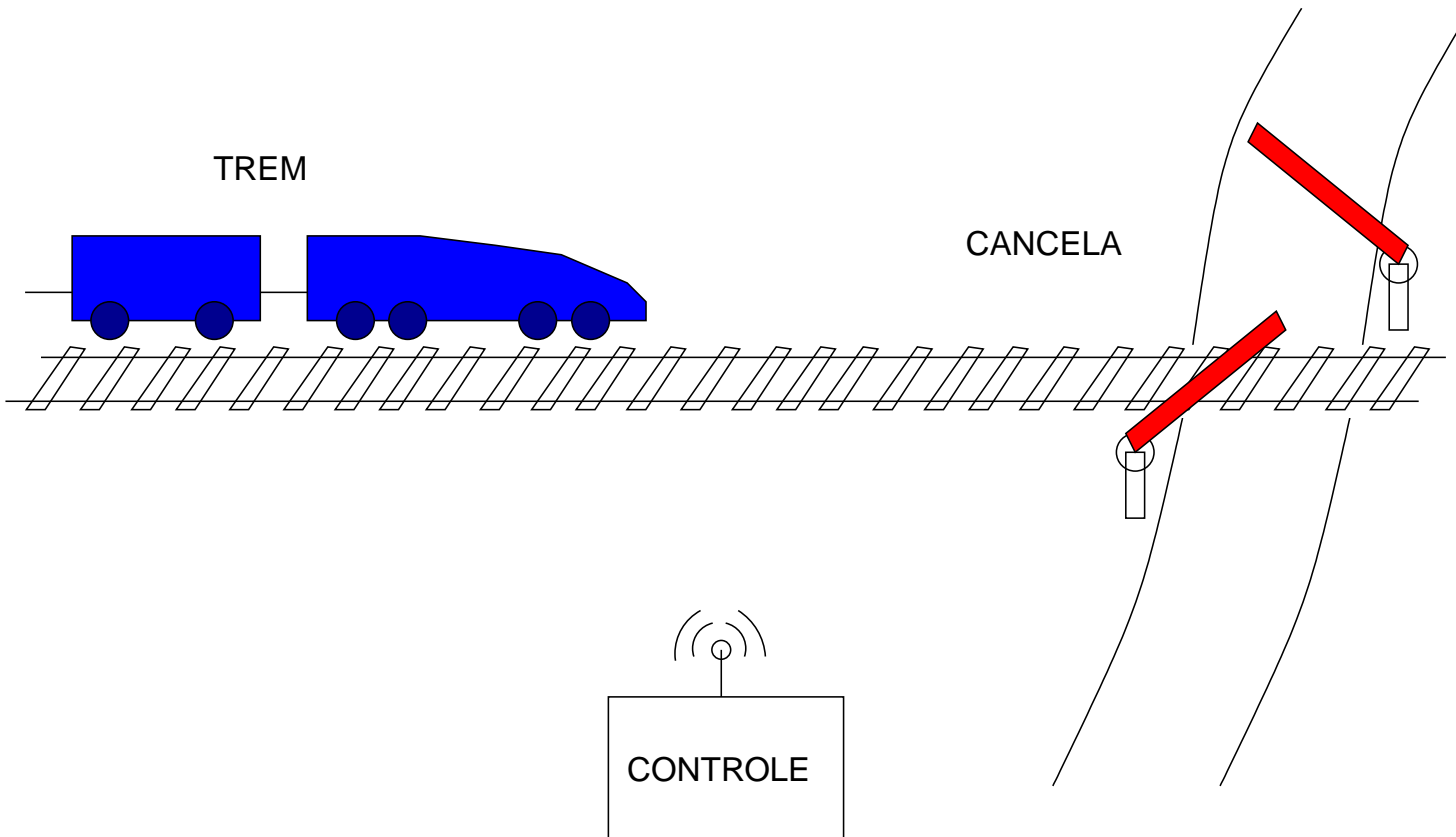


O que dá para fazer...

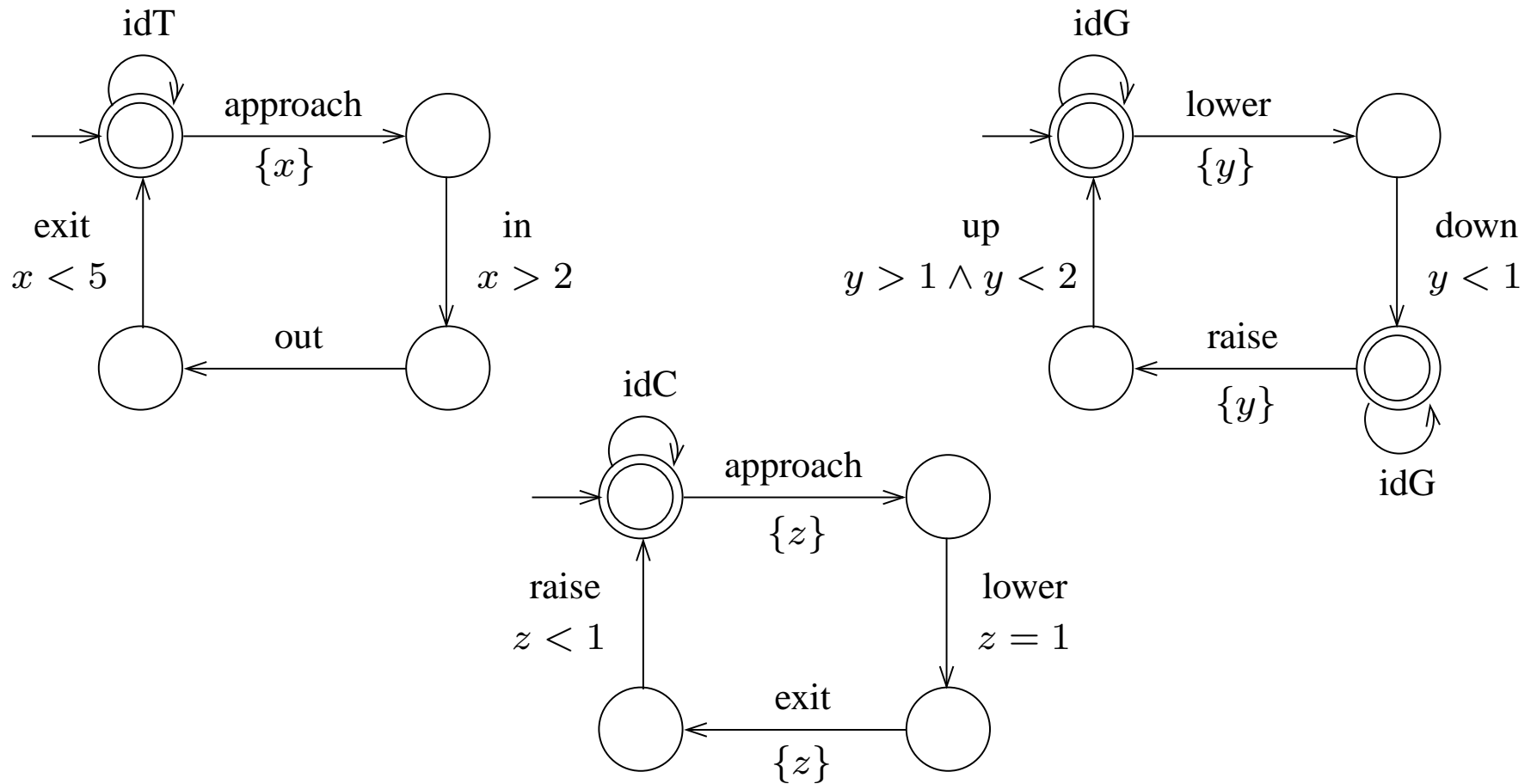
Distância estritamente decrescente



Verificação usando Linguagens

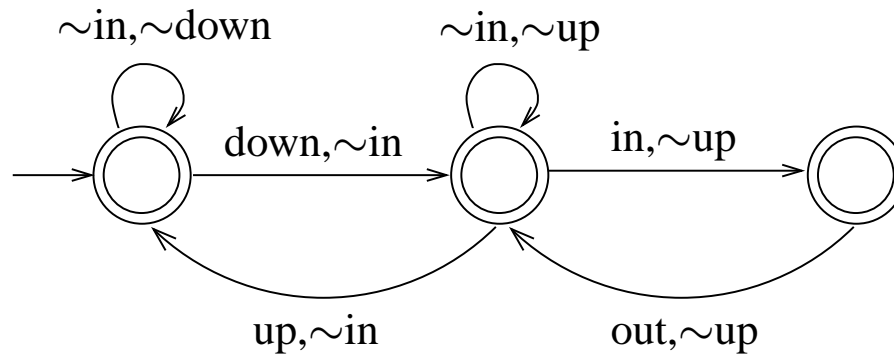


Composição de Autômatos

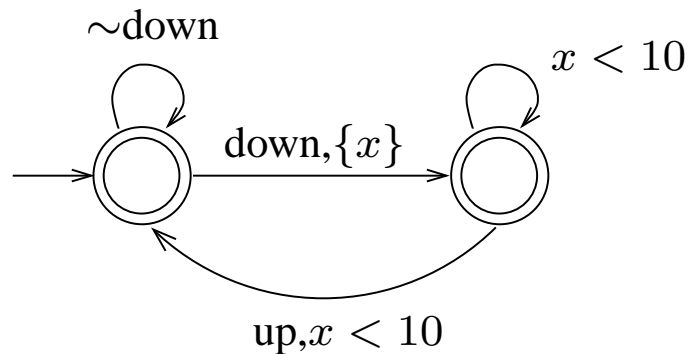


Sistema = Trem || Controle || Cancela

Propriedades



P_1 : *Safety*: Se o trem está cruzando a passagem, então a cancela está fechada



P_2 : *Real-Time Liveness*: A cancela sempre fica menos de 10 minutos fechada

○ Problema de Verificação é:

