Universidade Federal de Santa Catarina

Relatório Final

Passeio dos Filósofos

Alunos
Fernando Battisti
Iago de Oliveira Silvestre
Ígor Assis Rocha Yamamoto
Thiago Maurici Espindola

Professores
Fábio Luíz Baldissera
José E. R. Cury
Max Hering de Queiroz

Julho de 2016

Sumário

1 Problema

1.1 Descrição

O sistema definido para um conjunto qualquer de processos e de recursos deve satisfazer as seguintes propriedades:

- Qualquer processo deve acessar uma sequência ordenada de recursos, a partir de qualquer recurso.
- Qualquer recurso é acessado por um único processo ao mesmo tempo (*Propriedade de exclusão mutua*).
- Qualquer recurso solicitado será sempre atribuído em algum momento futuro (*Propriedade de justiça - "no starvation"*).

Os processos devem ter um comportamento que respeite as condições seguintes:

- Um processo pode solicitar um recurso somente se possuir o recurso anterior ou se não possuir qualquer recurso (*Propriedade de ordenamento entre os recursos*).
- Após liberar um recurso que ele possue, nenhum processo voltará a solicita-lo imediatamente (*Propriedade de uso, uma vez só*).
- Se cada processo obtiver cada recurso que solicita, ele liberará sempre qualquer recurso que lhe pertence (*Propriedade de liberação de recurso*).

A situação a evitar é conhecida como "gridlock"e definida por:

- cada recurso pertence a um processo,
- cada processo quer obter o novo recurso,
- nenhum processo libera seu recurso atual.

1.2 Comentários

Contrariamente ao caso do problema do almoço dos filósofos onde cada processo compete repetidamente por uma única seção critica, cada processo no caso do passeio dos filósofos compete por vários recursos em momentos diferentes, sendo que estes são solicitados seguindo uma ordem específica. Finalmente, o problema do passeio dos filósofos pode ser visto a partir de uma analogia com uma "rótula"num cruzamento múltiplo, dividida em partes iguais (vistas como recursos), na qual processos (vistos como processos) entram e saem (de e para várias direções). Neste problema, utilizaremos oito seções da "rótula"(ou recursos) e processos (ou processos) vindo de e indo para 4 direções. A Figura 1 mostra a rótula de forma esquemática.

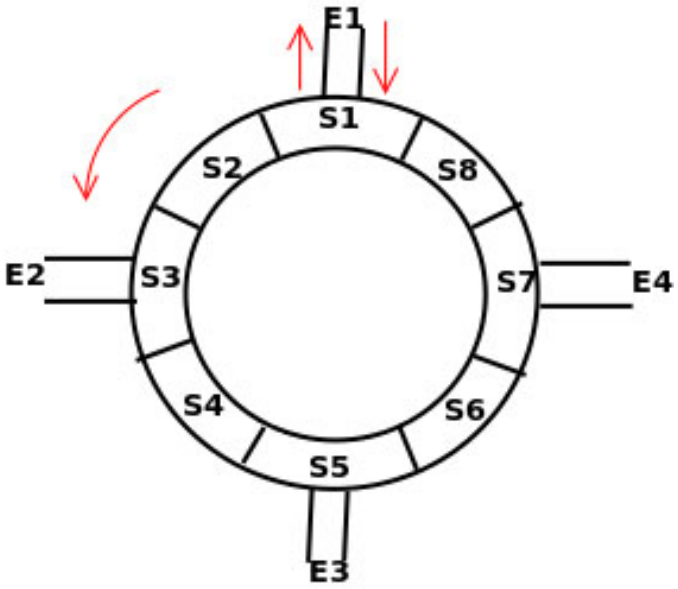

Figura 1: Rótula

2 Solução

2.1 Considerações Iniciais

A solução proposta neste relatório leva em consideração que existe um número de processos (filósofos), desconhecidos a priori, que competem por 8 recursos (secções da rótula - Figura 1).

