Universidade Federal de Santa Catarina

Experiência 3

Análise e Verificação de uma Planta Industrial

Alunos
Ígor Assis Rocha Yamamoto
Luis Felipe Pelison

Professores
Max Hering de Queiroz
Fábio Luíz Baldissera

Maio de 2016

1 Atividades

As atividades desta aula prática são as seguintes: (i) analisar o funcionamento da estação controlada, (ii) corrigir o controlador implementado, para que a estação satisfaça determinadas propriedades, e (iii) remodelar o controlador para levar em conta o trabalho conjugado com a estação seguinte, a Estação de Testes.

O arquivo do Tina a ser utilizado nesta aula prática é o PlantaControlador.ndr, disponível no moodle. Este arquivo contém o modelo em rede de Petri para a Estação de Distribuição controlada pelo CLP. As ações de comando de cada um dos atuadores (cilindro, driver rotativo, dispositivo de vácuo e dispositivo de sopro) foram projetadas pelo engenheiro responsável pela planta, segundo a linha de raciocínio descrita abaixo:

- Se há peça no magazine e o cilindro está recuado, avançar o cilindro;
- Se o cilindro avançou, então há peça na posição E1. Logo, deslocar o driver da posição E2 para a posição E1;
- Se o driver rotativo está em E1, ligar o vácuo;
- Se o driver rotativo está em E1 e o vácuo está ligado, deslocar o driver de E1 para E2;
- Ligar o sopro somente quando o vácuo estiver desligado.

1. **Análise de Alcançabilidade do Sistema em Malha Fechada**

   Analisar o comportamento da Estação de Distribuição, a partir da análise de alcançabilidade do modelo contido em PlantaControlador.ndr.

2. **Verificação Formal do Comportamento do Sistema em Malha Fechada**

   Verificar, usando model-checking, se o modelo controlado satisfaz as propriedades listadas abaixo (indicar quais fórmulas em LTL foram empregadas em cada caso). Se o sistema não satisfizer uma dada propriedade, mostre qual a sequência de eventos leva ao contra-exemplo e simule esta trajetória com o Tina.

   - O sistema não é bloqueante;
   - O cilindro nunca coloca peça em E1, caso já haja peça em E1;
   - O sopro e o vácuo nunca estão acionados simultaneamente;
   - Em algum momento o vácuo será acionado;
   - Se o sopro está desligado, ele permanecerá desligado até que o vácuo seja acionado;
   - Se há peça na posição E1, esta peça eventualmente chegará à posição E2;

- O driver rotativo nunca vai da posição E2 para a posição E1 sem ter deixado a peça que carrega em E2;
- O sopro nunca é acionado sem que haja peça no driver rotativo na posição E2;
- Proponha agora uma outra propriedade de interesse e verifique se o sistema a satisfaz.

3. **Correção das ações de Controle**

   Proponha alterações no controlador, de modo que o sistema em malha fechada satisfaça todas as propriedades listadas no item anterior. Verifique formalmente todas as propriedades testadas para o novo controlador.

4. **Modelagem e Verificação com Inclusão da Estação de Teste:**

   A Figura 1 apresenta em rede de Petri um sistema utilizado para sincronizar a Estação de Distribuição com a Estação de Teste, que não faz parte deste experimento. Este sistema consiste em um bit lógico (sensor óptico) que sinaliza que a Estação de Teste encontra-se em operação. Este bit visa evitar que o driver rotativo desloque uma nova peça para a Estação de Teste (posição E2) caso esta ainda esteja trabalhando.

   - Propor um novo controlador levando em conta esta nova restrição;
   - Propor uma formula LTL para verificar que o driver rotativo respeitará esta nova restrição.