○ Sistema satisfaz a propriedade P_2 sse

$$\mathcal{L}(\text{Sistema}) \subseteq \mathcal{L}(P_2)$$

○ Problema de Verificação é:

○ Sistema satisfaz a propriedade P_2 sse

$$\mathcal{L}(\text{Sistema}) \subseteq \mathcal{L}(P_2)$$

Mas isso equivale a

$$\mathcal{L}(\text{Sistema}) \cap \overline{\mathcal{L}(P_2)} = \emptyset$$

Propriedades de Fechamento

Autômato	Interseção	União	Complemento
Aut. Finito	fechado	fechado	fechado
ω -autômato	fechado	fechado	fechado
Aut. Temporizado	fechado	fechado	aberto

Problemas de Decisão

Autômato	Universalidade (Validade)	Não-vacuidade (Satisfatibilidade)
Aut. Finito	PSPACE-completo	linear
ω -autômato	PSPACE-completo	linear
Aut. Temporizado	Π_2^1, Π_1^1 -difícil [Alur 94]	PSPACE-completo

Verificação

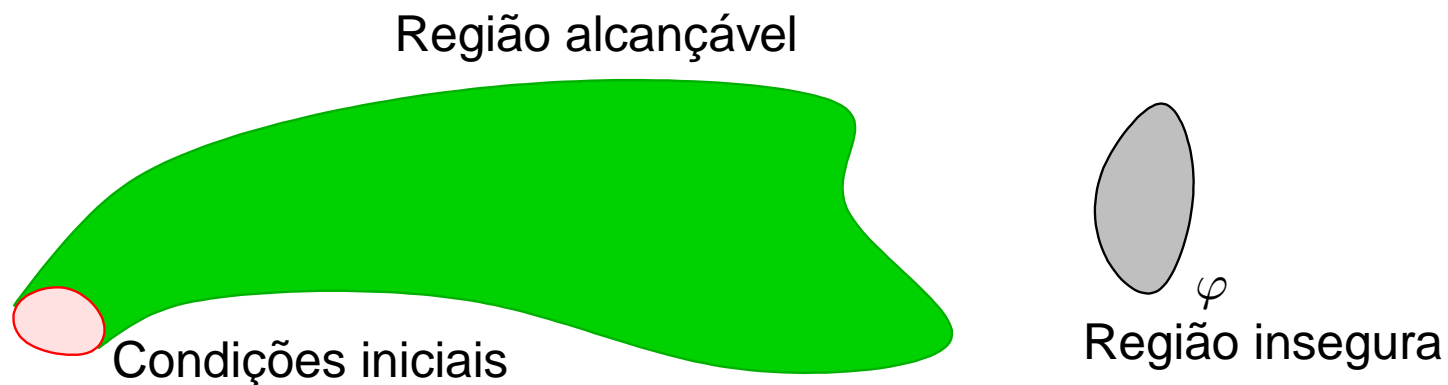
1. Autômato \models Autômato

2. Autômato $\models \phi$ (Model-Checking)

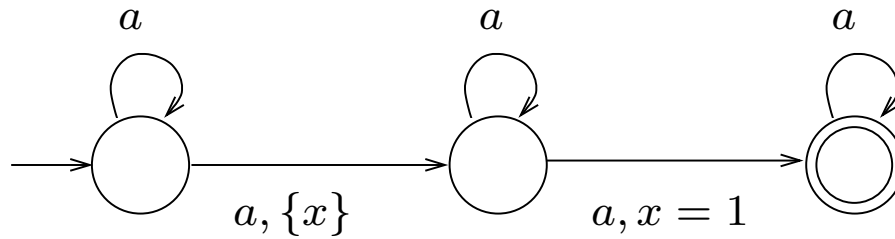
ϕ é fórmula de uma Lógica Temporal conveniente

3. Autômato $\models \text{AG}(\neg\varphi)$ (Safety/Reachability)

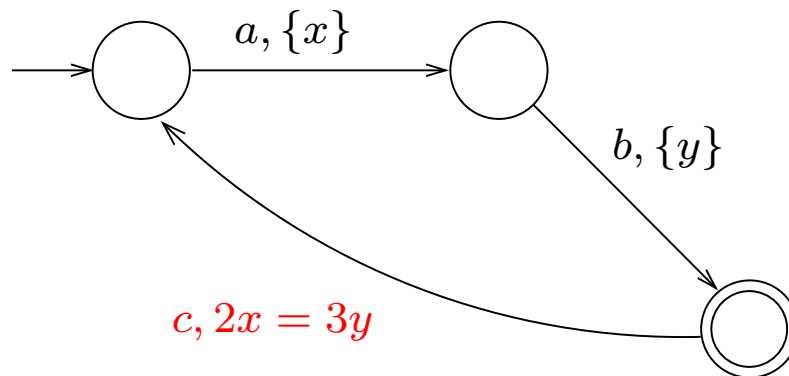
φ é fórmula sem operadores temporais



O que **não** dá para fazer...



← Complementar isso!



← Guardar intervalos!

