PUC RIO - DEPARTAMENTO DE INFORMÁTICA

Monografias em Ciência da Computação

Editor: Carlos J. P. Lucena

Agosto, 97

ESPECIFICAÇÃO DE ESTRUTURAS CONCEITUAIS BÁSICAS ATRAVÉS DE REDES DE PETRI COMPACTAS

Eliana Silva de Almeida
Bruno Maffeo

Departamento de Informática

SUMÁRIO

ESPECIFICAÇÃO
DE
ESTRUTURAS CONCEITUAIS BÁSICAS
ATRAVÉS DE
REDES DE PETRI COMPACTAS

Eliana Silva de Almeida
DI/PUC - Rio
e-mail: eliana@inf.puc-rio.br
Bruno Maffeo
DI/PUC - Rio
e-mail: maffeo@inf.puc-rio.br

PUC-Rio.Inf.MCC26/97

Abstract:
This work presents procedures to formally specify each primitive operational activity contained in activity schemes, that model systems behavior when the Essential Model is used as conceptual framework. This formal specification employs Compact Petri Nets as representation language and, for each procedure presented, the primitive operational activity to be specified belongs to a conceptual pattern typically found in essence modeling. It is suggested that these procedures should be employed in formal specifications of large systems based on Compact Petri Nets, since its use in connection with the structures commonly found in the Essential Model allows a gradual control of representational complexity, otherwise excessive. It is also believed that significant gains of quality and productivity in the systems development process should be achieved through reuse of these conceptual pattern specifications.

Keywords :
Compact Petri Nets, Essential Model, Conceptual Framework, Conceptual Pattern, Activity Scheme, Primitive Operational Activity, Reuse.

Resumo:
Neste trabalho, são apresentados procedimentos para especificar formalmente cada atividade operacional primitiva, contida em esquemas de atividades, que modelam o comportamento de sistemas quando o arcabouço conceitual (“conceptual framework”) baseia-se no Modelo da Essência. Essa especificação formal utiliza Redes de Petri Compactas como linguagem de representação e, em cada procedimento proposto, a atividade operacional primitiva a ser especificada faz parte de uma estrutura conceitual básica (“conceptual pattern”) encontrada tipicamente na modelagem da essência. É sugerido que esses procedimentos sejam utilizados em especificações formais de sistemas de grande porte, baseadas em Redes de Petri Compactas, pois seu emprego, relacionado a esquemas de atividades obtidos na modelagem da essência, permite graduar a complexidade da representação a ser construída, em geral excessiva. Acredita-se também que ganhos significativos de produtividade e qualidade no processo de desenvolvimento de sistemas podem ser obtidos através do reúso das especificações formais associadas a essas estruturas conceituais básicas.

Palavras Chaves:
Redes de Petri Compactas, Modelo da Essência, Arcabouço Conceitual, Estrutura Conceitual Básica, Esquema de Atividades, Atividade Operacional Primitiva, Reuso.

1 INTRODUÇÃO

Durante as últimas duas décadas, a assim chamada área de Métodos Estruturados vem trazendo, para as etapas de Análise e de Design do processo de desenvolvimento de sistemas de software, um conjunto poderoso e diversificado de ferramentas conceituais e técnicas de modelagem. Desde o advento da análise estruturada [1, 2], passando pela Análise Essencial [3] e chegando a análise Estruturada Moderna [4, 5, 6, 7], diferentes autores têm investido considerável esforço no sentido de prover instrumentos de modelagem adequados ao desenvolvimento tanto de sistemas de informação convencionais de grande porte, quanto de sistemas de tempo-real para controle de processos.

Ultimamente, tendo em vista a criticalidade da etapa de especificação de requisitos para o sucesso de um projeto de desenvolvimento, esse esforço tem visado primordialmente estabelecer uma transição suave entre a **declaração de necessidades**, que o cliente/usuário de Informática enuncia informalmente, e uma **especificação conceitual de requisitos** (originalmente conhecida como **projeto lógico**) que traduza formalmente a visão que o **Analista** deve transmitir ao **Projetista** (**Designer**) do sistema a ser desenvolvido.