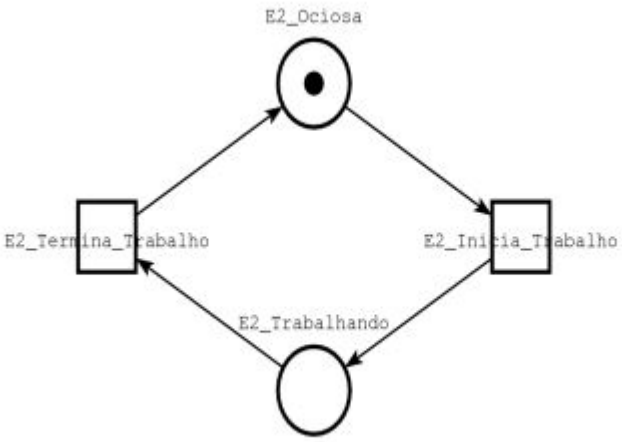

Figura 1: Bit de sincronização entre as plantas 1 e 2

2 Solução

1. **Análise de Alcançabilidade do Sistema em Malha Fechada**

   Usando o recurso de análise de alcançabilidade da ferramenta TINA (Figura 3) podemos verificar as seguintes propriedades da rede de Petri PlantaControlador (Figura 2):

   - Limitada: cada um dos lugares da rede de Petri pode assumir um número limite fixo k de fichas;
   - Não viva: as transições podem ser disparadas muitas vezes arbitrariamente, porém nem sempre uma transição poderá ser disparada a partir de determinada marcação.
   - Não reversível: a marcação inicial nem sempre poderá ser alcançada a partir de uma marcação futura.
   - Sem estado bloqueante: não há nenhum estado do sistema em que uma transição não possa ser disparada.

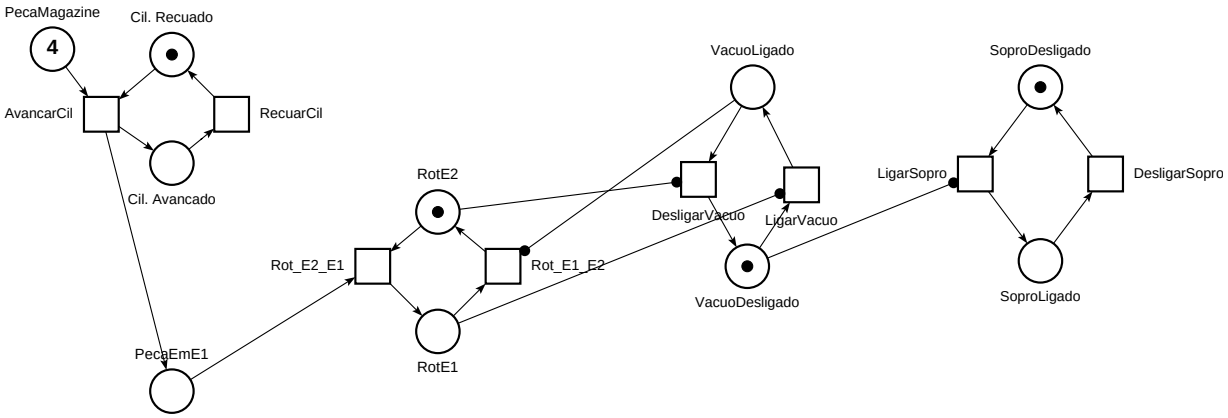

Figura 2: Planta Controlador

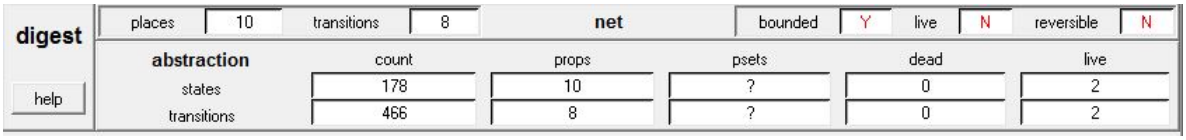

Figura 3: Análise de Alcançabilidade

2. **Verificação Formal do Comportamento do Sistema em Malha Fechada**

   - O sistema não é bloqueante (VERDADEIRO);
     Fórmula Proposicional: [] - dead;
     Saída: Verdade
   - O cilindro nunca coloca peça em E1, caso já haja peça em E1 (FALSO);
     Fórmula Proposicional: [] (PecaEmE1 <= 1);
     Saída: Falso
     Contra-exemplo 1: σ = AvancarCil, RecuarCil, AvancarCil

