

JSARP – Simulador e Analisador de Redes de Petri

Aluno: Felipe G. de Oliveira Lino

Orientador: Alexandre Sztajnberg

DICC / IME / UERJ

Rio de Janeiro, RJ, Brasil

felipelino44@gmail.com, alexszt@ime.uerj.br

Sumário

- ◆ Motivação
- ◆ Redes de Petri
- ◆ Trabalhos relacionados
- ◆ JSARP
- ◆ Execução do programa
- ◆ Trabalhos futuros
- ◆ Conclusão

Motivação

- ◆ Potencial da modelagem com RdP em sistemas computacionais:
 - Escalonamento de processos
 - Protocolos de comunicação
 - Sistemas concorrentes
- ◆ Verificação de propriedades como: *deadlock e livelock*. Refletem características do sistema modelado
- ◆ Funcionalidades desejáveis:
 - “Token Animation Game”
 - Geração da Árvore de Alcançabilidade
 - Editor Gráfico
 - Verificação de Propriedades

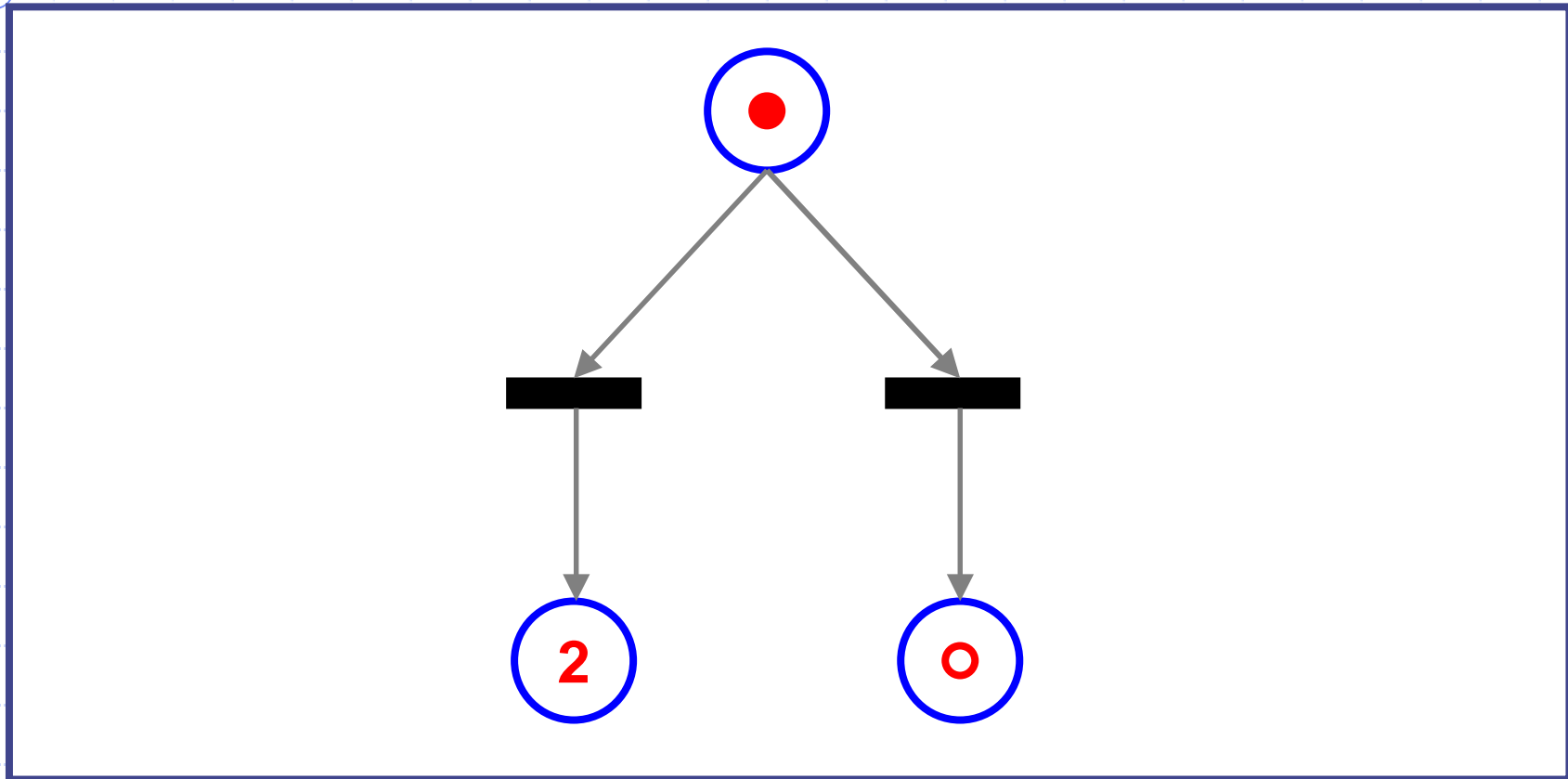
Objetivo

- ◆ Desenvolvimento e implementação de uma ferramenta que reúne funcionalidades desejáveis em ferramentas de RdP
- ◆ Independente de plataforma
- ◆ Resultados gráficos
- ◆ Interação com usuário
- ◆ Extensível

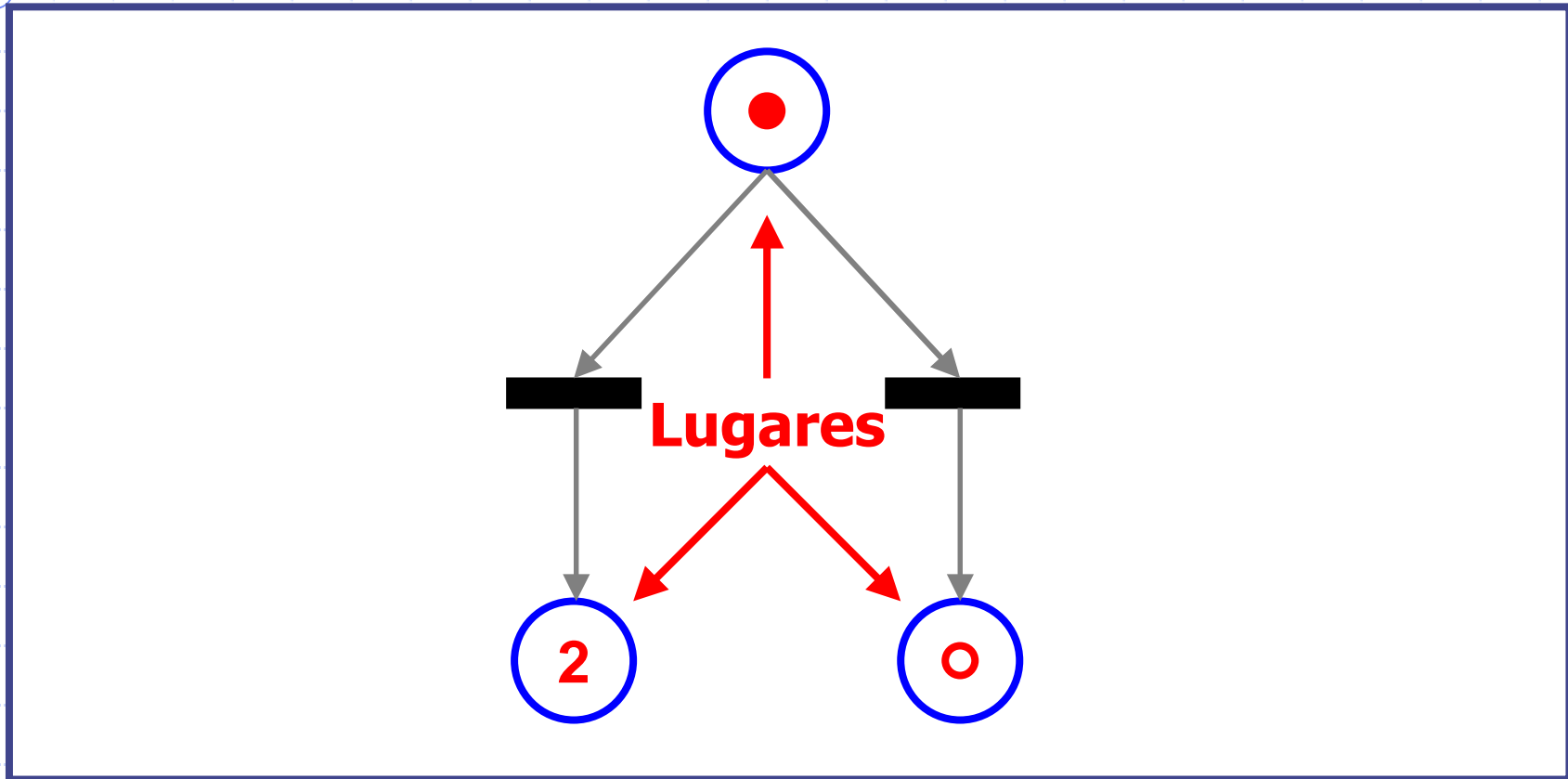
Redes de Petri

- ◆ Grafo composto por 3 elementos:
 - Lugares
 - ◆ Representam os estados da rede
 - Transições
 - ◆ Representam as ações da rede
 - Arcos
 - ◆ Ligam os lugares às transições e vice-versa