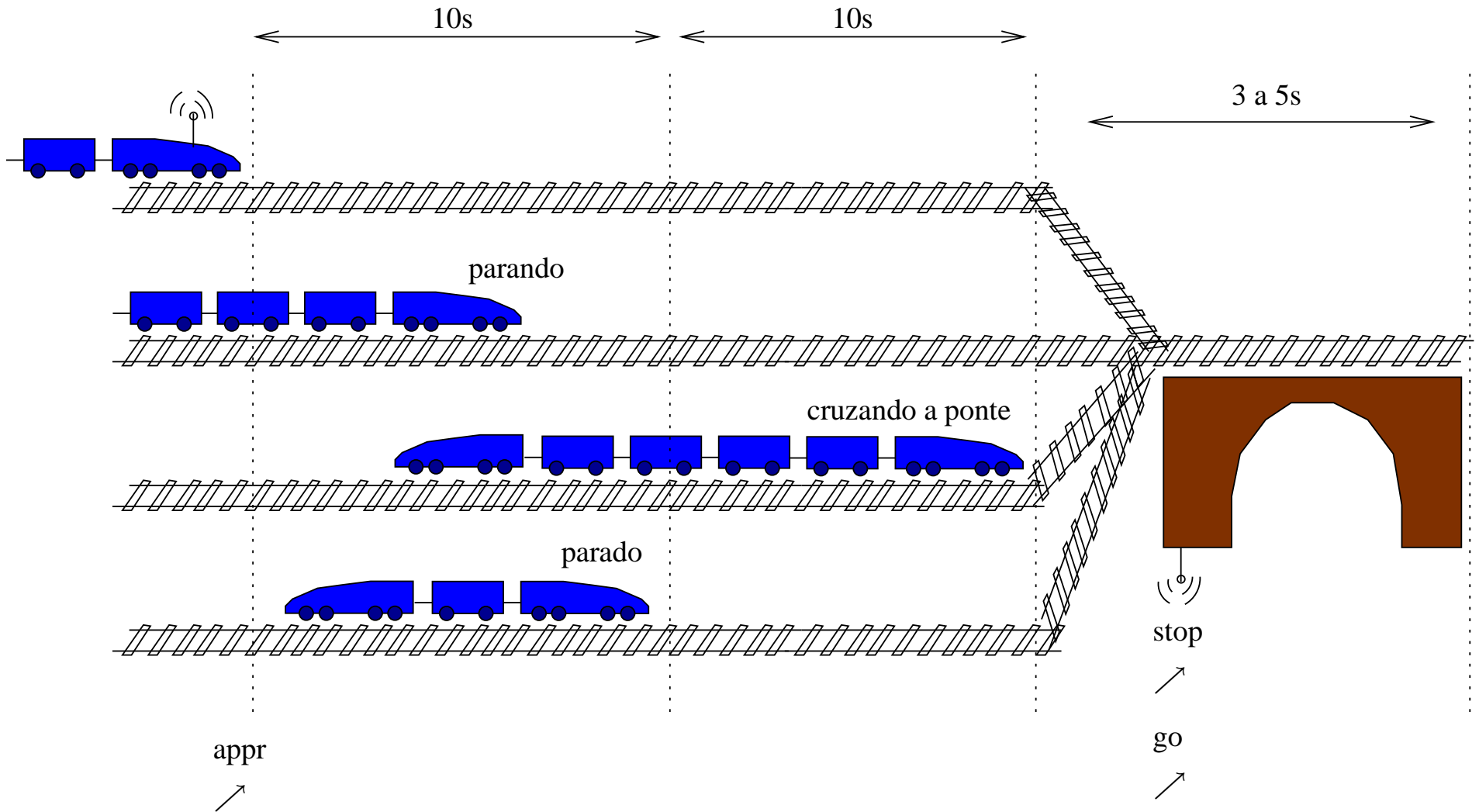
Lógicas Equivalentes

Autômato	Lógica Monádica	Lógica Temporal
Aut. Finito	LMSO	
ω -autômato	ω -LMSO	Q-LTL
Aut. Temporizado	$\mathcal{L} \overset{\leftarrow}{\overrightarrow{d}}$ [Wilke 94]	P-EventClockTL [Henzinger 98]

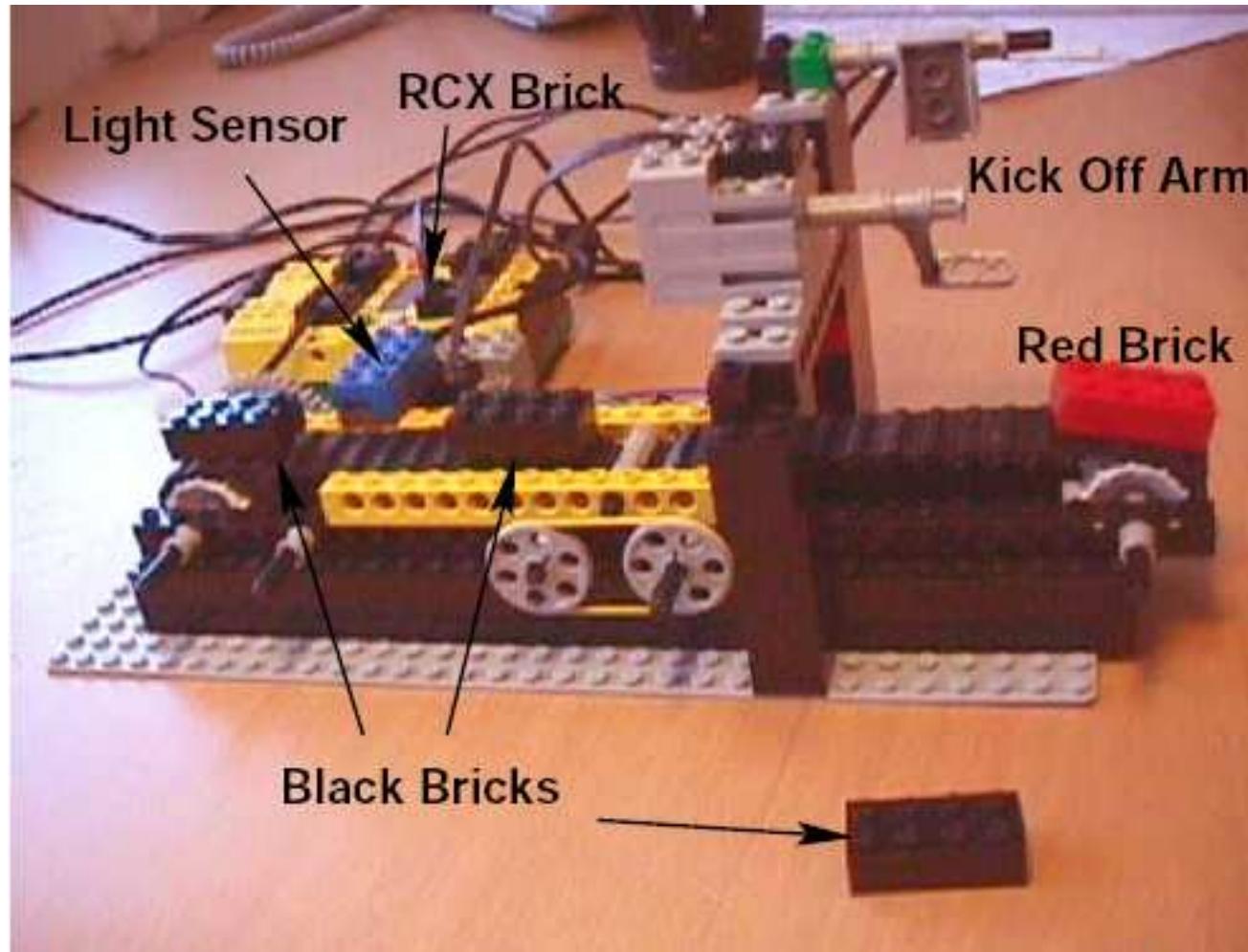
Ferramentas para Verificação Automática

1. UPPAAL ↗
2. TIMES ↗ (*Scheduling* usando Aut. Temporizados)
3. KRONOS, TReX, Rabbit, ... ↘
4. HyTECH ↘ (Aut. Híbridos e alcançabilidade)
5. SPIN ↗ (ω -autômatos e LTL)