A modelagem da planta foi feita sob o ponto de vista dos elementos da rótula: 8 seções ($S_i, i = 1, ..., 8$) e 4 entradas ($E_j, j = 1, ..., 4$), com estados informando sobre a presença de filósofos. Enquanto as especificações respeitam as propriedades enunciadas no problema.

Cada processo chega na rótula a partir de uma das entradas E_j com destino de saída (por uma das 4 direções) estabelecido. Conhecido o destino de cada processo, o mesmo deve seguir um caminho fixo para cumpri-lo.

A modelagem proposta pode ser compreendida levando-se em conta a existência de 4 "camadas de rótula"($C_k, k = 1, ..., 4$) superpostas. Onde cada camada de rótula contém todos os caminhos que conduzem o processo a sua destinação fornecida ao chegar na entrada da rótula.

Todos os modelos desenvolvidos durante o projeto foram criados com o auxílio da ferramenta computacional *supremica*.

2.1.1 Camadas da Rótula

O conceito abstrato das camadas da rótula, representadas na Figura 2, foi criado para que haja diferenciação dos caminhos que cada processo deve seguir para sair pela direção já definida na entrada. Os processos podem ingressar em qualquer camada (C_k) a partir de de qualquer entrada (E_j). Os processos devem sair pela direção correspondente de sua camada:

- **Camada 1 (C_1):** saída da rótula por E_1;
- **Camada 2 (C_2):** saída da rótula por E_2;
- **Camada 3 (C_3):** saída da rótula por E_3;
- **Camada 4 (C_4):** saída da rótula por E_4.