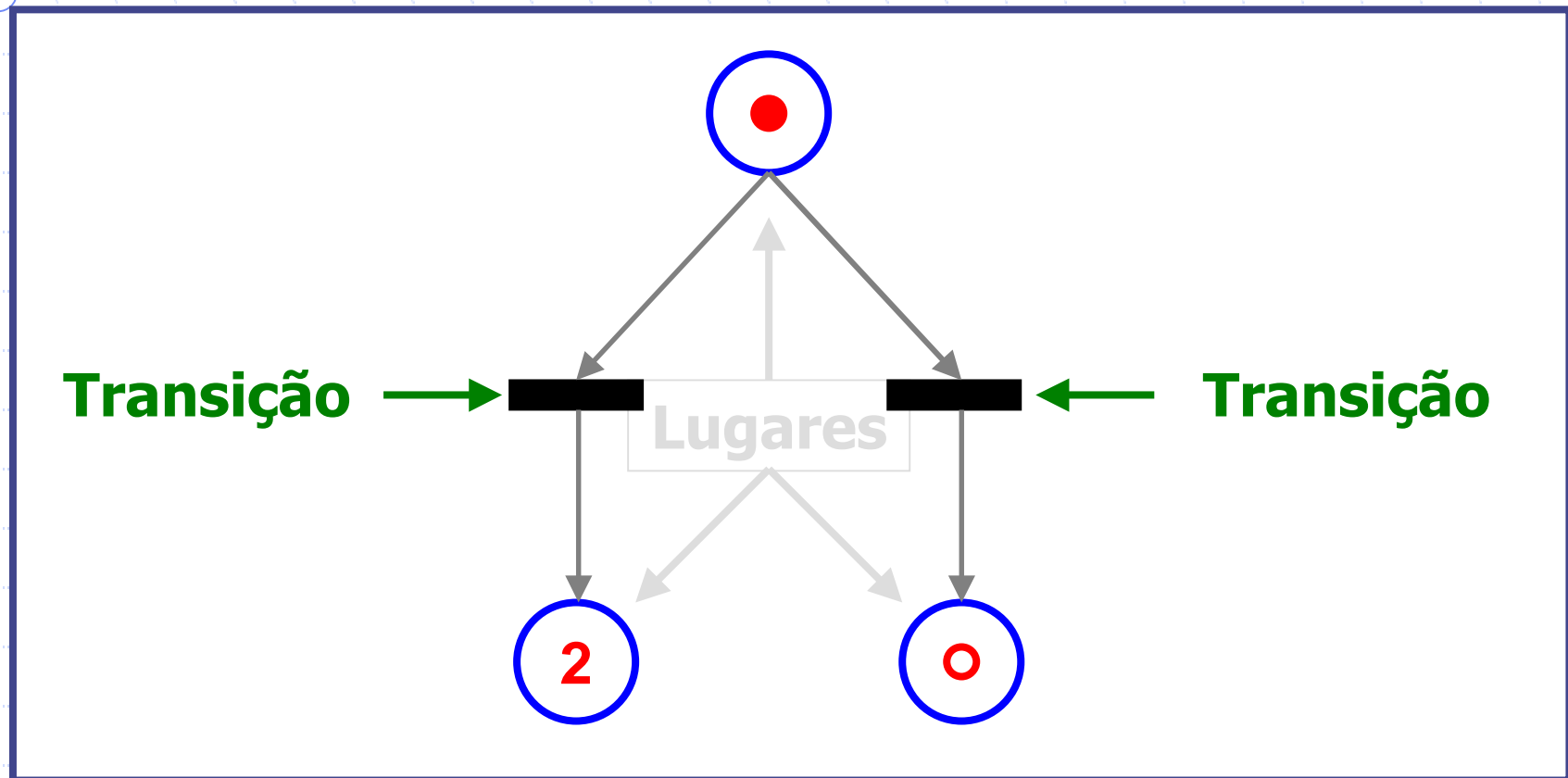
Redes de Petri



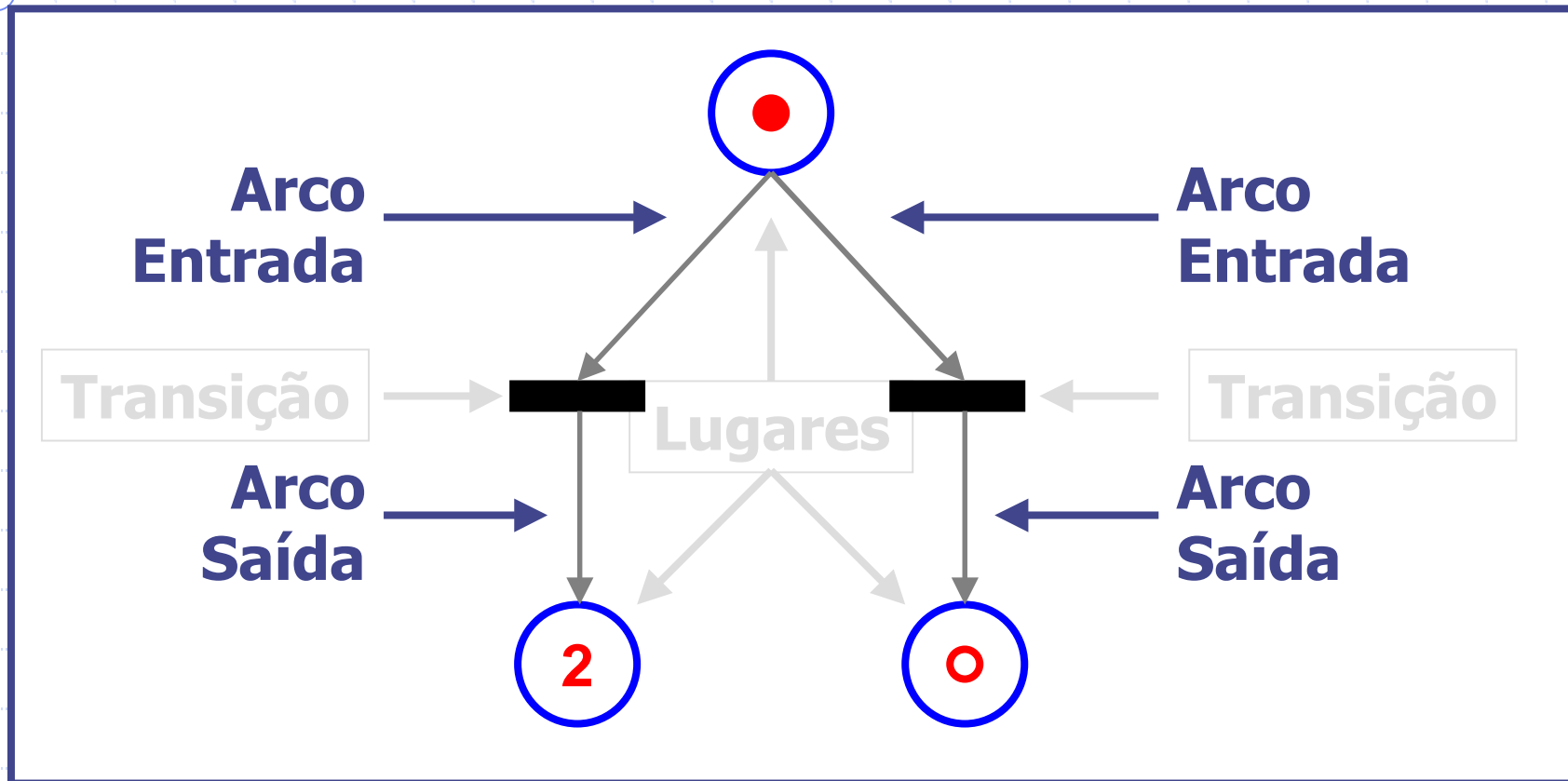
Redes de Petri



Redes de Petri



Redes de Petri



RdP – Regras de Execução

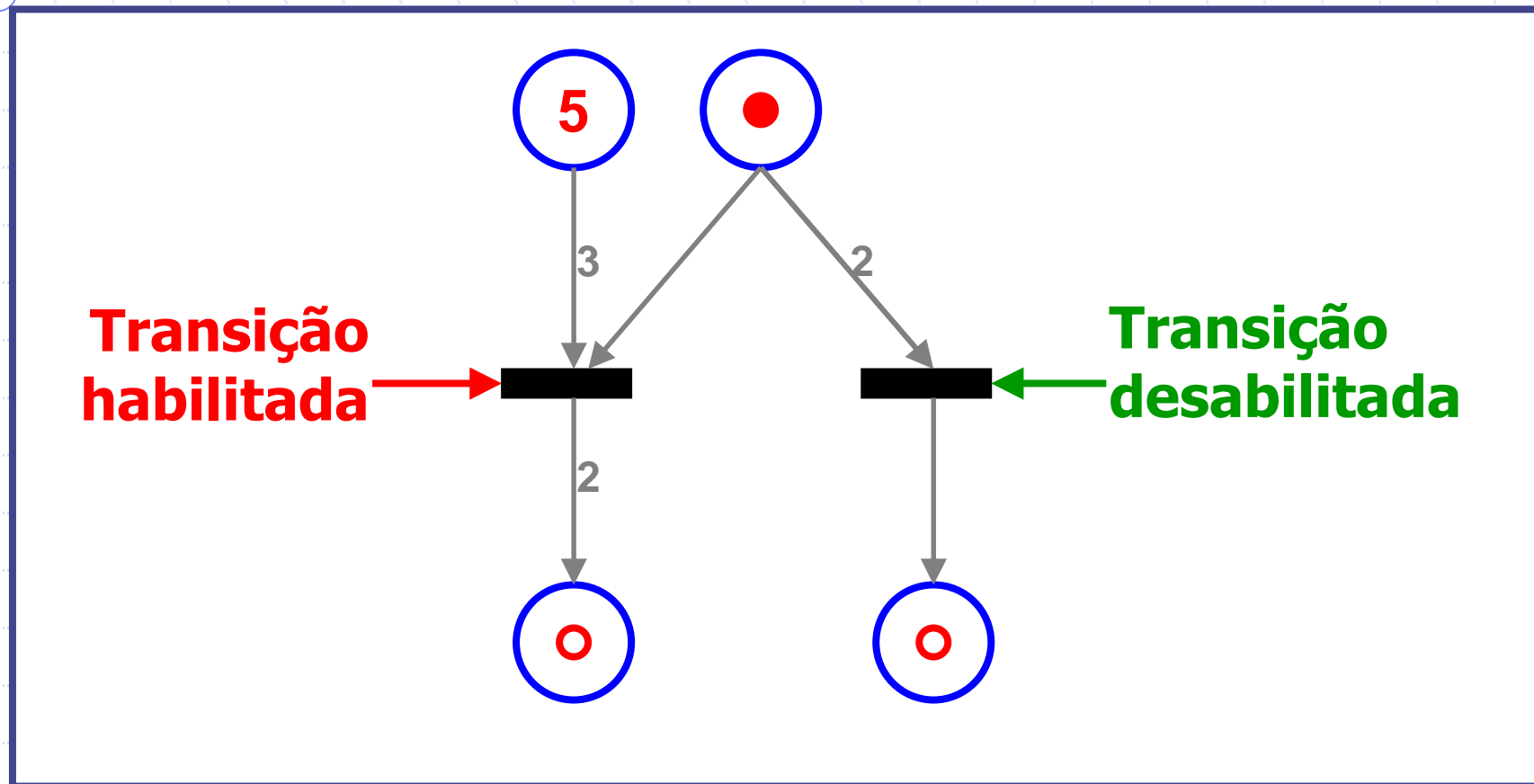
◆ Transição habilitada

- Uma transição pode disparar quando todos os seus lugares de entrada possuírem fichas em quantidade superior ou igual ao peso do arco que o liga até a transição.