Exemplo em UPPAAL: Trem e Ponte

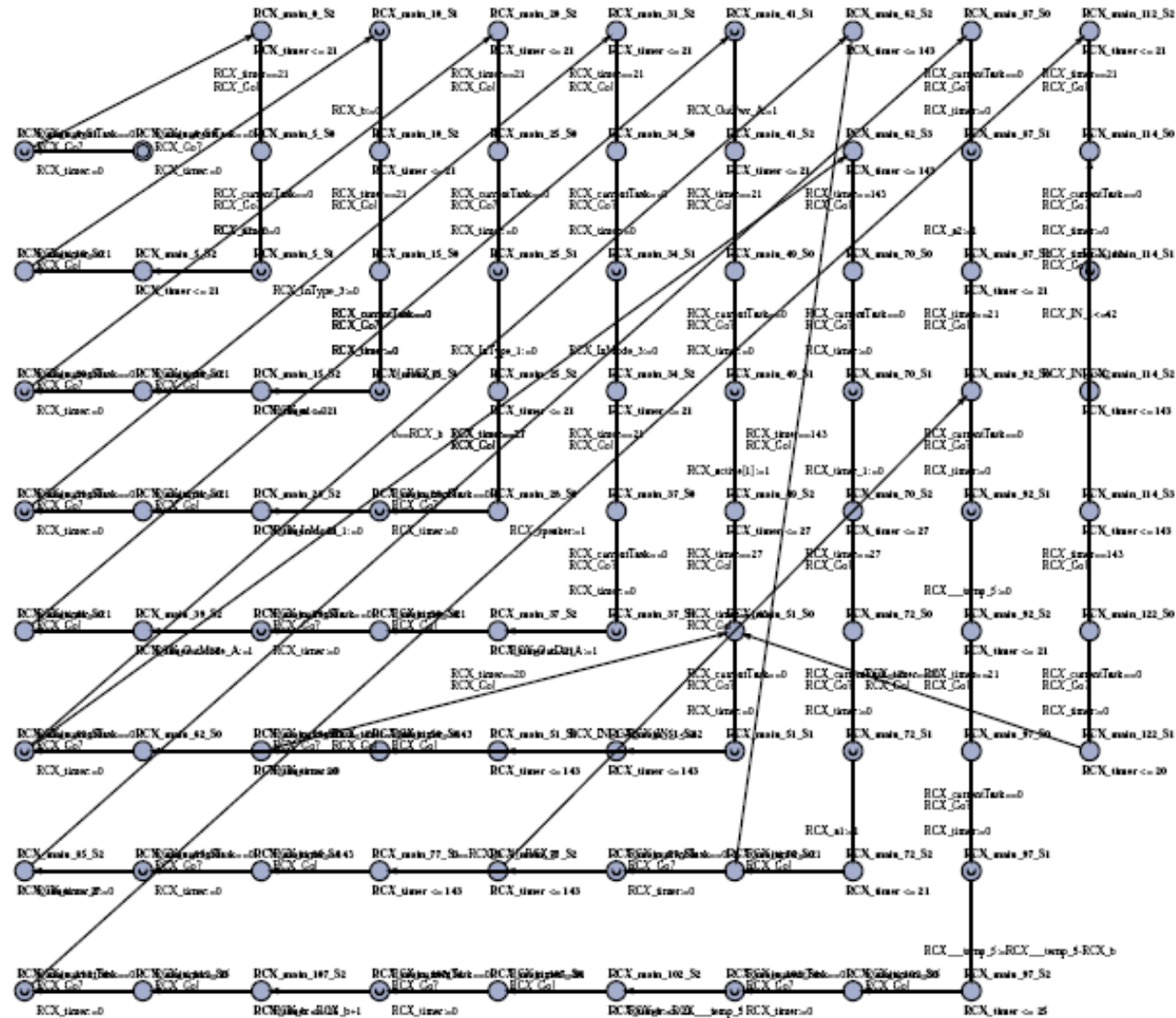


Exemplo em UPPAAL: Linguagem de Controle para LEGO

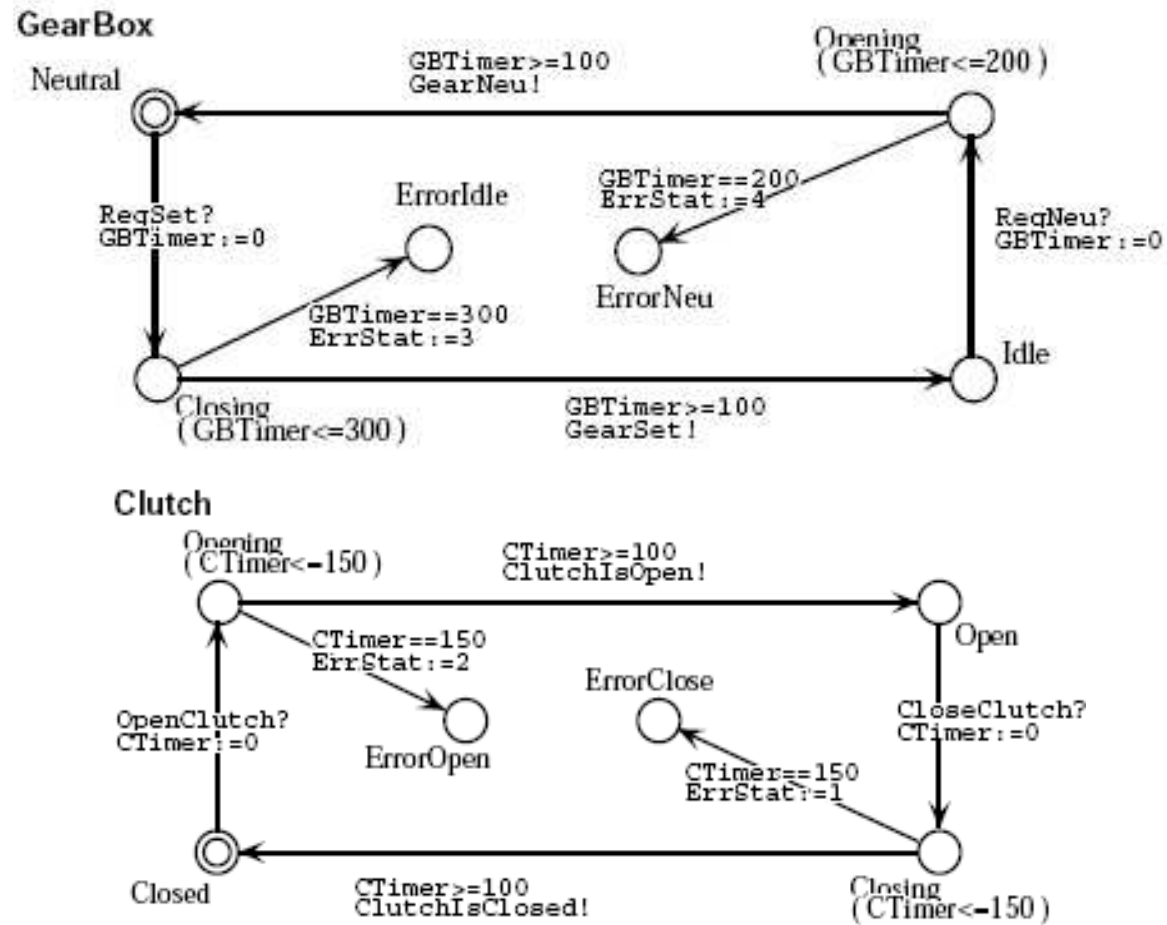


[Iversen, Larsen, et.al., 2000]

Exemplo em UPPAAL: Linguagem de Controle para LEGO



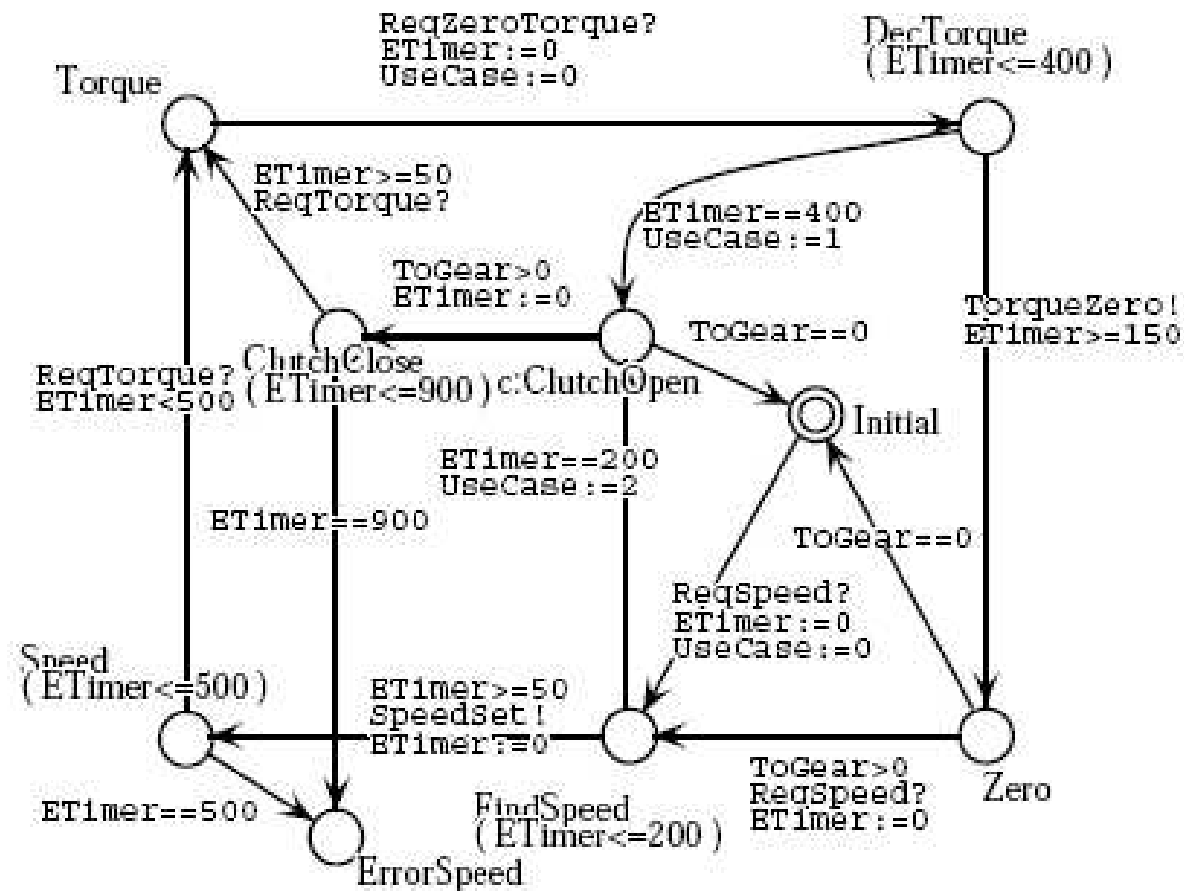
Exemplo em UPPAAL: Controle de *Câmbio* em automóveis