Apesar de preocupados em trazer o maior rigor possível para esse processo de transição, a estratégia adotada por esses autores para definir e refinar esses instrumentos de modelagem, tem privilegiado, em relação à adoção imediata de aspectos formais clássicos da Ciência da Computação, a pesquisa experimental sistemática e abrangente mais característica de uma disciplina de Ciência da Natureza ou Engenharia. Analisando em retrospecto, parece ter-se acreditado, possivelmente de forma intuitiva, que experimentos "de campo", em projetos relacionados ao desenvolvimento de sistemas realistas destinados a uso intensivo, constituiriam o teste de sobrevivência requerido para esses instrumentos, preliminar a qualquer tentativa de formalização.

Mais recentemente, considerada bem sucedida a etapa da experimentação controlada, iniciaram-se os esforços que visam primordialmente trilhar o necessário caminho da completa formalização das ferramentas conceituais e técnicas de modelagem utilizadas na área dos Métodos Estruturados [8, 9, 10]. O presente trabalho constitui um desses esforços.

Na análise Estruturada Moderna, a especificação conceitual do comportamento de um sistema emprega linguagens de representação gráficas para construir **esquemas de atividades** (originalmente conhecidos como **diagramas de fluxos de dados**) que modelam os elementos funcionais ativos do sistema e as interfaces desses elementos, internas ao sistema e com o ambiente externo. Em seu formato hierarquizado, esses esquemas evidenciam uma decomposição funcional, correspondente a um refinamento descendente ("top-down") da visão do sistema ativo. Nessa abordagem, esse refinamento termina em atividades-folha denominadas **atividades operacionais primitivas**. Por definição, essas atividades não são decompostas funcionalmente em termos de esquemas de atividades; sua especificação detalhada, necessariamente, deve ser realizada em termos de outra linguagem de representação, desejavelmente formal.

No presente trabalho, é apresentado um tratamento formal para a **especificação declarativa** das atividades operacionais primitivas (originalmente conhecida como **mini-especificação**) contidas em um esquema de atividades. Cada especificação refere-se a uma **estrutura conceitual básica** (**conceptual pattern**) comumente encontrada na modelagem de sistemas com base no **Modelo da Essência** [7] — esta abordagem constituindo instância específica, freqüentemente utilizada, de **arcabouço conceitual** (**conceptual framework**) para especificação de sistemas —. É utilizado o formalismo das **Redes de Petri Compactas** [11, 12, 13] para essa especificação declarativa, a qual permanece gráfica, e procura-se demonstrar sua viabilidade através de exemplos em que especificações semi-formais baseadas em listas de Pré/Pós-Condições são convertidas para esse formalismo.

Na seção 2, são apresentados sumariamente, os fundamentos das Redes de Petri Compactas. Na seção 3, são especificados procedimentos para empregar esse tipo de rede a algumas estruturas conceituais básicas típicas. Na seção 4 é ilustrada a utilização dessa linguagem de representação, aplicando-a a exemplos de situações específicas, relacionadas às estruturas conceituais básicas consideradas na seção precedente. A seção 5 apresenta as conclusões deste trabalho. O apêndice descreve a linguagem de anotação associada às Rede de Petri Compactas definidas na seção 2.

2 REDES DE PETRI COMPACTAS

Nesta seção são apresentados, com base em [11, 12], os conceitos básicos relacionados às Redes de Petri Compactas.

2.1 Redes de Petri

Rede de Petri é uma linguagem de representação gráfica que pode ser utilizada para a modelagem conceitual de sistemas. Em geral, uma Rede de Petri compõe-se de:

- **Conjunto de Lugares** representados por circunferências
- **Conjunto de Conexões** representadas por retângulos
- **Conjunto de Ramos Rotulados** representados por setas que unem lugares a conexões.