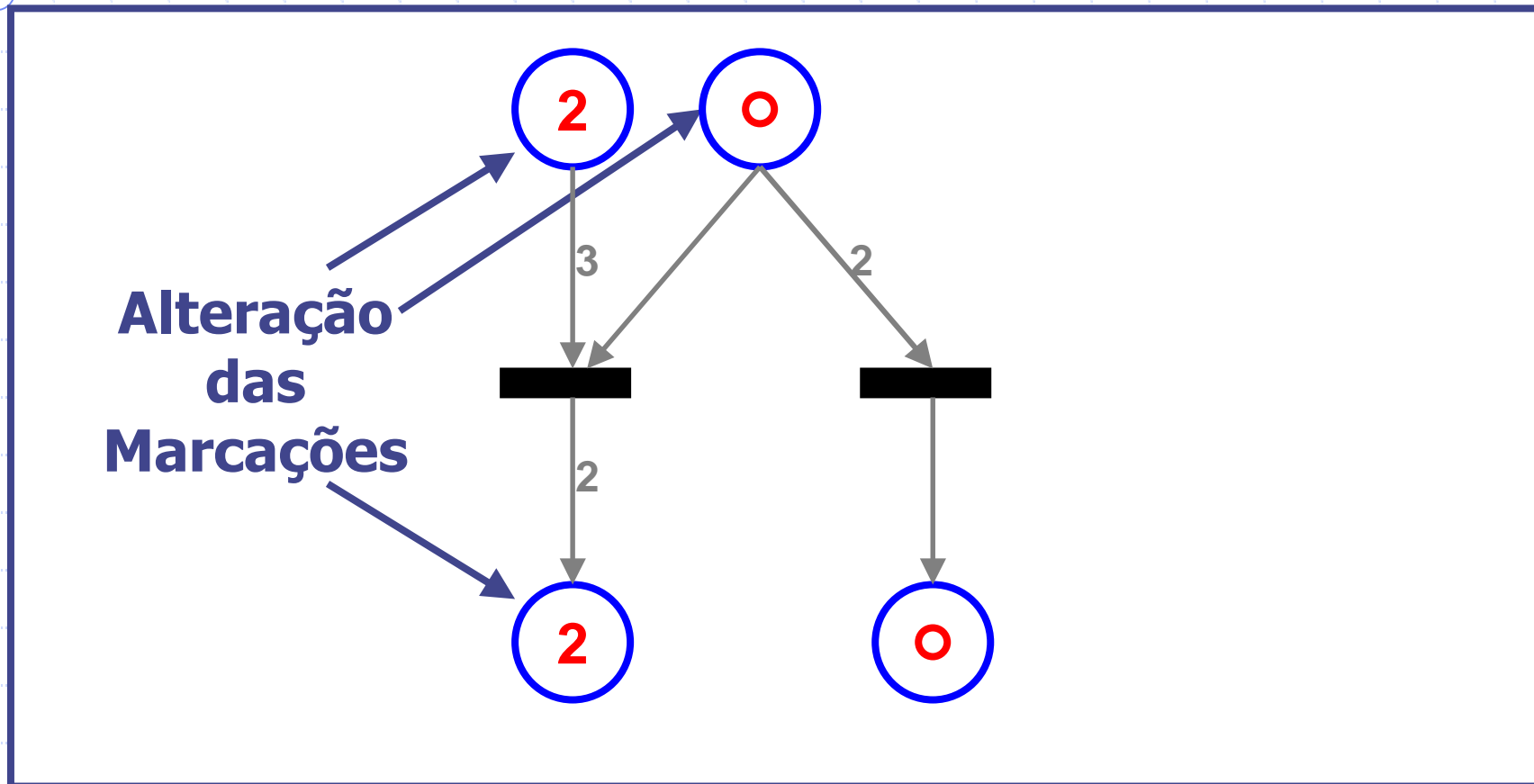
◆ Regra

- Para todo lugar de entrada da transição disparada é removida a quantidade de fichas equivalente ao peso do arco de entrada, enquanto para todo lugar de saída é adicionada a quantidade de fichas equivalente ao peso do arco de saída.

RdP – Regras de Execução



RdP – Regras de Execução



RdP – Propriedades

- ◆ Rede limitada
- ◆ Rede segura
- ◆ Rede reinicializável ou própria
- ◆ Rede conservativa
- ◆ Rede viva
- ◆ Bloqueio ou *deadlock*

RdP – Representação Matricial

Rede de Petri: $RP = (P, T, I, O, \mu_0)$

Lugares: $P = (P_1, P_2, \dots, P_n)$

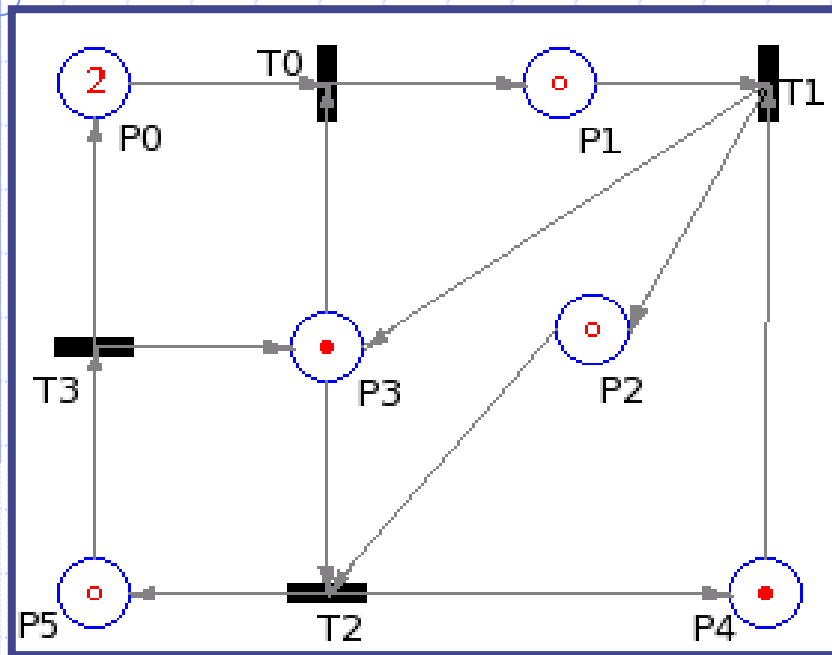
Transições: $T = (T_1, T_2, \dots, T_m)$

Matriz de Entrada (*Input*): $I_{m \times n}$

Matriz de Saída (*Output*): $O_{m \times n}$

Marcação Inicial: $\mu_0 = (\mu_0^1, \mu_0^2, \dots, \mu_0^n)$

Exemplo: Atelier Simples



- P0 – Número de peças
- P1 – Robô está carregando
- P2 – Peça na usina
- P3 – Robô livre para operar
- P4 – Torno está livre
- P5 – Robô descarregando
- T0 – Robô pega peça nova
- T1 – Robô carrega o torno
- T2 – Robô descarrega o torno
- T3 – Robô solta a peça pronta

Atelier Simples: Árvore

[2,0,0,1,1,0]

Atelier Simples: Árvore

Fronteira [2,0,0,1,1,0]

Atelier Simplex: Árvore

Fronteira [2,0,0,1,1,0] Transições Habilitadas: T0

Atelier Simples: Árvore

Fronteira [2,0,0,1,1,0]



Atelier Simplex: Árvore

Interior $[2,0,0,1,1,0]$



$[1,1,0,0,1,0]$

Atelier Simplex: Árvore

Interior $[2,0,0,1,1,0]$

↓ T0

Fronteira $[1,1,0,0,1,0]$

Atelier Simplex: Árvore

Interior [2,0,0,1,1,0]

↓ T0

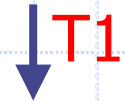
Fronteira [1,1,0,0,1,0] Transições Habilitadas: T1

Atelier Simplex: Árvore

Interior $[2,0,0,1,1,0]$



Fronteira $[1,1,0,0,1,0]$

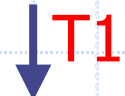


Atelier Simplex: Árvore

Interior [2,0,0,1,1,0]



Interior [1,1,0,0,1,0]



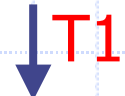
[1,0,1,1,0,0]

Atelier Simples: Árvore

Interior [2,0,0,1,1,0]



Interior [1,1,0,0,1,0]



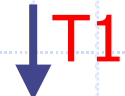
Fronteira [1,0,1,1,0,0]

Atelier Simples: Árvore

Interior [2,0,0,1,1,0]