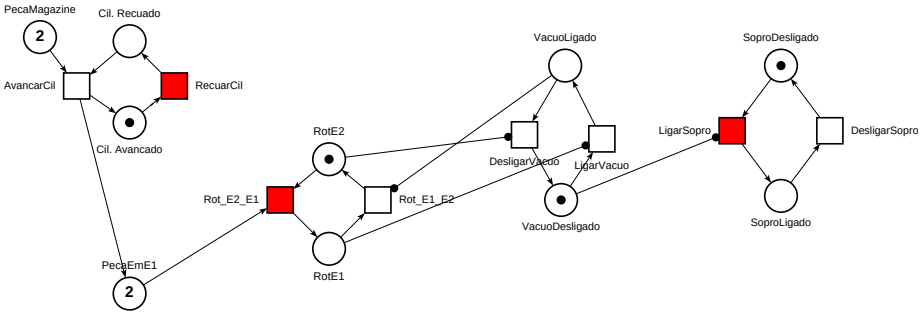

Figura 4: Contra-exemplo 1

   - O sopro e o vácuo nunca estão acionados simultaneamente (FALSO);
     Fórmula Proposicional: [] - (SoproLigado /\$VacuoLigado$);
     Saída: Falso
     Contra-exemplo 2: σ = AvancarCil, RotE2E1, LigarSopro, LigarVacuo

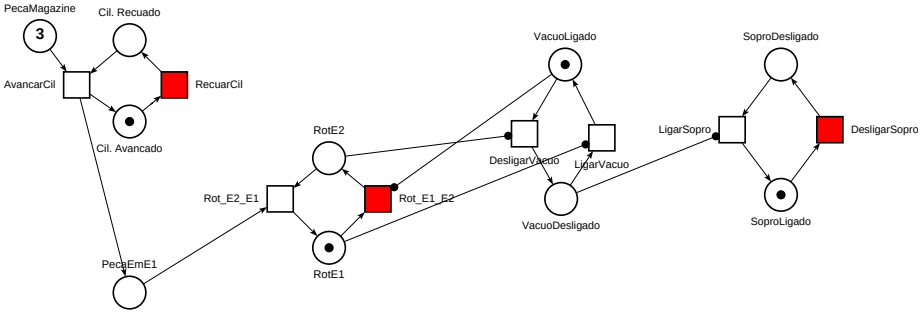

Figura 5: Contra-exemplo 2

- Em algum momento o vácuo será acionado (FALSO);
  Fórmula Proposicional: <> VacuoLigado;
  Saída: Falso
  Contra-exemplo 3: σ = LigarSopro, DesligarSopro, LigarSopro, DesligarSopro, ...

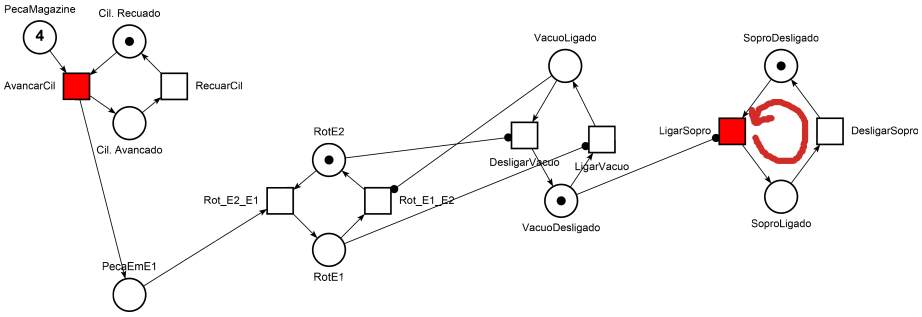

Figura 6: Contra-exemplo 3

- Se o sopro está desligado, ele permanecerá desligado até que o vácuo seja acionado (FALSO);
  Fórmula Proposicional: SoproDesligado =¿ (SoproDesligado U VacuoLigado);
  Saída: Falso
  Contra-exemplo 4: σ = LigarSopro