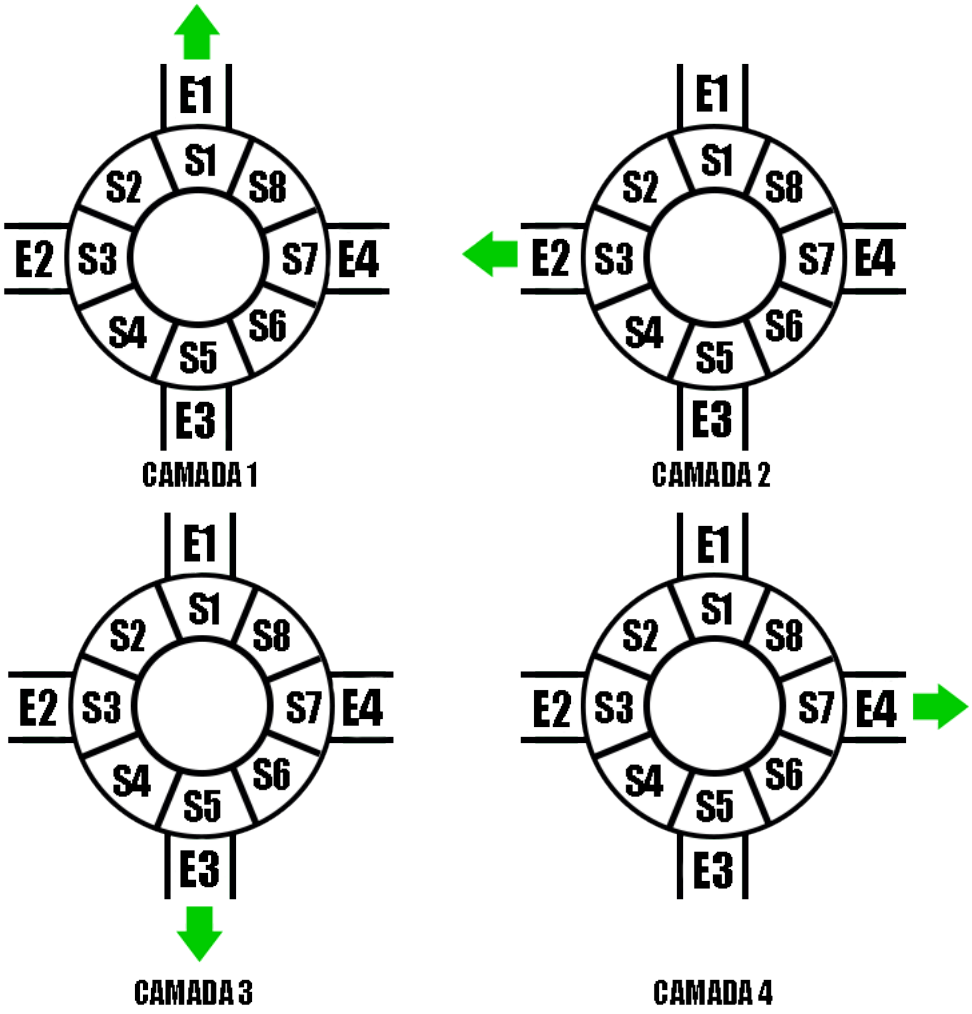

Figura 2: Representação das Camadas da Rótula

2.1.2 Alfabeto do Sistema

Cada camada da rótula possui seu próprio alfabeto com os seguintes eventos:

- **cj_Ck:** processo chega (ocupa) à entrada E_j com destino a E_k;
- **ej_Ck:** processo entra (ocupa secção da rótula, vindo da entrada) pela camada C_k partindo da entrada E_j;
- **pri_Ck:** processo pega (ocupa secção da rótula, vindo de outra secção) o recurso S_i, estando ele na camada C_k;

- **Lri_Ck:** processo libera (desocupa secção da rótula em direção à outra secção) o recurso S_i da camada C_k.

Além dos eventos próprios de cada camada, existe 1 evento independente:

- **sj:** processo sai (desocupa secção em direção à saída) da rótula por E_j.

Quanto a controlabilidade dos eventos, temos:

- **controláveis:** ej_Ck, pri_Ck, Lri_Ck, sj;
- **não-controláveis:** cj_Ck.

2.2 Modelagem da Planta

2.2.1 Secções da Rótula (Recursos S_i)

Estados:

- **Si_L:** recurso S_i livre;
- **Si_O:** recurso S_i ocupado;

Descrição: Para os modelos dos recursos S_i, foram considerados 2 estados denominados Si_L e Si_O, que representam um recurso livre e um ocupado por um processo, respectivamente. Para os recursos S_i adjacentes às entradas E_j, após ocorrer um evento pri_Ck ou um $\text{ej}_1\text{\_Ck}_2$, os mesmos passam a ficar ocupados e só ficaram livres novamente caso ocorra Lri_Ck ou sj. Já para os recursos S_i não adjacentes à E_j, a ocorrência de um evento pri_Ck, ocupa S_i, que só ficará livre novamente caso ocorra Lri_Ck.

Existem 8 modelos de recursos (um para cada secção). Para ilustração dos mesmos, apresentamos aqui somente um exemplo de recurso adjacente à uma entrada (**S1**: Figura 3) e outro não adjacente (**S2**: Figura 4).