Interior [1,1,0,0,1,0]



Fronteira [1,0,1,1,0,0] Transições Habilitadas: T0, T2

Atelier Simples: Árvore

Interior [2,0,0,1,1,0]



Interior [1,1,0,0,1,0]



Fronteira [1,0,1,1,0,0]



Atelier Simplex: Árvore

Interior $[2, 0, 0, 1, 1, 0]$



Interior $[1, 1, 0, 0, 1, 0]$



Fronteira $[1, 0, 1, 1, 0, 0]$



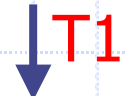
$[0, 1, 1, 0, 0, 0]$

Atelier Simples: Árvore

Interior [2,0,0,1,1,0]



Interior [1,1,0,0,1,0]

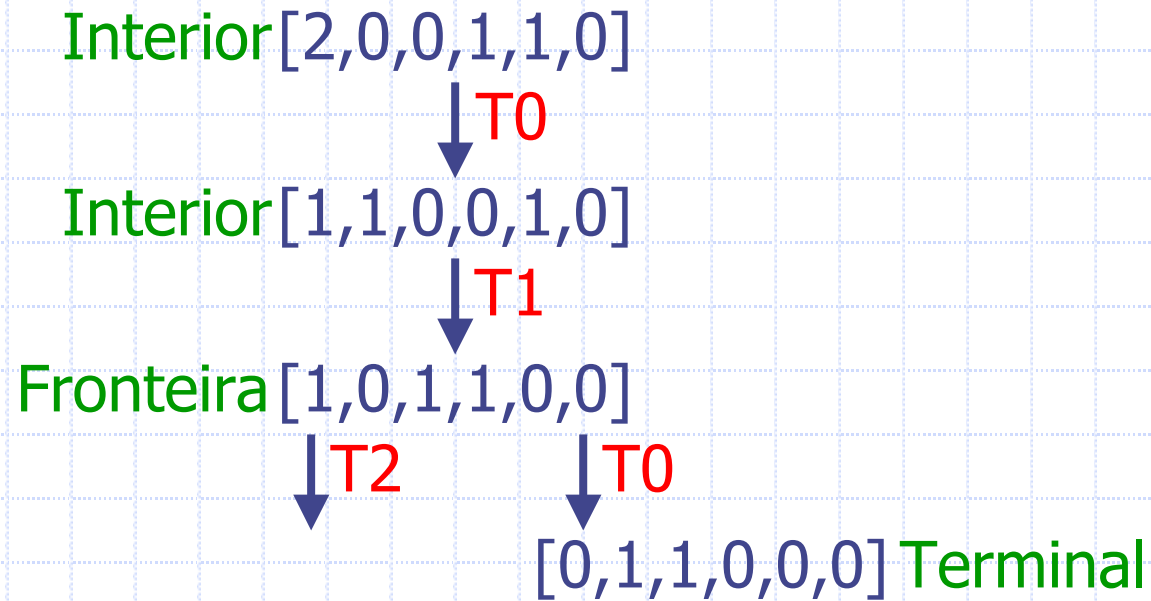


Fronteira [1,0,1,1,0,0]