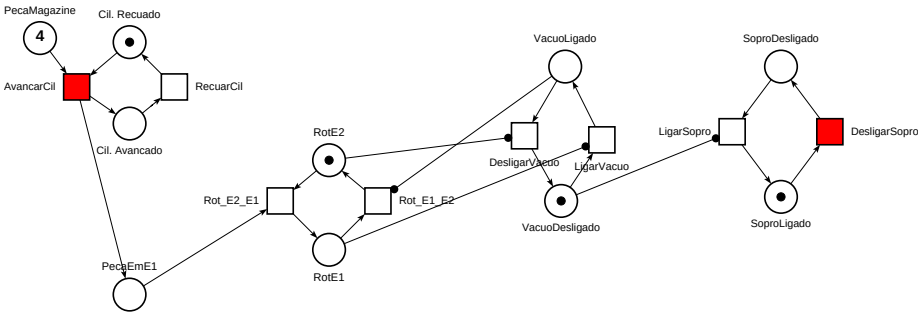

Figura 7: Contra-exemplo 4

- Se há peça na posição E1, esta peça eventualmente chegará à posição E2 (VERDADEIRO);
  Fórmula Proposicional: []($\text{PecaEmE1} => <> \text{PecaEmE2}$);
  Saída: Verdade

Obs.: para a verificação desta propriedade fazer sentido, o sistema original teve que ser modificado para incluir o lugar PecaEmE2 (Figura 8).

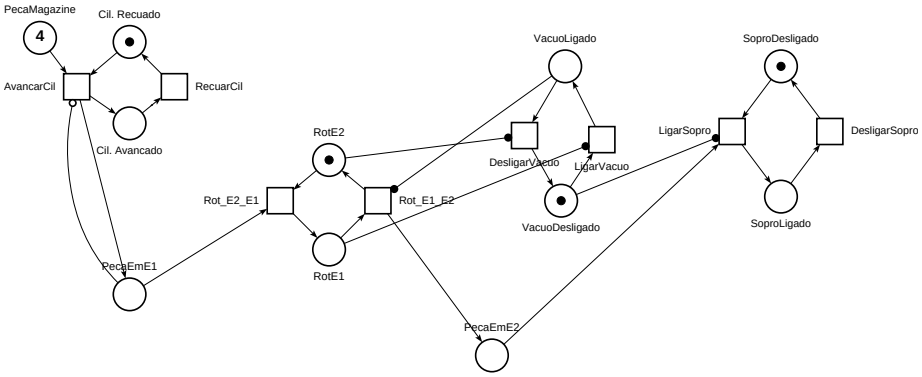

Figura 8: Sistema Modificado para incluir E2

- O driver rotativo nunca vai da posição E2 para a posição E1 sem ter deixado a peça que carrega em E2 (FALSO);
  Fórmula Proposicional: $[]((\text{RotE2} /\backslash PecaNoDriver) => (-RotE1))$;
  Saída: Falso
  Contra-exemplo 5: $\sigma =$ AvancarCil, RecuarCil, AvancarCil, RotE2E1, LigarVacuo, RotE1E2, RotE2E1
  Obs.: para a verificação desta propriedade fazer sentido, o sistema original teve que ser modificado para incluir o lugar PecaNoDriver (Figura 9).

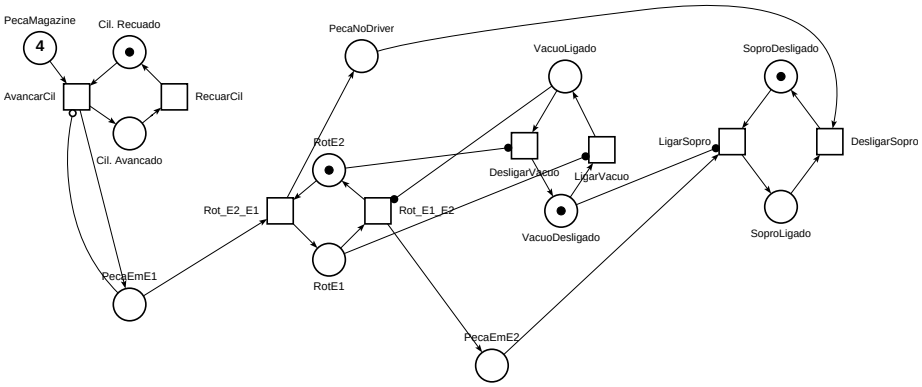

Figura 9: Sistema Modificado para incluir PecaDriver

- O sopro nunca é acionado sem que haja peça no driver rotativo na posição E2 (FALSO);

  Fórmula Proposicional: [](($SoproDesligado U SoproLigado)=>RotE2);

  Saída: Falso

  Contra-exemplo 6: σ = AvancarCil, RecuarCil, AvancarCil, RotE2E1, LigarVacuo, RotE1E2, RotE2E1

- Proponha agora uma outra propriedade de interesse e verifique se o sistema a satisfaz.