A figura 1 apresenta uma Rede de Petri onde todos os componentes acima mencionados podem ser identificados.

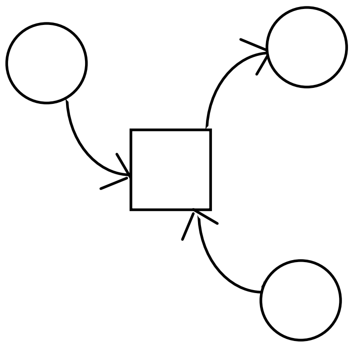


figura 1 - Componentes de uma Rede de Petri

A rotulação dos ramos da Rede de Petri permite estabelecer um papel diferente para cada elemento (lugar ou conexão) da rede. Os ramos possuem duas formas de classificação: por um lado, são classificados como **ramos de entrada** e **ramos de saída**. Por outro lado, podem ser **alteradores, restauradores** e **mantenedores**. A tabela 1 ilustra a notação gráfica associada a essas classificações.

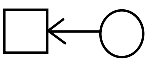
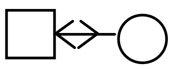
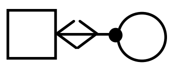
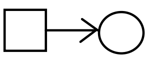
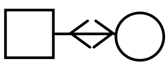
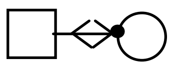
	ramos alteradores	**ramos restauradores**	**ramos mantenedores**
ramos de entrada			
ramos de saída			

tabela 1 - Classificação dos ramos de uma rede de Petri

Para apresentar as regras básicas de funcionamento de uma rede de Petri, será considerada uma abordagem formal designada por modelagem **condição/evento: CEM.** Nessa modelagem, um lugar da rede designa um elemento funcional passivo que permite modelar condições (estados locais) de um sistema. Uma conexão designa um elemento funcional ativo que permite modelar eventos que ocorrem em um sistema. Os ramos definem a participação do lugar na conexão, permitindo modelar a dinâmica do sistema. A modelagem condição/evento é basicamente apoiada por dois tipos de Rede estudadas nesta seção: Redes de Petri **elementares** e Redes de Petri **compactas.**

2.2 Redes de Petri Elementares

Nas Redes de Petri Elementares, cada lugar define uma **marca**, a qual modela uma condição (um estado local) do sistema considerado. Essa marca é representada por uma ficha, denotada por um círculo escuro contido na circunferência que denota o lugar. Essa ficha pode estar presente ou ausente, representando a presença ou ausência da marca. A presença ou ausência da marca no lugar indica o vigorar ou não vigorar da condição modelada. Graficamente [figura 2]:

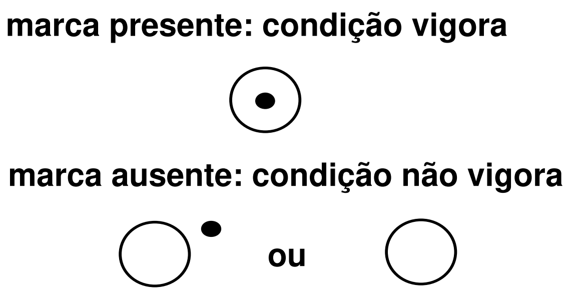


figura 2 - Representação gráfica da marca definida por um lugar em uma Rede de Petri Elementar

Dada uma rede elementar, o conjunto de todas as marcas (presentes e ausentes) da rede denomina-se **marcação** da rede. É essa marcação que modela um estado global do sistema considerado. Por sua vez, uma conexão de rede elementar define uma **alteração**. Uma alteração modela um evento que **pode** ocorrer em determinado estado global do sistema. O efeito dessa ocorrência é modelado através do aparecimento e coincidente desaparecimento de marcas. A ocorrência de uma alteração depende e afeta apenas as marcas de lugares a ela vinculados através dos ramos associados à conexão que define a alteração. A possibilidade de ocorrência da alteração (ou do evento modelado) frente a uma marcação específica é determinada pela seguinte regra de habilitação:

Definição 1 (Regra de Habilitação)

Uma alteração está *habilitada* frente a uma marcação quando:

- suas marcas de entrada (marcas dos lugares ligados à conexão por ramos de entrada), caso a conexão as tenha, estão presentes;
- suas marcas de saída (marcas dos lugares ligados à conexão por ramos de saída), caso as tenha, estão ausentes.