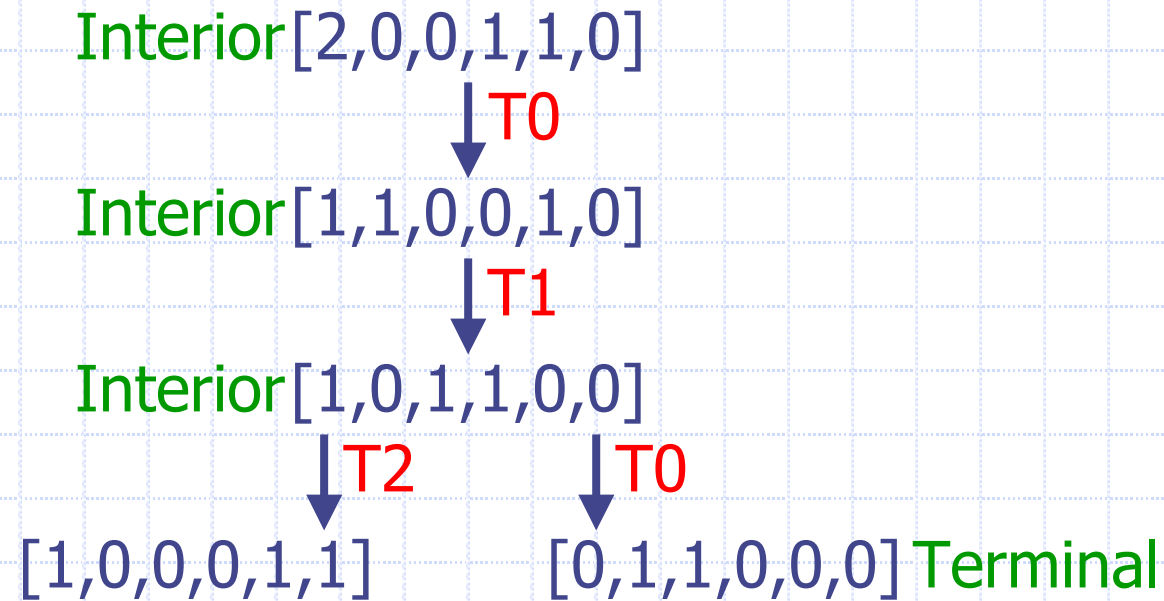


[0,1,1,0,0,0] Terminal

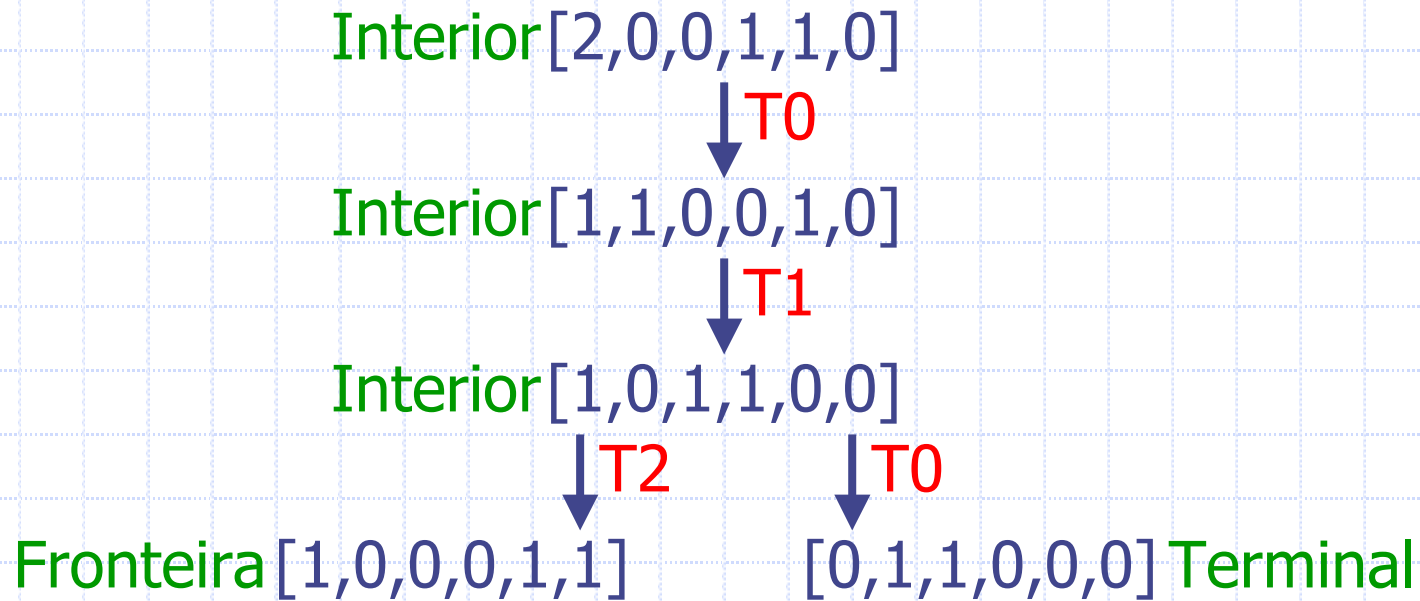
Atelier Simples: Árvore



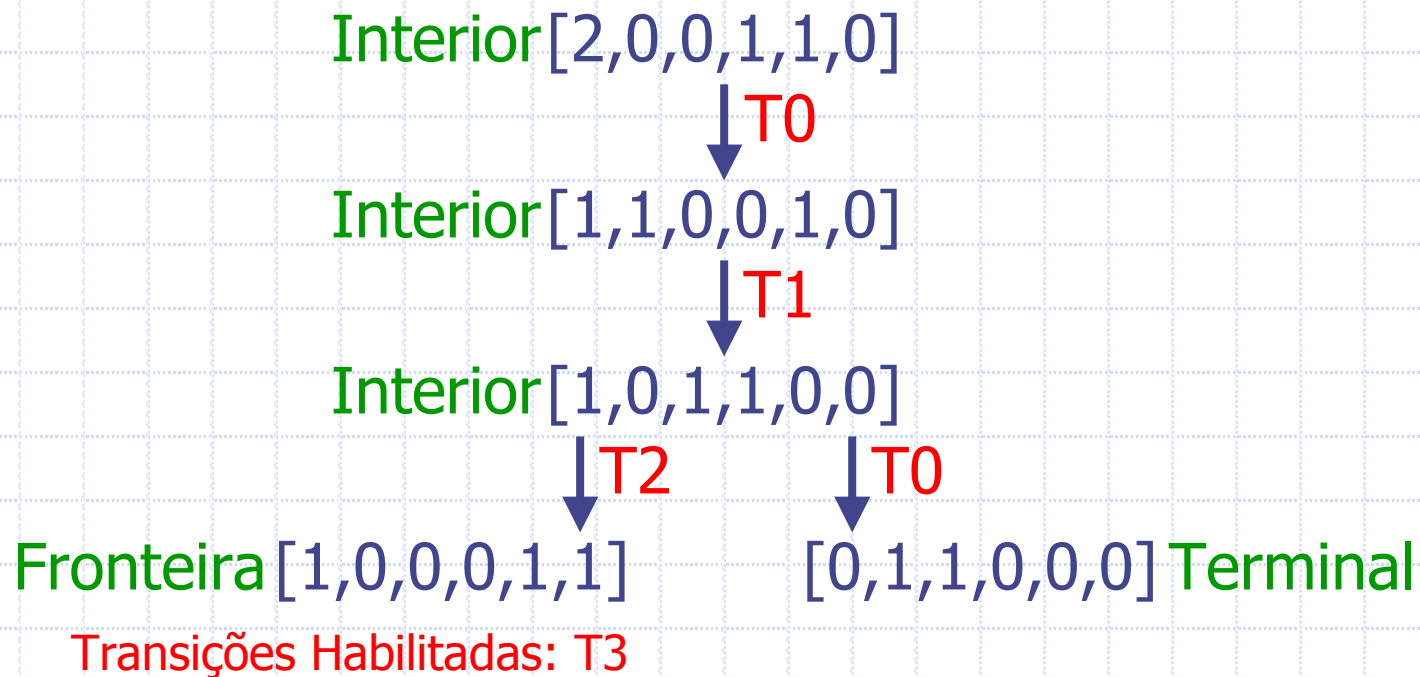
Atelier Simplex: Árvore



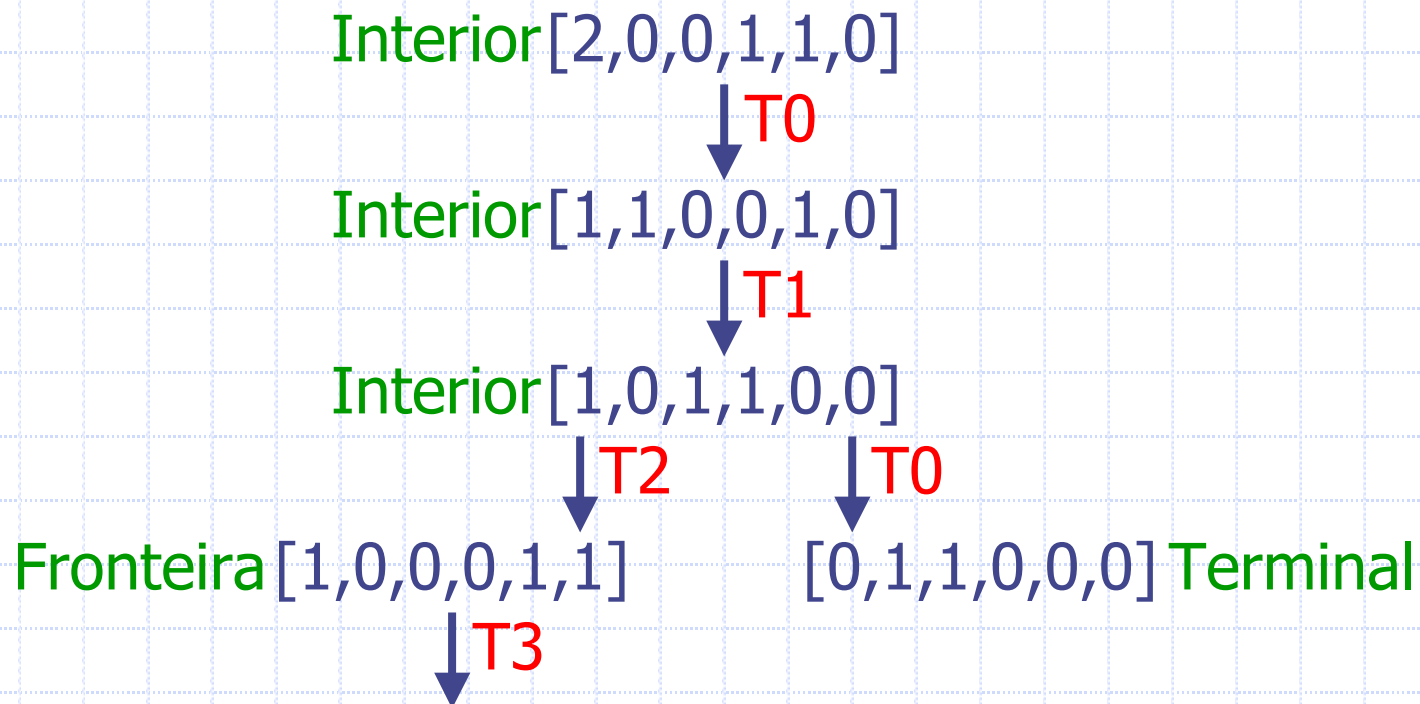
Atelier Simplex: Árvore



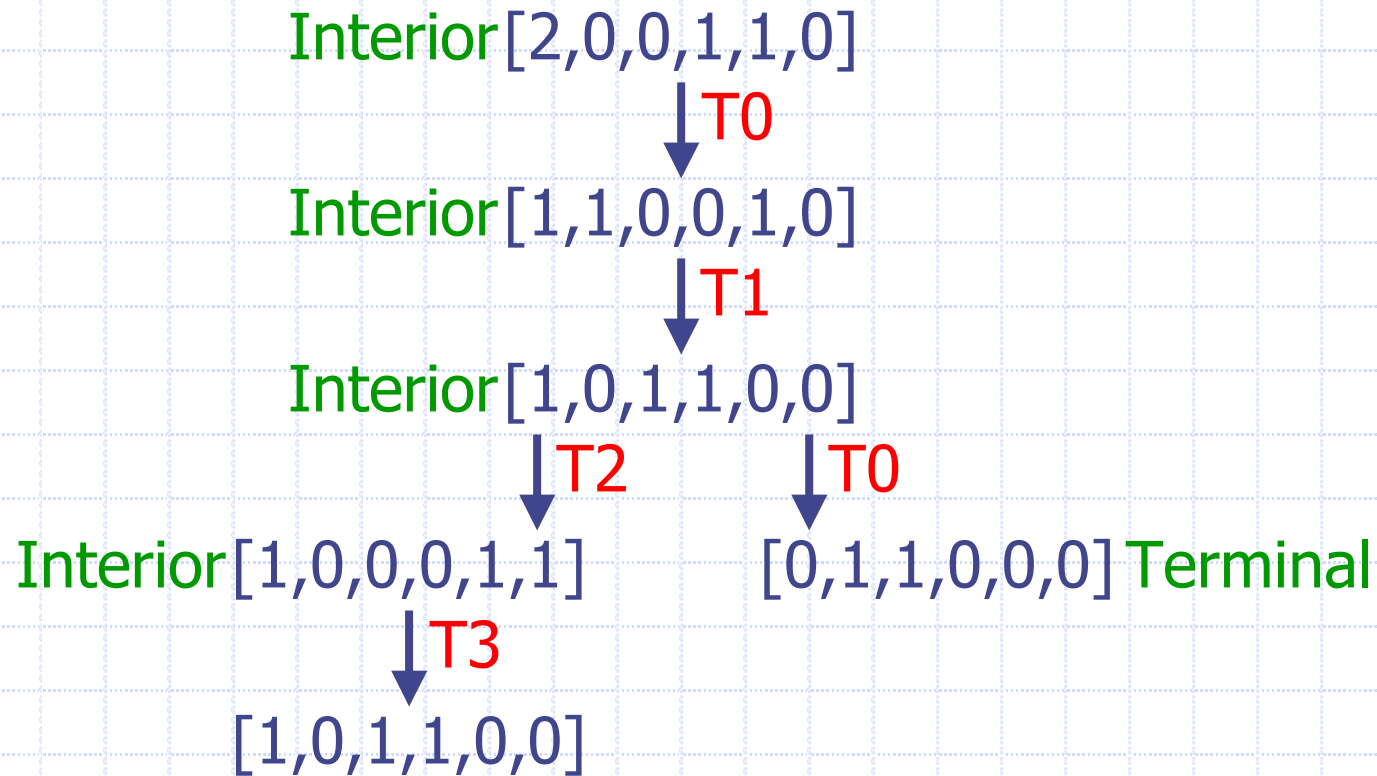
Atelier Simples: Árvore



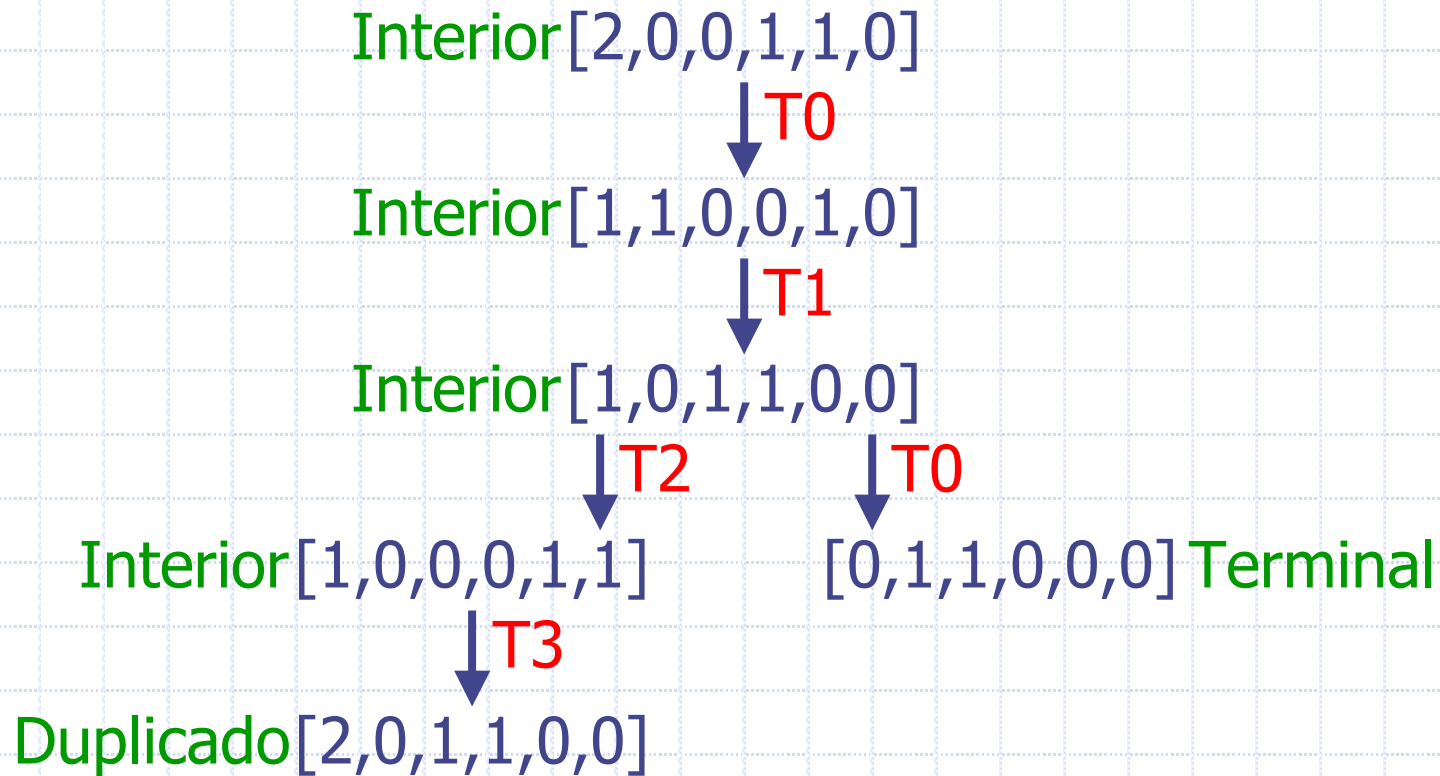
Atelier Simplex: Árvore



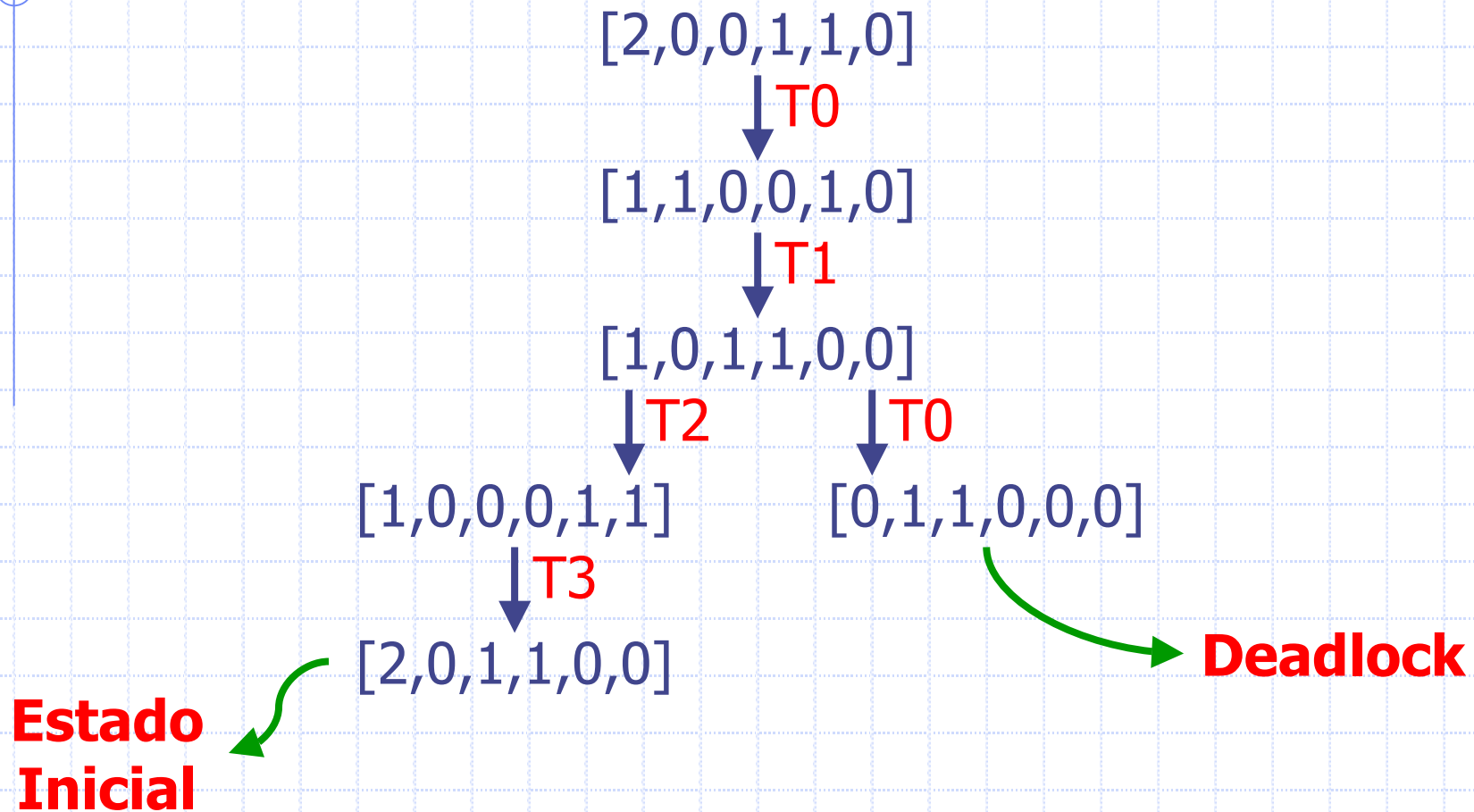
Atelier Simplex: Árvore



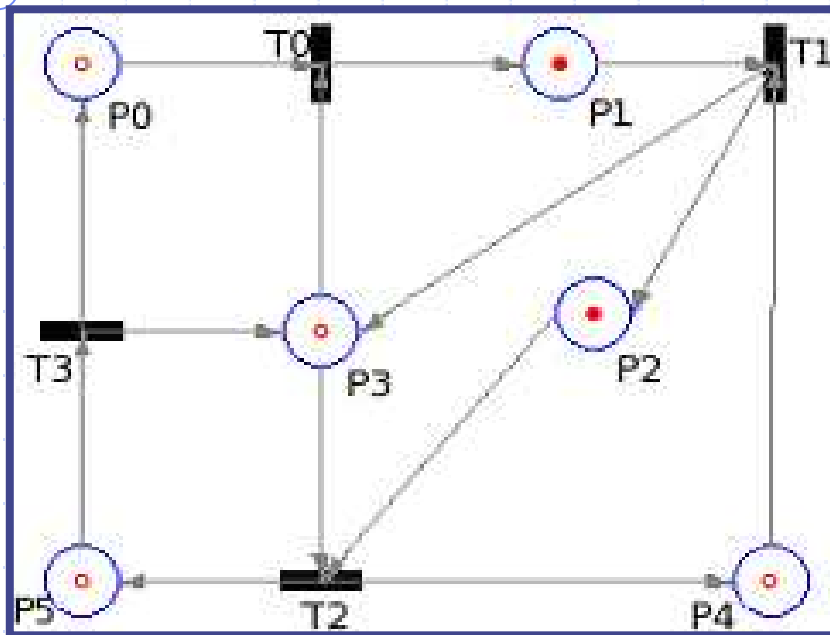
Atelier Simples: Árvore



Atelier Simples: Árvore



Atelier Simples: Propriedades



- ◆ Limite = 2
- ◆ Não reinicializável
- ◆ Não conservativa
- ◆ Não é viva
- ◆ Possui *deadlock*

Rede em Estado
de *Deadlock*

Trabalhos Relacionados

Atributos	JARP	PetriTool	jPNS	JSARP
Composição visual	➔	➔	➔	➔
Simulação interativa	➔	➔	➔	➔
Simulação automática		➔	➔	
Geração de árvore alcançabilidade		➔		➔
Verificação de propriedades	➔	➔		➔
Resultados gráficos				➔
Persistir rede	➔	➔	➔	➔

Ferramenta: ARP

- ◆ DOS 3.0 ou superior
- ◆ Edição de Redes - texto
- ◆ Rdp temporizadas
- ◆ Análise por Enumeração de Estados – Árvore de Alcançabilidade
- ◆ Cálculo de Invariantes
- ◆ Verificação de Equivalência
- ◆ Simulação
- ◆ Avaliação de Desempenho