  Propriedade: O cilindro nunca avança sem que exista peça no magazine (VERDADEIRO)

  Fórmula Proposicional:-PecaMagazine U -CilAvancado

  Saída:Verdade

3. **Correção das ações de Controle**

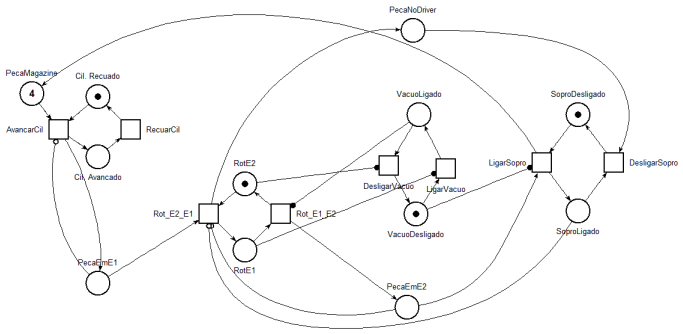

Figura 10: Sistema Modificado

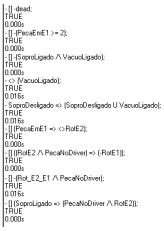

Figura 11: Validação do novo Sistema

4. **Modelagem e Verificação com Inclusão da Estação de Teste:** Inserindo o bit de transição, a rede fica mostrada na Figura 12:

   No model check, verificamos que o driver rotativo nunca deslocará uma peça para posição E2 caso a estação de teste esteja trabalhando com a seguinte fórmula proposicional:

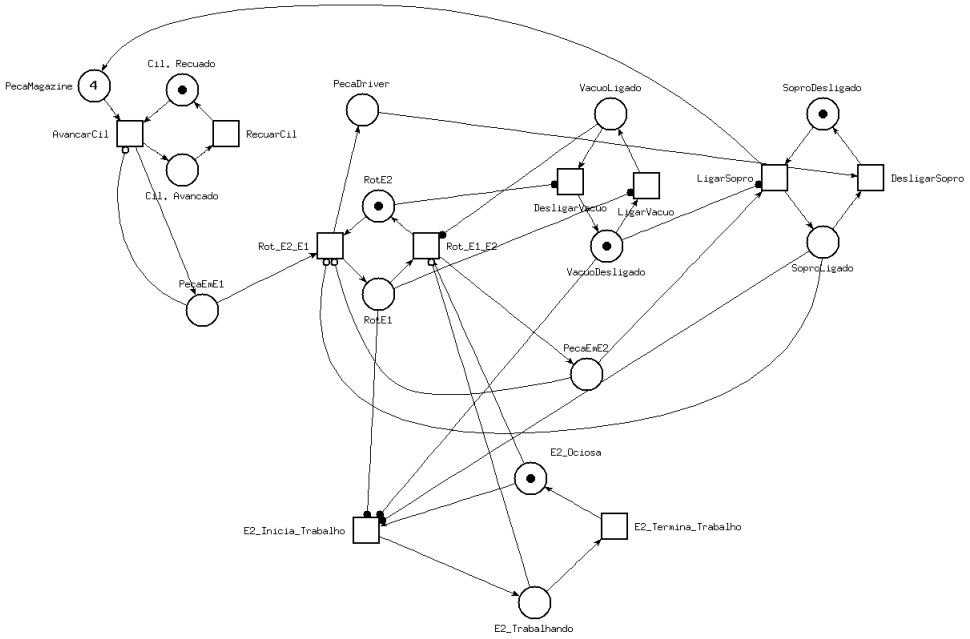

Figura 12: Modelo com bit de transição

[ ] - (E2_Trabalhando /\RotE2);

Com a fórmula proposicional acima, obtivemos a resposta TRUE, verificando que quando a Estação de Teste está trabalhando, o driver rotativo nunca deslocará uma peça para E2.