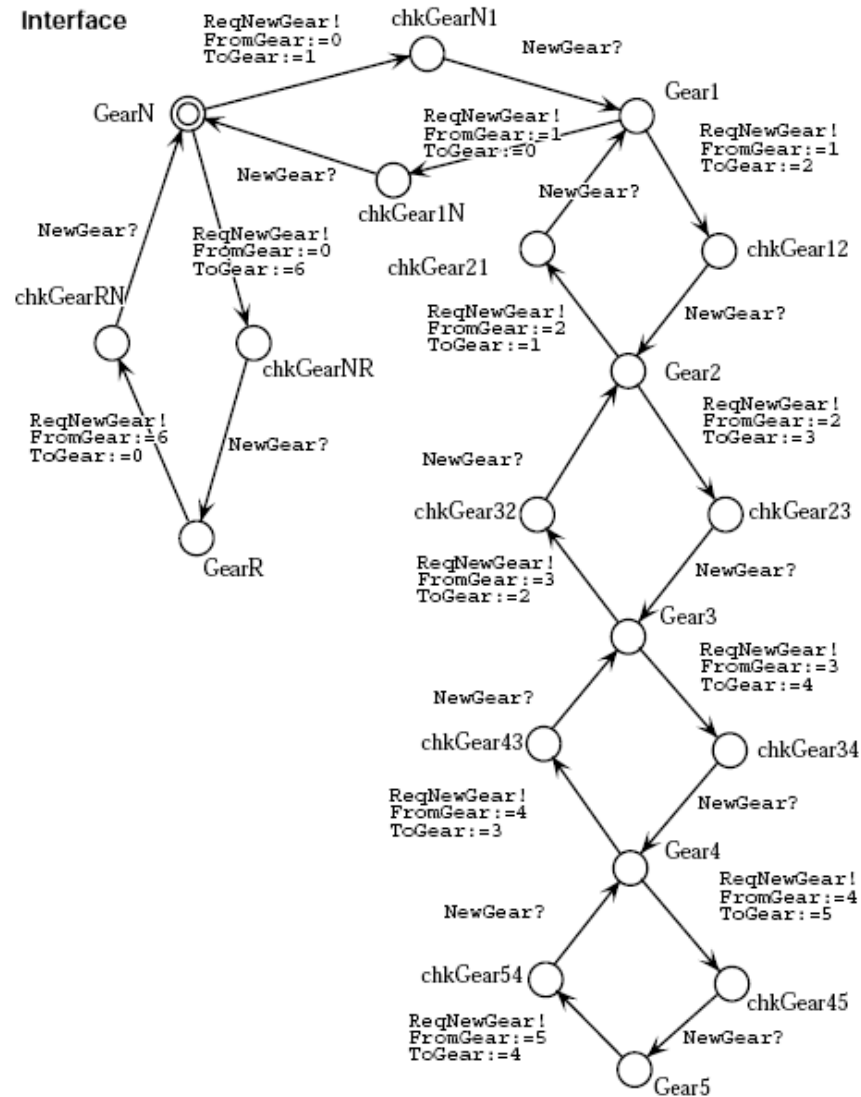
[Lindahl, et.al., 2001]

Exemplo em UPPAAL: Controle de *Câmbio* em automóveis

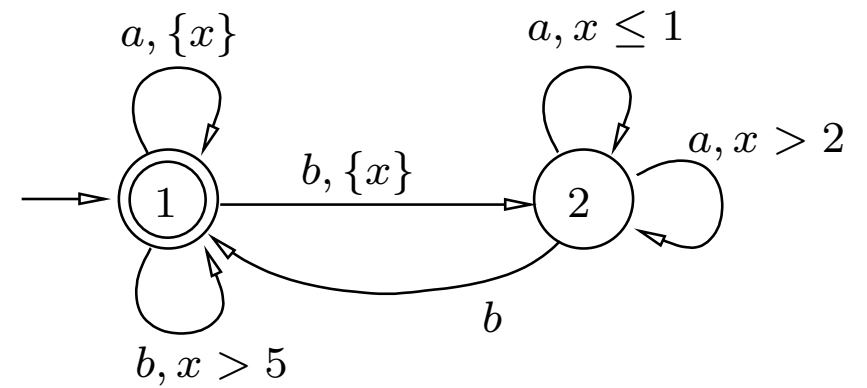
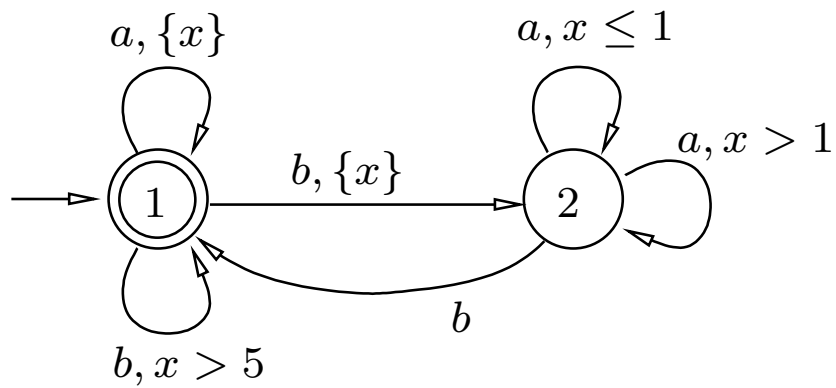
Engine



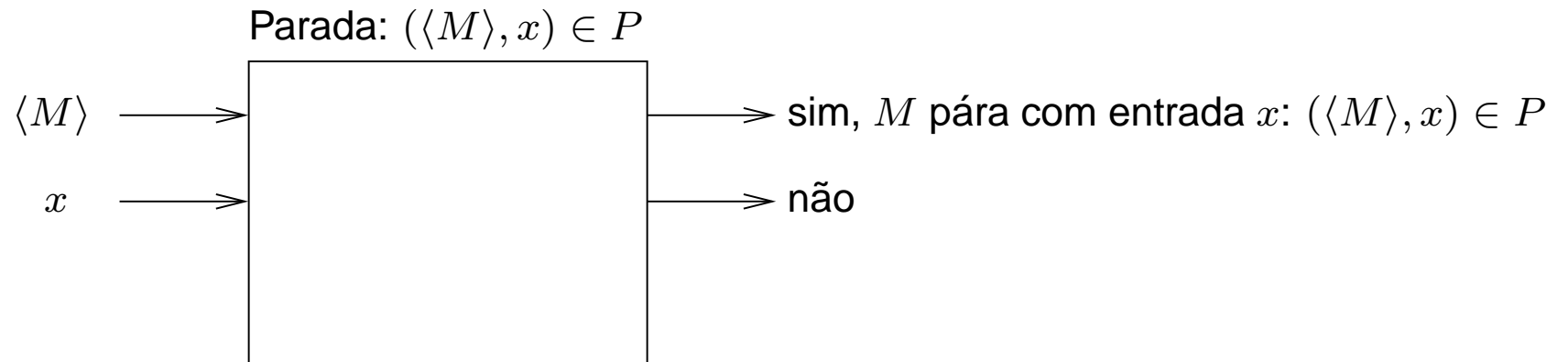
Exemplo em UPPAAL: Controle de *Câmbio* em automóveis