Ferramenta: ARP

```
C:\WINDOWS\system32\cmd.exe - arp2-4.exe

ARP - 2.4 LCMI-EEL-UFSC
Current Net : D:\REDESP~1\ARP\ARP24BIN\EX01.PM
Directory   : D:\REDESP~1\ARP\ARP24BIN

aNalisys           Simulation           Edit           Help
PeRformance Evalua Verification       Print          Quit

===== Petri Nets Editor ===== L: 16 C: 1 Ins

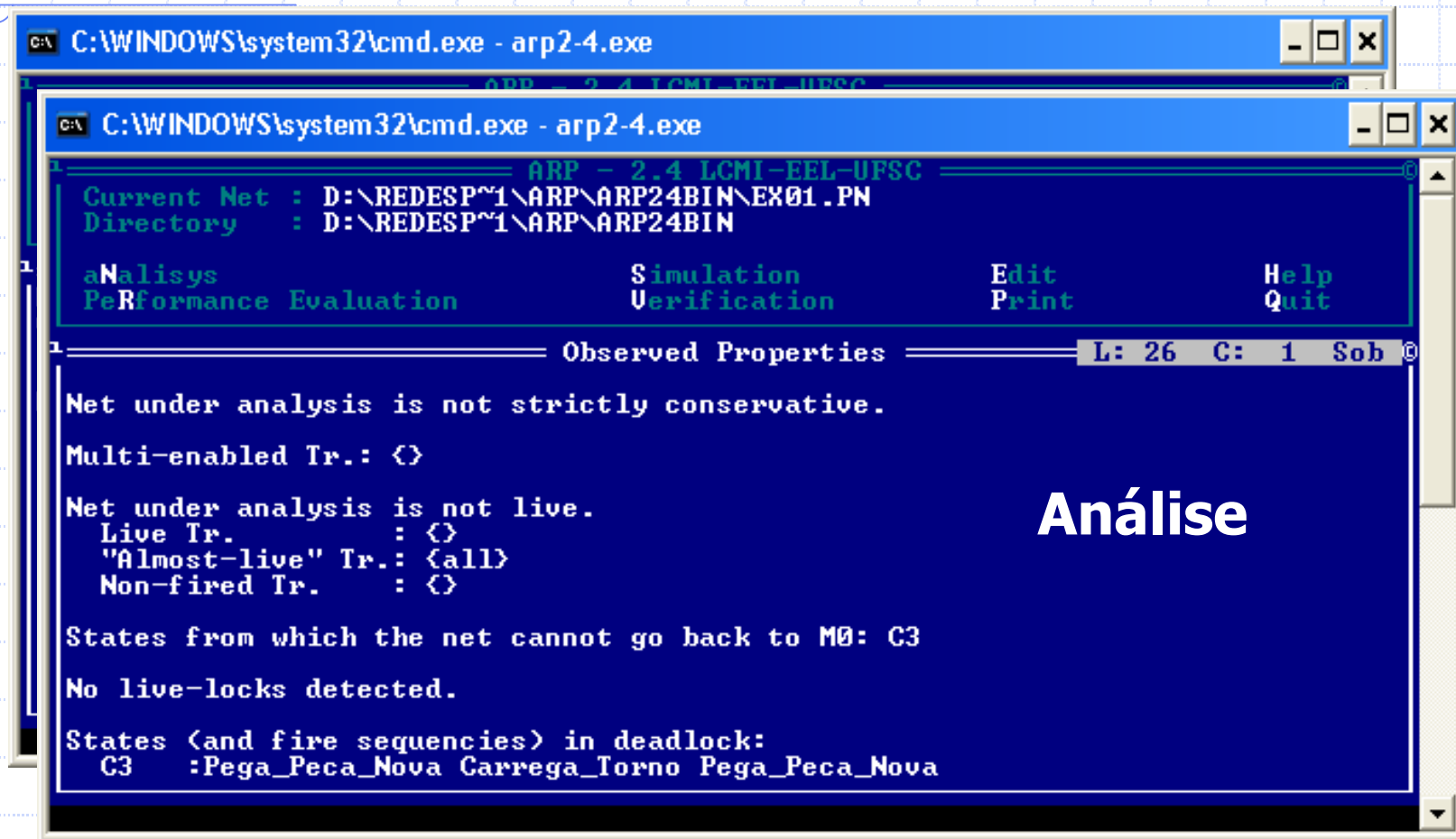
Net Atelier_Simples ;
Const
  Pecas_Entrada = 2 ; { pecas na entrada da máquina }

Nodes
  { armazen de pecas }
  Armazen       : Place (Pecas_Entrada) ;

  { nodos do robo que alimenta o torno }
  Robo_Livre    : Place (1) ;
  Robo_Carregando,
  Robo_D Descarregando : Place ;
  Pega_Peca_Nova : Transition [1,10] ;
  Solta_Peca_Pronta : Transition [3, 3] ;
  Carrega_Torno,
  Descarrega_Torno : Transition [5,15] Normal (5) ;
```

Edição

Ferramenta: ARP



The screenshot shows the ARP 2.4 tool interface. The title bar reads "C:\WINDOWS\system32\cmd.exe - arp2-4.exe". The main window displays the following information:

```
ARP - 2.4 LCMI-EEL-UFSC
Current Net : D:\REDESP~1\ARP\ARP24BIN\EX01.PN
Directory  : D:\REDESP~1\ARP\ARP24BIN

aNalisys          Simulation          Edit          Help
PeRformance Evaluation Verification Print        Quit

===== Observed Properties ===== L: 26 C: 1 Sob ©

Net under analysis is not strictly conservative.
Multi-enabled Tr.: {}

Net under analysis is not live.
Live Tr.          : {}
"Almost-live" Tr.: {all}
Non-fired Tr.     : {}

States from which the net cannot go back to M0: C3

No live-locks detected.

States <and fire sequencies> in deadlock:
C3 :Pega_Peca_Nova Carrega_Torno Pega_Peca_Nova
```

Análise

Ferramenta: ARP

```
C:\WINDOWS\system32\cmd.exe - arp2-4.exe
C:\WINDOWS\system32\cmd.exe - arp2-4.exe
C:\WINDOWS\system32\cmd.exe - arp2-4.exe

Simulation
-----
Fire      State
Return   Trace
Edit      Memory
Places    Help      Quit

Marking
-----
Depth : 2.
Marking: <Armazen, Robo_Livre, Usinando>
Bounds : <Pega_Peca_Nova[1,10], Descarrega_Torno[5,15]>
Fired  : <Pega_Peca_Nova, Descarrega_Torno>

Current Track of Simulation
-----
Reached State      Fired Transition
Inicial State [1] | Pega_Peca_Nova
                [2] | Carrega_Torno
                [3] | Descarrega_Torno
                | Solta_Peca_Pronta

State Visualization L: 1 C: 1 Ins

Fire sequence beginning at Inicial State <depth 0>
and ending at <depth 2>
-> Pega_Peca_Nova [1] Carrega_Torno [5]

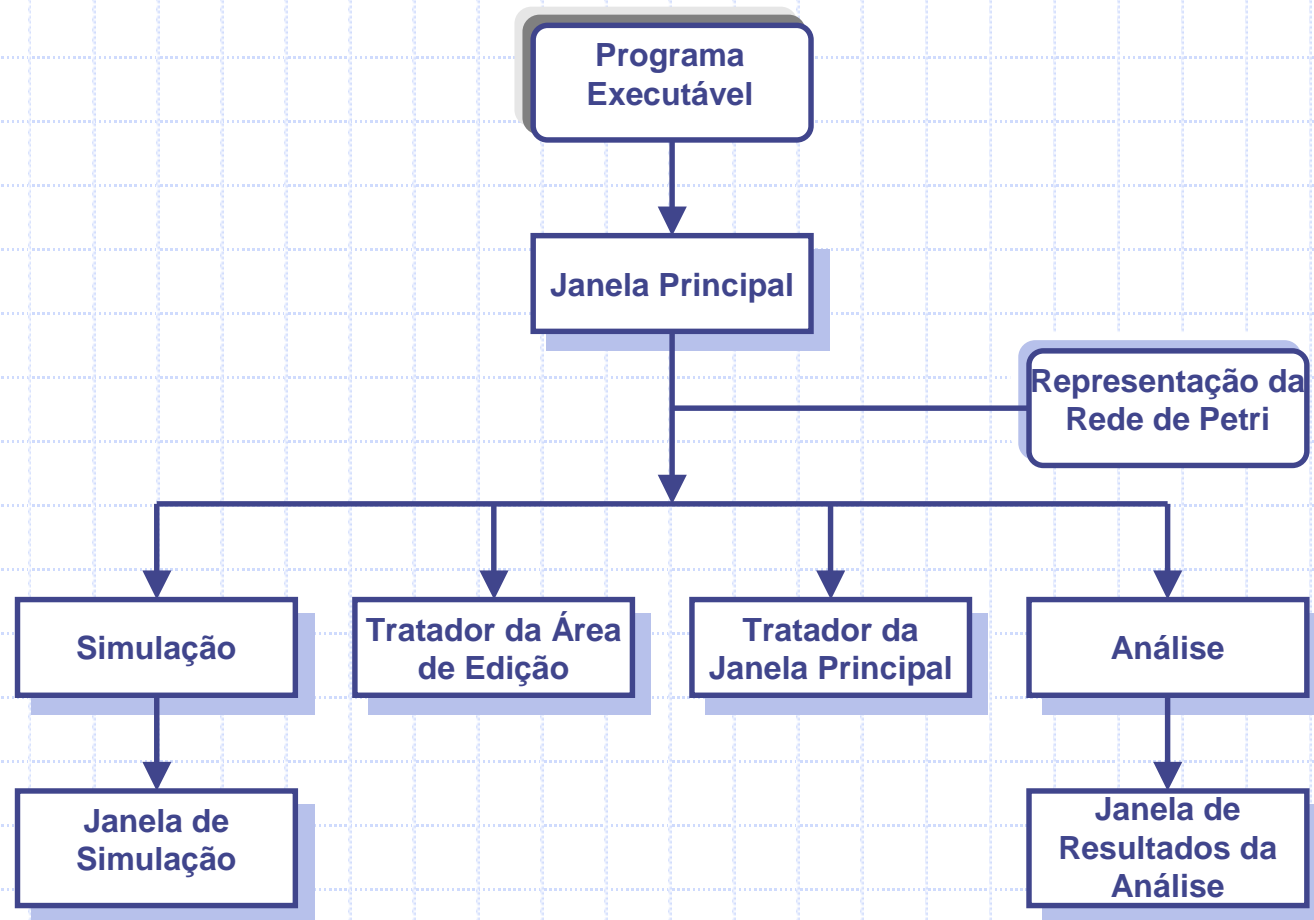
*-----*
```