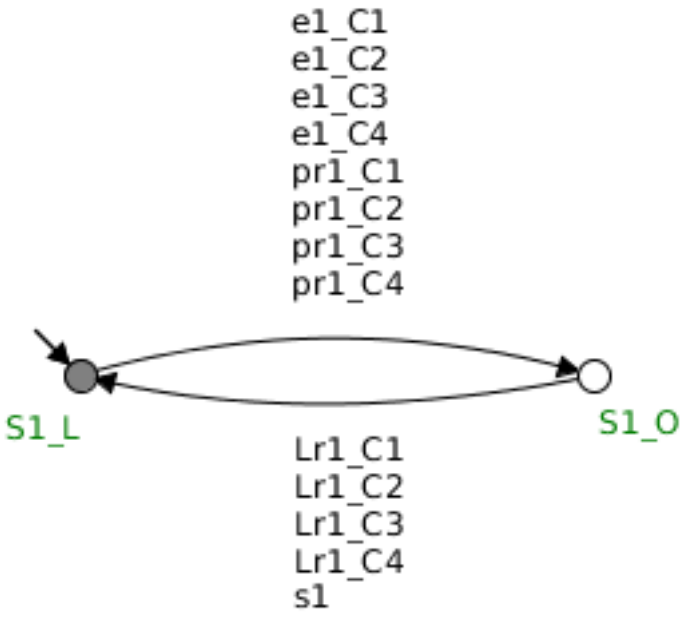

Figura 3: Recurso S1

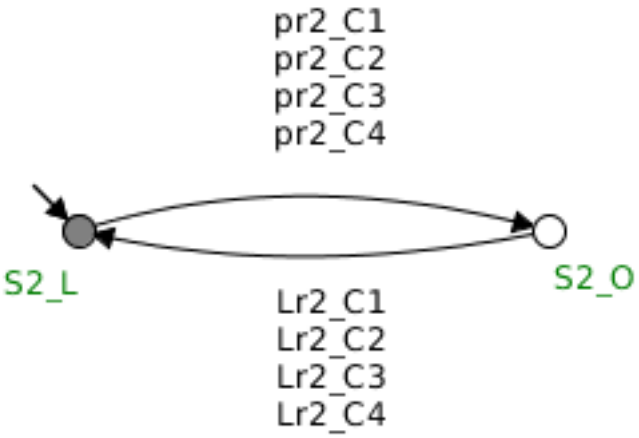

Figura 4: Recurso S2

2.2.2 Entradas da Rótula (E_j)

Estados:

- **Ej_L:** direção E_j livre;
- **Ej_O:** direção E_j ocupada;

Descrição: Para os modelos das entradas, consideramos 2 estados nomeados Ej_L e Ej_O que representam, respectivamente, quando uma entrada está livre ou ocupada. Quando o autômato está no estado Ej_L e ocorre $cj_1\_Ck_2$ a entrada passa para a ficar Ej_O e então só ficará livre novamente caso ocorra $ej_1\_Ck_2$.

Temos 4 modelos para as entradas e, para não deixarmos o relatório cansativo, como os outros são análogos, apresentamos aqui somente **E1**.

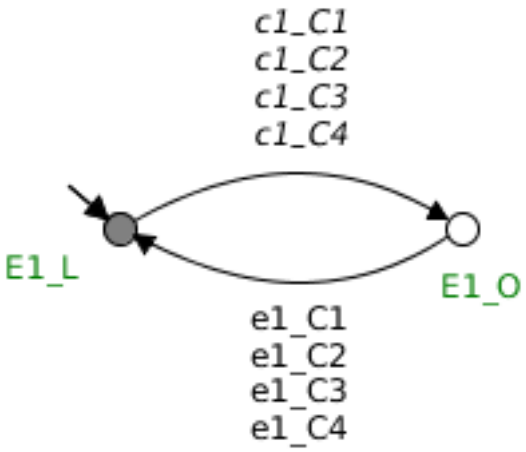

Figura 5: Entrada E1

2.3 Modelagem das Especificações

2.3.1 Propriedade de Ordenamento Entre os Recursos

Os processos devem acessar os recursos respeitando uma sequência ordenada (no sentido anti-horário). Foram criados modelos com as condições de acesso de cada recurso em cada camada. (8 recursos x 4 camadas = 32 modelos de acesso), denominados **Ck:Acesso_Si**.