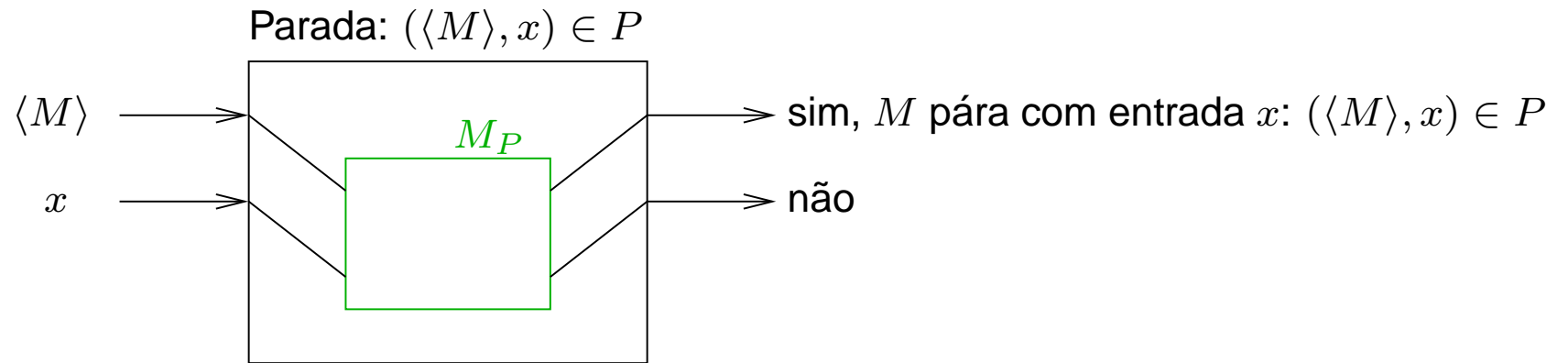


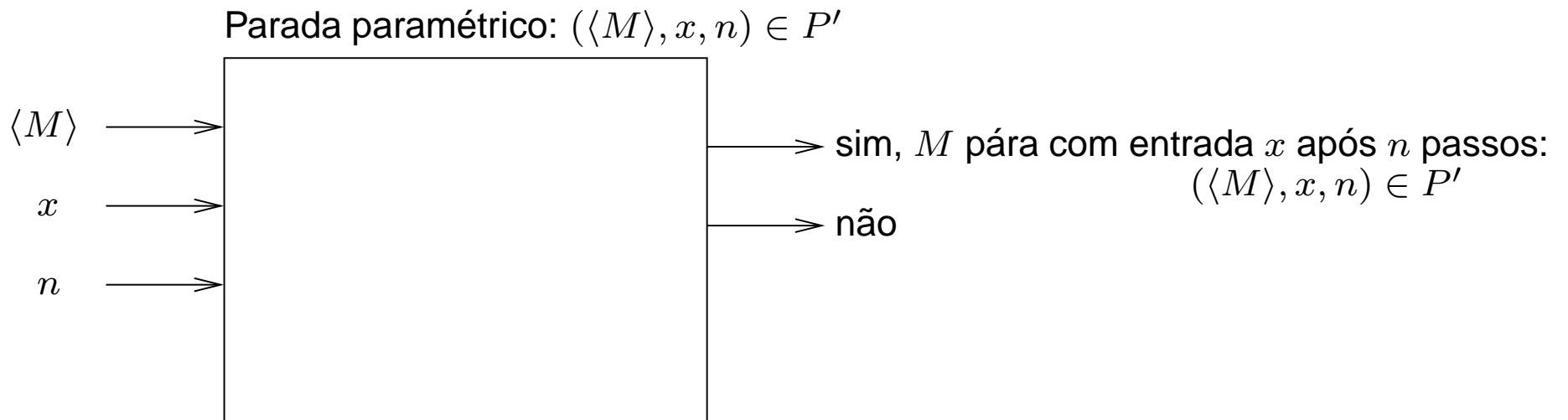
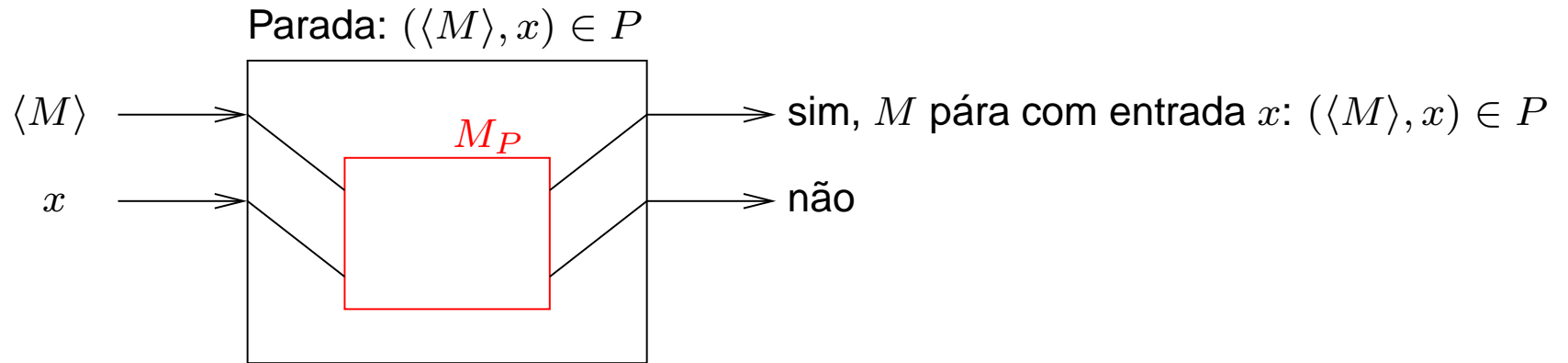
Grau de Indecidibilidade da Universalidade

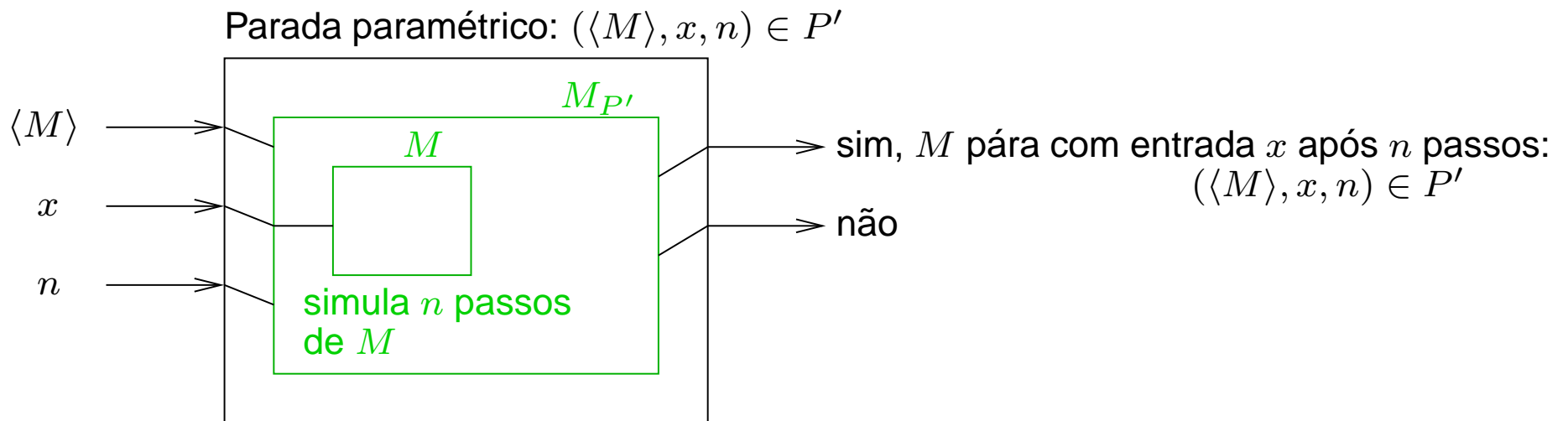
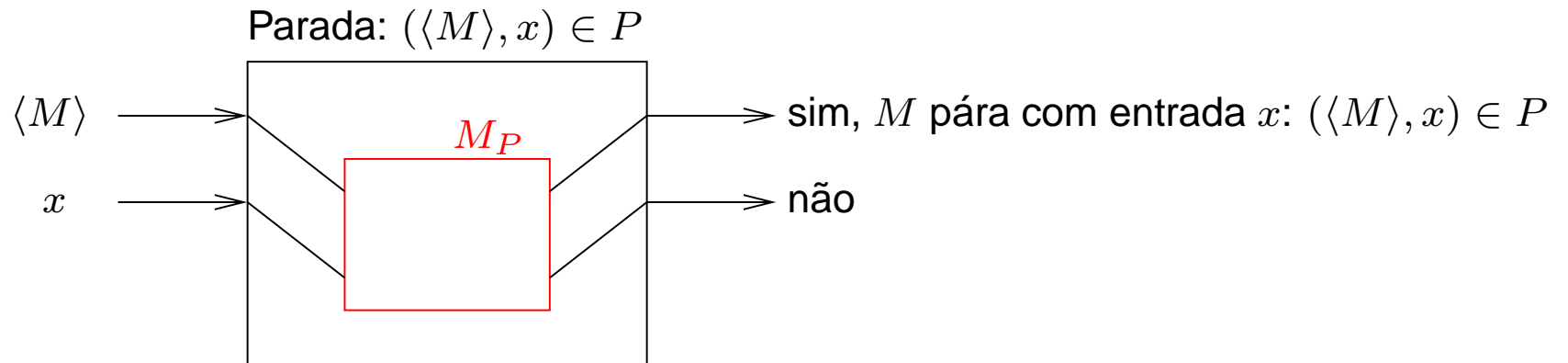