Simulação

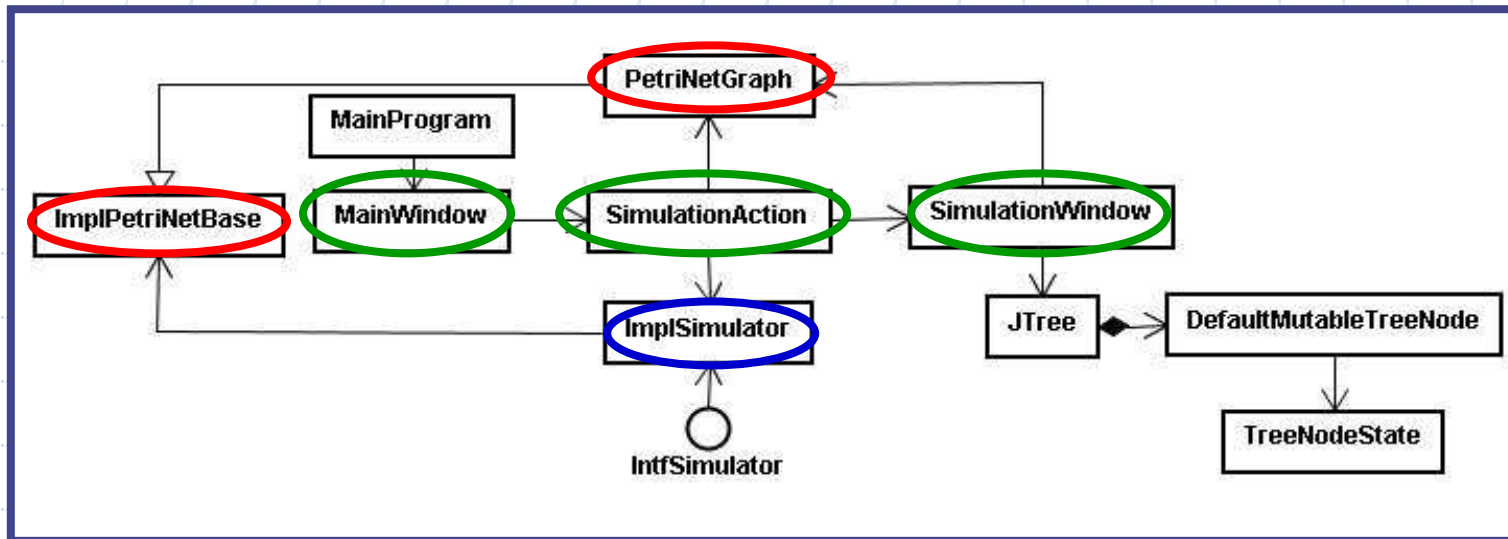
JSARP – Descrição Geral

- ◆ Tecnologia Orientada Objetos
- ◆ Java – Multiplataforma
- ◆ GUI: *Swing* e *API Forms JGoodies*
- ◆ Exporta para XML: *API XStream*
- ◆ Idiomas: Inglês e Português
- ◆ Composição Visual
- ◆ Verificação de Propriedades
- ◆ Gera Árvore de Alcançabilidade
- ◆ Simulação Interativa

JSARP – Arquitetura



JSARP – Estrutura do Código



- ◆ As ações que dirigem a simulação são capturadas através da Classe *SimulationAction*
- ◆ Notifica as ações para a interface gráfica, classe *SimulationWindow*, e para a classe que representa a rede estendida, *PetriNetGraph*
- ◆ A simulação é realizada pela classe *ImpSimulator*, que opera sobre a classe *ImpIPetriNetBase*

JSARP – Formato XML

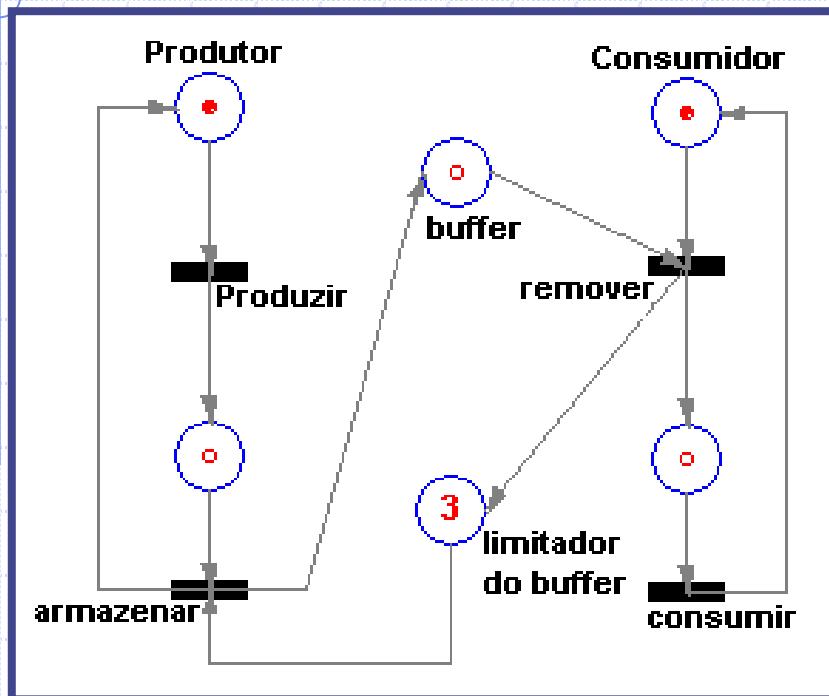
```
<br.uerj.petrinetanalyzer.gui.objects.PetriNetGraph>
  <nome>RedeSimples.xml</nome>
  <listLugar>
    <br.uerj.petrinetanalyzer.gui.objects.PlaceGraph>
      <x>27</x>
      <y>69</y>
      <nome>P0</nome>
      <posicao>0</posicao>
      <fichas>2</fichas>
      <identificador>P0</identificador>
    </br.uerj.petrinetanalyzer.gui.objects.PlaceGraph>
  </listLugar>
  <listTransicao>
    <br.uerj.petrinetanalyzer.gui.objects.TransitionGraph>
      <orientation>1</orientation>
      <x>83</x>
      <y>69</y>
      <nome>T0</nome>
      <posicao>0</posicao>
      <identificador>T0</identificador>
      <seft>0.0</seft>
      <slft>0.0</slft>
      <curvaDensidade>0</curvaDensidade>
    </br.uerj.petrinetanalyzer.gui.objects.TransitionGraph>
  </listTransicao>
```

...

Formato *PNML*

```
<?xml version="1.0" encoding="ISO-8859-1"?><pnml xmlns = "">
  <net id = "n1" type = "http://www.irt.rwth-aachen.de/download/netlab/pntd/pnsmNet">
    <name><text>Petri net1</text></name>
    <place id = "p1">
      <graphics>
        <position x = "300"
                  y = "100"/>
        <dimension x = "40"
                  y = "40"/>
      </graphics>
      <initialMarking>
        <text>1</text>
      </initialMarking>
      <capacity>
        <text>1</text>
      </capacity>
    </place>
    <transition id = "t1">
      <graphics>
        <position x = "300"
                  y = "200"/>
        <dimension x = "40"
                  y = "40"/>
      </graphics>
    </transition>
    ...
```

JSARP – Exemplo



Produtor e Consumidor
com Buffer Limitado

- P0 – Produtor pronto p/ armazenar
- P1 – Consumidor pronto p/ retirar do buffer
- P2 – Consumidor pronto p/ consumir
- P3 – Limitador do buffer
- P4 – Buffer
- P5 – Produtor pronto para produzir
- T0 – Produzir
- T1 – Armazenar no buffer
- T2 – Remover do buffer
- T3 – Consumir

JSARP – Exemplo

The screenshot displays the JSARP software interface, titled "JSARP - Simulator/Analyzer of Petri Nets # ProdutorConsumidorBufferLimitado.xml". The interface includes a menu bar (File, Simulation, Analysis, Help) and a toolbar with buttons for Save, Move, Delete, Place, Trans, Arc, Start, Stop, and Back. A central workspace shows a Petri net diagram with five places and two transitions. The places are represented by blue circles with red dots (tokens). The transitions are represented by black rectangles. A central place contains 3 tokens. The diagram illustrates a producer-consumer system with a limited buffer.

Configuration panels on the left include:

- Place's Information:** Name: Limitador_Buffer, Tokens: 3
- Transition's Information:** Name: (empty), SEFT: (empty), SLFT: (empty), Density: Uniform
- Arc's Information:** Place: (empty), Transition: (empty), Weight: 0

A status bar at the bottom indicates "Safe archive successfully".

JSARP – Exemplo

The screenshot displays the 'Information of Analysis' window in JSARP. It is divided into three main sections:

- Reachability Tree:** A hierarchical tree structure showing states from State 0 to State 14. State 0 is the root state with marking [0,1,0,3,0,1]. The tree shows a sequence of transitions leading to State 14 with marking [0,0,1,0,3,1].
- State Description:** A panel showing the current state's markings and available transitions. The markings are: Pronto_armazenar = 0, Pronto_retirar = 1, Pronto_consumir = 0, Limitador_Buffer = 3, Buffer = 0, and Pronto_produzir = 1. The available transitions are: 1) T0 - Produzir.
- Properties of Petri Net:** A text area containing the following information:
 - Name of the Petri Net: ProdutorConsumidorBufferLimitado
 - The net is alive.
 - The net is limited.
 - Limit of the net is equal to 3.
 - The net is conservative.
 - Total of tokens of the net: 5.

Trabalhos Futuros

- ◆ Redes de Petri Temporizadas
 - Usar o arcabouço existente para análise/simulação em RdP temporizadas
- ◆ Verificações de Propriedades
 - Através da representação matricial
 - Através da árvore de alcançabilidade
- ◆ Melhorias na Interface Gráfica
 - Permitir edição de rótulos
 - Menu de ajuda

Conclusão

- ◆ Código fonte e documentação disponível

- ◆ Produção:

- Workshop de Sistemas Operacionais 2007

Fim

◆ Perguntas ?

◆ Contatos:

- felipelino44@gmail.com
- http://br.geocities.com/jsarp_project