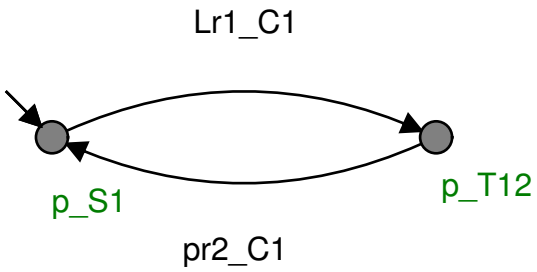

Figura 6: Acesso ao Recurso S_2

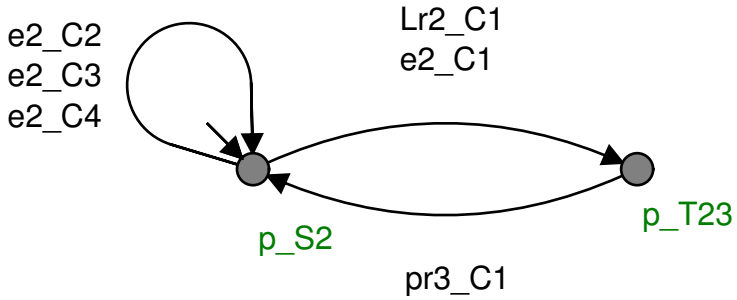

Figura 7: Acesso ao Recurso S_3

2.3.2 Diferenciação entre Camadas

Cada camada possui seu próprio alfabeto e representa a saída por um local específico. Para diferencia-las, foram modelados diferentes autômatos de modo que se um recurso pertencente a determinada camada ele não pode passar para outra camada, pois isso alteraria seu destino.

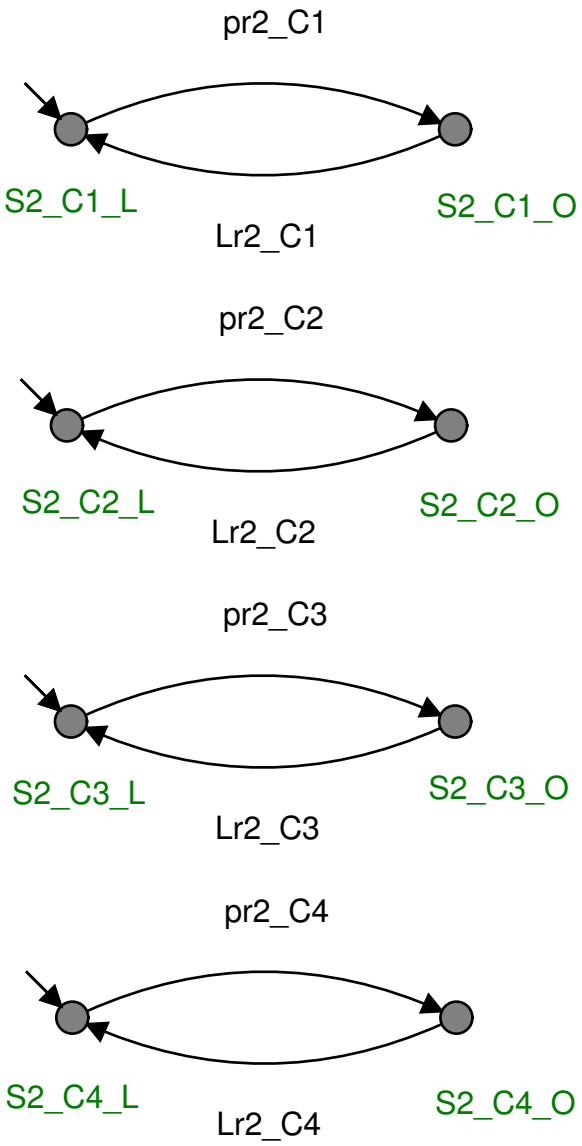

Figura 8: Diferenciação entre camadas para S_2

2.3.3 Propriedade de Justiça

Para esta propriedade, foram modeladas as saídas das rótulas. Para que um processo possa sair, ele deve estar ocupando o recurso S_i que está em contato com a saída E_j que deseja sair (ao entrar na rótula ele já sabe o seu destino). Então, toda vez que ocorre pr1_C1 (processo pega recurso S_1 com destino a E_1), pr3_C2 (processo pega S_3 com destino a E_2), pr5_C3 (processo

pega recurso S_5 com destino a E_3) e pr7_C4 (processo pega recurso S_7 com destino a E_4), ele não poderá mais liberar o recurso que pegou, somente sair.

Temos 4 modelos para as saídas e, para não deixarmos o relatório cansativo, como os outros são análogos, apresentamos aqui somente **C1:Saida_E1**.