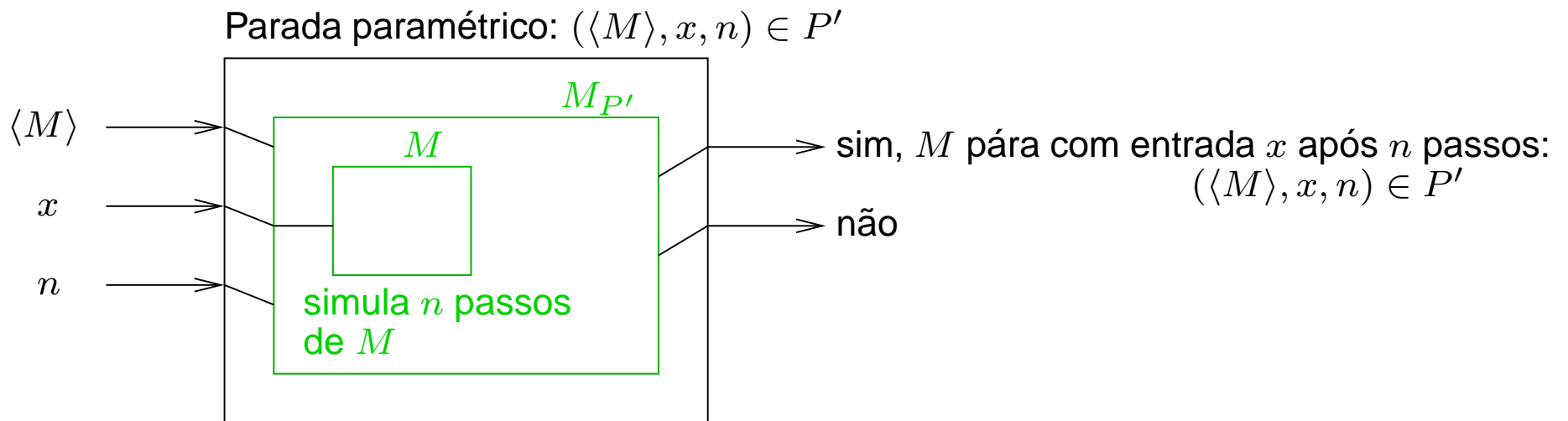
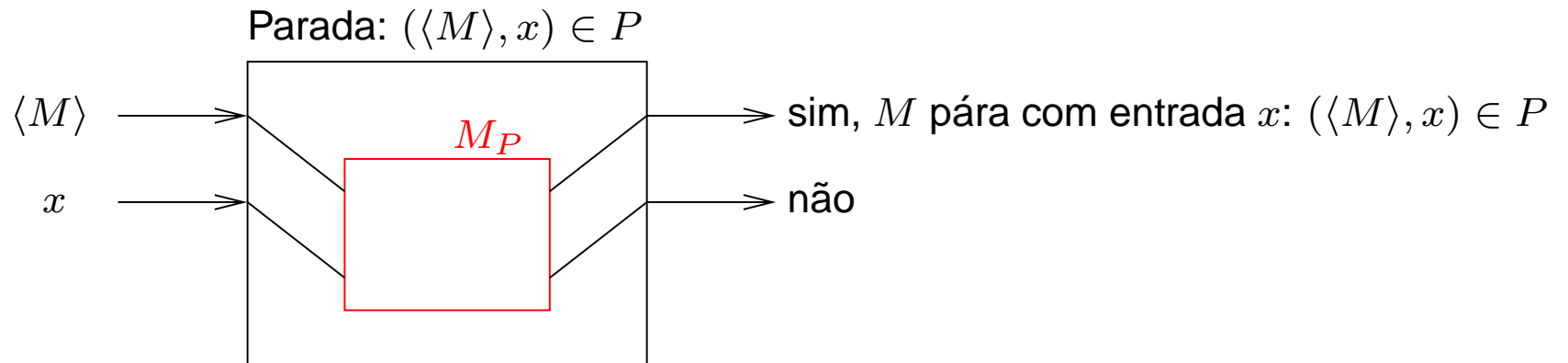
- U_{ABT} é indecidível, Π_1^1 -difícil [Alur 94]











$(\langle M \rangle, x) \in P$ se e somente se $\exists n [(\langle M \rangle, x, n) \in P']$

Hierarquia Aritmética

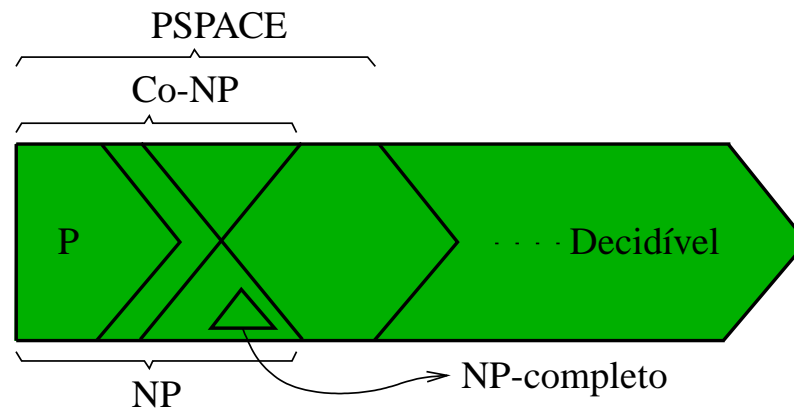
$$\left\{ \begin{array}{l} \Sigma_1^0 \longleftarrow \exists n \\ \Pi_1^0 \longleftarrow \forall n \\ \Sigma_2^0 \longleftarrow \exists n \forall m \\ \Pi_2^0 \longleftarrow \forall n \exists m \\ \vdots \end{array} \right.$$

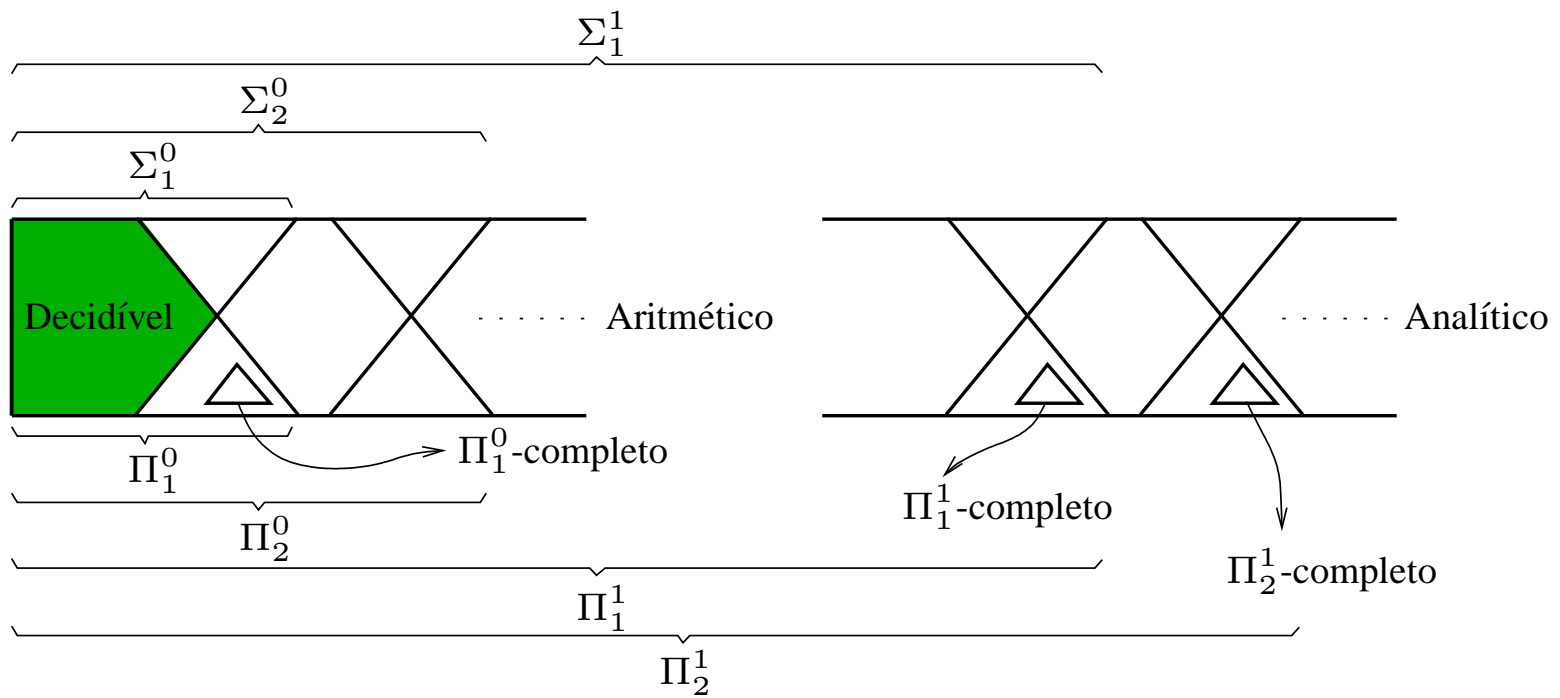
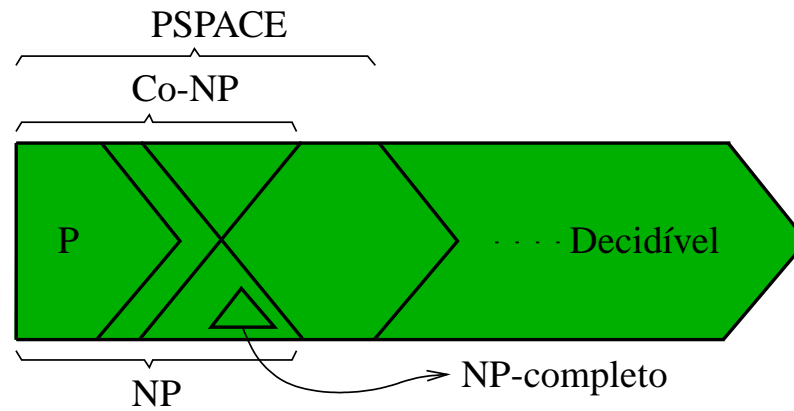
Hierarquia Aritmética

$$\left\{ \begin{array}{l} \Sigma_1^0 \longleftarrow \exists n \\ \Pi_1^0 \longleftarrow \forall n \\ \Sigma_2^0 \longleftarrow \exists n \forall m \\ \Pi_2^0 \longleftarrow \forall n \exists m \\ \vdots \end{array} \right.$$