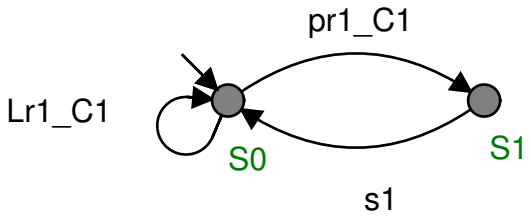

Figura 9: Saída da Camada 1 da Rótula (via E_1)

2.3.4 Propriedade de Uso, uma Vez só

Aqui queremos garantir que após liberar um recurso que um processo possui, nenhum processo voltará a solicita-lo imediatamente. Para isso, ao ocorrer Lri_Ck, impedimos que o recurso volte a ser ocupado através de pri_Ck ou $\text{ej}_1\text{\_Ck}_2$ (para os recursos S_i em contato com as entradas E_j) até que o processo que havia liberado este recurso pegue o recurso seguinte.

Foram modelados 8 autômatos para satisfazer esta propriedade, um para cada recurso e, para não deixarmos o relatório cansativo, como os outros são análogos, apresentamos aqui somente **S1:Uso** e **S2:Uso**.

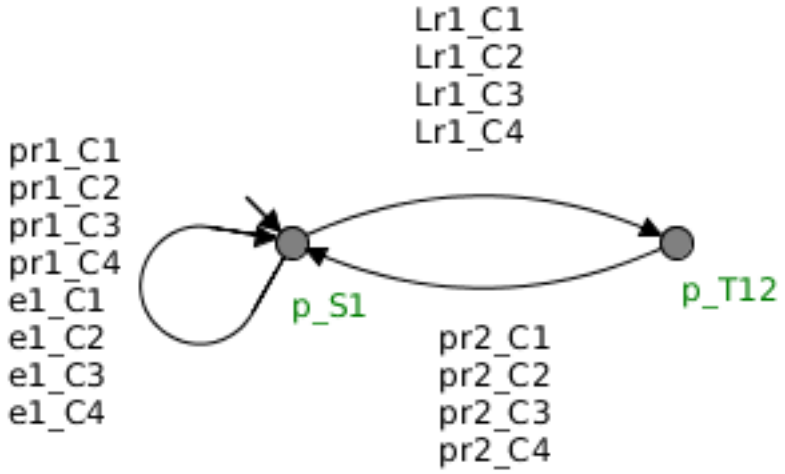

Figura 10: Uso do Recurso S_1

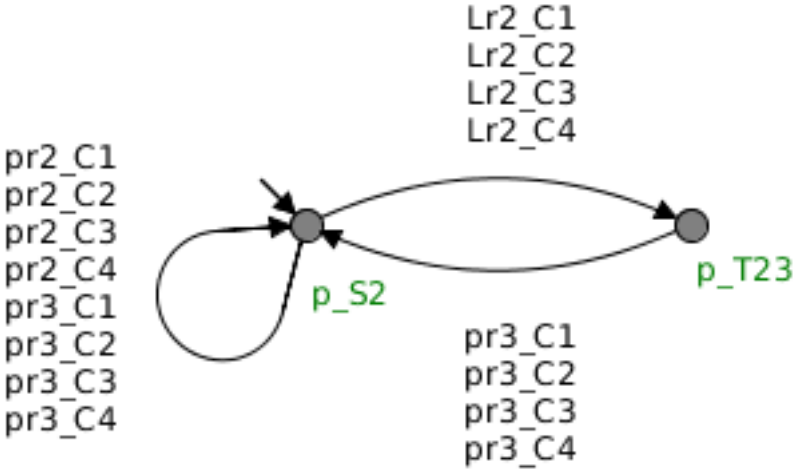

Figura 11: Uso do Recurso S_2

2.4 Análise do Modelo

A partir dos modelos das plantas e das especificações descritas nas seções 2.2 e 2.3, foram realizadas inspeções no sistema através do simulador da ferramenta computacional a fim de detectar situações indesejáveis (bloqueio).

Uma situação de bloqueio específica foi identificada por meio da simulação. A Figura 12 mostra a situação, conhecida domo *gridlock*. Fisicamente, esta situação representa que todos os processos estão em um recurso S_i, desejam acessar o recurso S_{i+1} e nenhum processo libera seu recurso atual. Nenhum processo também está no recurso S_i que faz contato com o destino E_j que ele deseja sair.

2.5 Síntese do Controlador

2.5.1 Síntese Monolítica

Devido a um elevado número de estados presentes na modelagem do problema, a ferramenta computacional não foi possível capaz de calcular um supervisor monolítico.

2.5.2 Síntese Modular

Os supervisores modulares (Figura **??**) foram obtidos com o uso da função *synthesize* do *supremica*. Os supervisores resultantes garantem a controlabilidade do sistema, porém a verificação de bloqueio não pode ser computada devido, novamente, ao tamanho dos modelos construídos.

2.5.3 Supervisor para Evitar o *Gridlock*

A impossibilidade de se fazer o cálculo do supervisor monolítico, foi contornada com a implementação de um supervisor para resolver a situação indesejada específica do *gridlock*. Através de um supervisor (Figura 13) feito manualmente, o problema pôde ser solucionado: o modelo obtido permite que a rótula fique completamente ocupada sem que haja bloqueio. O supervisor controla a ocupação de secções adjacentes às entradas, permitindo o ingresso de processos somente se a rótula tiver menos de 7 secções ocupadas ou se 7 secções estão ocupadas e há um processo dentro da rótula imediatamente ao lado da saída para qual ele está destinado (Figura 14).