Hierarquia Analítica

$$\left\{ \begin{array}{l} \Sigma_1^1 \longleftarrow \exists f \forall n \\ \Pi_1^1 \longleftarrow \forall f \exists n \\ \Sigma_2^1 \longleftarrow \exists f \forall g \exists n \\ \Pi_2^1 \longleftarrow \forall f \exists g \forall n \\ \vdots \end{array} \right.$$





A Situação atual do Problema

- U_{ABT} pertence a Π_2^1

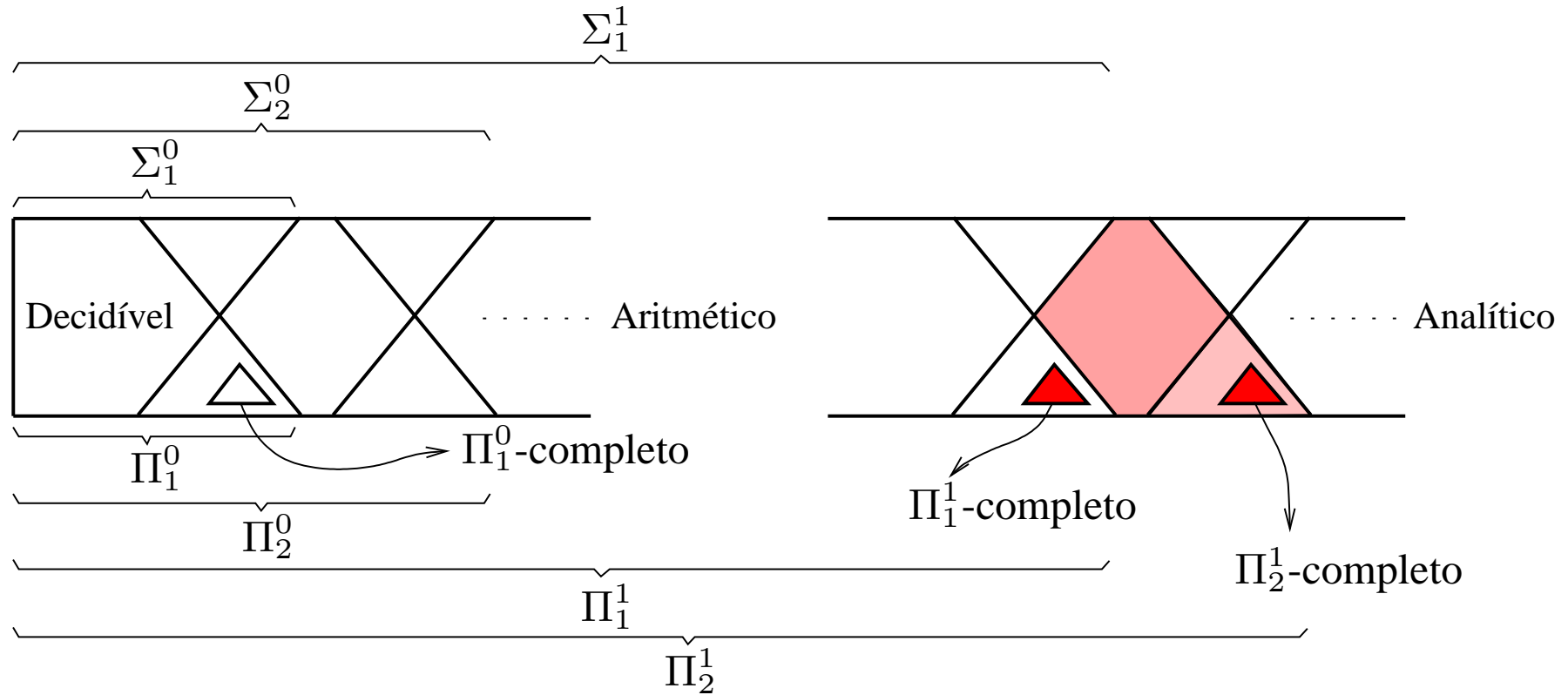
Para toda palavra existe trajetória de aceitação



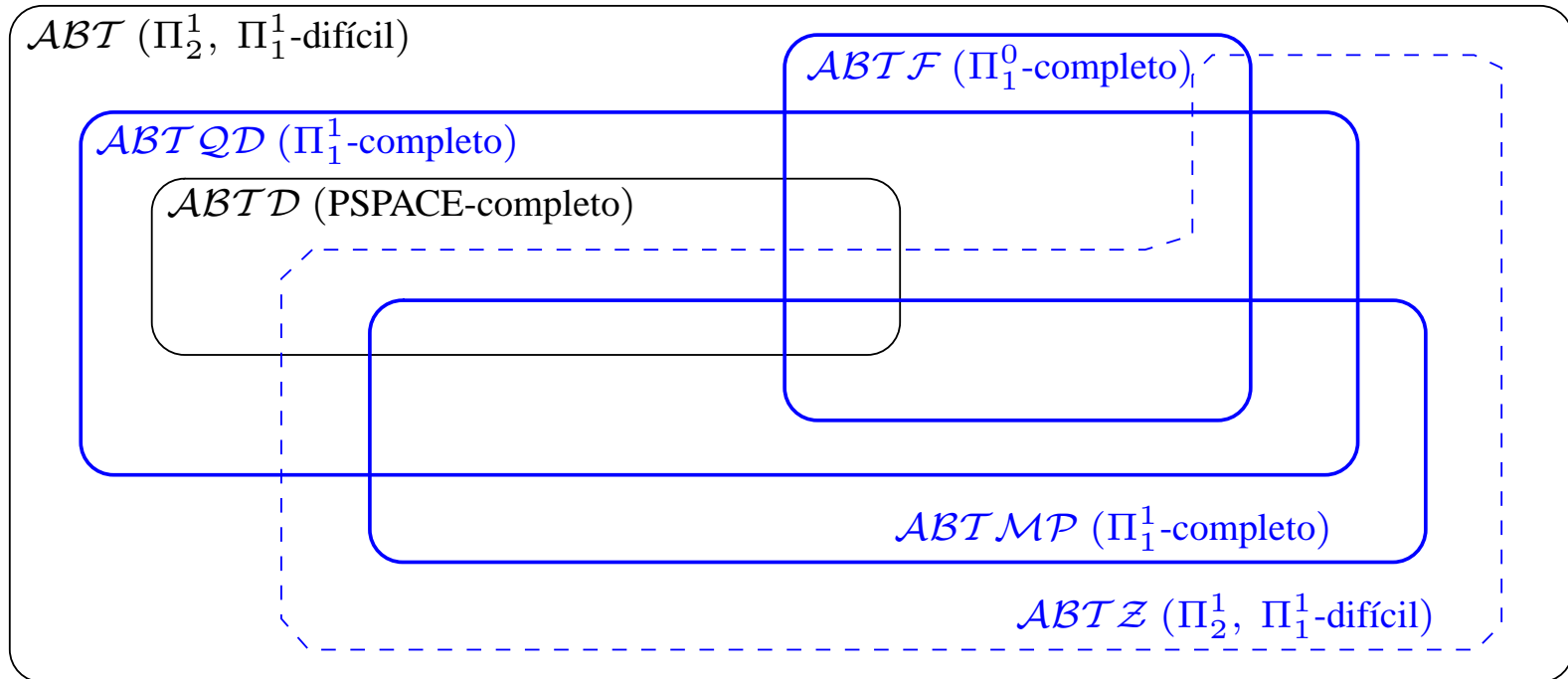
$$U_{\text{ABT}} = \{z \mid \forall f \exists g [(\forall i \exists j H_1(f, i, j)) \Rightarrow (\forall i \exists j H_2(f, g, i, j, z))] \}$$

- U_{ABT} é Π_1^1 -difícil [Alur 94]

A Situação atual do Problema



Resumo dos Resultados



Intuição

Complexidade da Universalidade