Figura 12: Gridlock

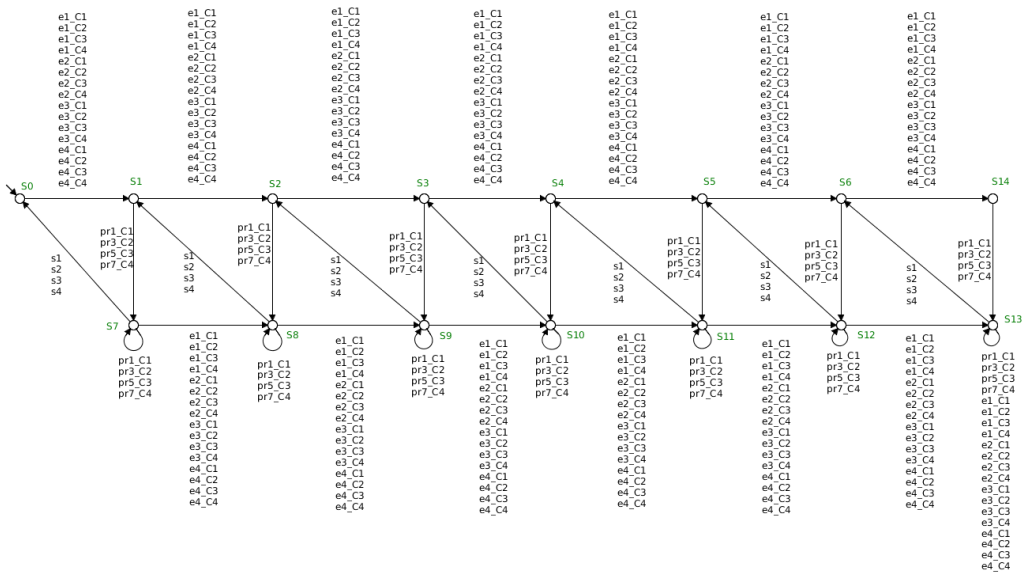

Figura 13: Supervisor

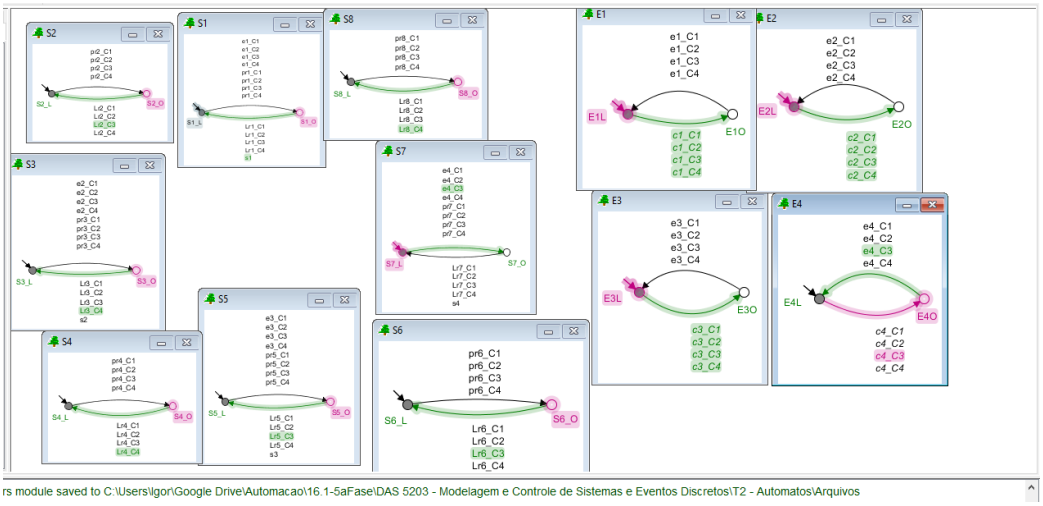

Figura 14: Situação em que a rótula pode se encher sem ocorrer *gridlock*

2.6 Conclusão

2.6.1 Sobre o Problema Proposto

Neste trabalho abordamos o problema do passeio dos filósofos. Os componentes físicos da rótula foram todos representados através do modelo da planta, enquanto as propriedades do sistema foram respeitadas através dos modelos de especificação. Após a fase de construção do modelo, o grupo fez a análise das situações indesejadas ao sistema. Infelizmente, a verificação formal de bloqueio e a síntese de supervisores monolíticos através de ferramenta computacional não foi factível devido ao sobredimensionamento do problema. Identificou-se, porém, uma situação específica indesejada: o *gridlock*. Um supervisor foi construído para eliminação do bloqueio gerado pelo *gridlock*. Com a síntese manual do supervisor, foi possível alcançar um modelo que permite a ocupação completa da rótula sem que aconteça bloqueio induzido pelo *gridlock*. Todavia, não é possível afirmar formalmente que o modelo obtido não permite estados bloqueantes.

2.6.2 Avaliação dos Resultados Obtidos

Os resultados finais do modelo proposto não contemplam de maneira plena a solução desejável para o problema. A impossibilidade de verificação e síntese de controladores, tornaram a solução ideal impraticável. Alguns aspectos da abordagem tomada para o problema puderam ser debatidos em conjunto com os professores da disciplina após a fase de desenvolvimento do projeto.

O grupo, então, pôde identificar diversos fatores que elevaram a complexidade do modelo. Entre eles, destacam-se a representação de estados e eventos desnecessários para a solução do problema: no modelo proposto foi adicionado um estado de transição entre cada secção da rótula, dobrando o tamanho de estados. A simples passagem direta de um processo entre duas secções adjacentes, representaria muito bem a situação física da rótula.

O modelo complexo construído, e consequentemente, a solução parcial obtida foram reflexos da má interpretação do grupo das especificações fornecidas. Uma discussão mais aprofundada com os professores da disciplina durante a fase de desenvolvimento do projeto poderia ter direcionado a resolução do problema para um caminho desejável. Apesar dos esforços despendidos e da aplicação do grupo para resolver o problema, a falta de uma metodologia bem definida para atacar o problema é outro fator que fica de aprendizado para a equipe. Em linhas gerais, o trabalho realizado pelo grupo foi de grande valia para a formação dos participantes, apesar dos resultados parciais obtidos.