Logo, o fato de uma alteração estar ou não habilitada indica a possibilidade de ocorrência ou não ocorrência do evento modelado. Dessa forma, toda a dinâmica do sistema é modelada através da ocorrência de eventos, isto é, a passagem do sistema de um estado para outro é modelado através da ocorrência de transições entre marcações da rede. Graficamente, o efeito da ocorrência de uma alteração é representado da seguinte forma [figura 3]:

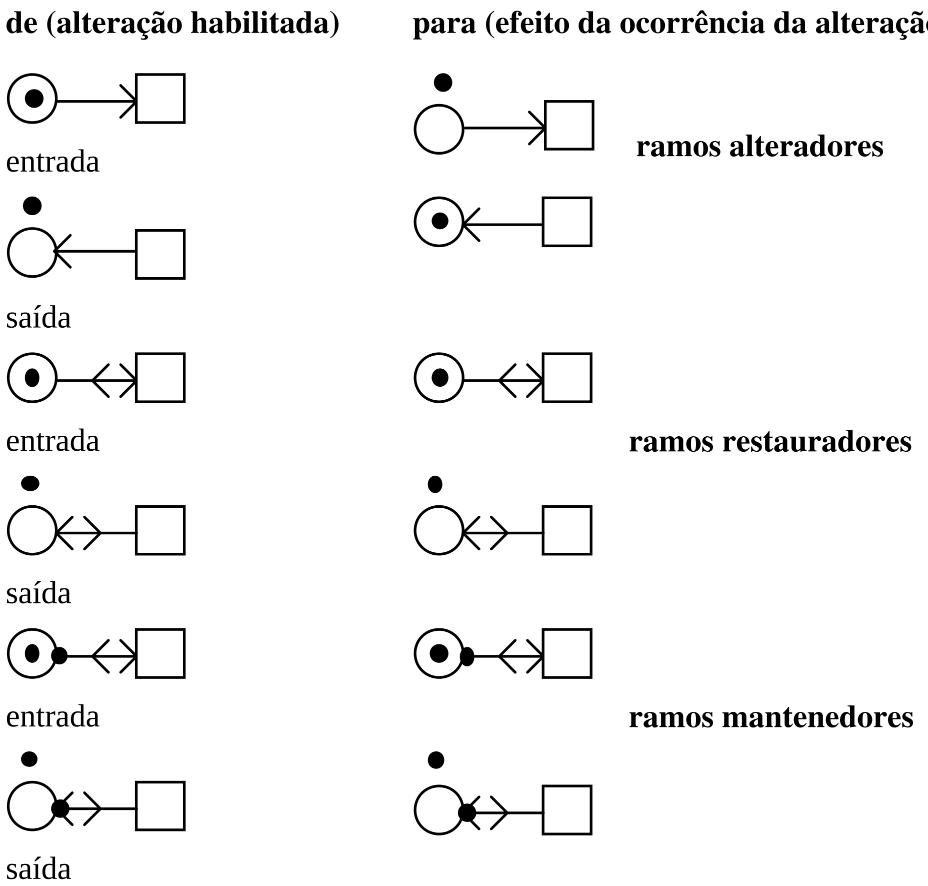


figura 3 - Representação gráfica do efeito da ocorrência de uma alteração em uma Rede de Petri Elementar

Da figura 3 acima, observa-se que, do ponto de vista do resultado final da ocorrência de uma alteração, não há distinção entre ramos restauradores e mantenedores. Esses ramos distinguem-se pela forma como definem a disponibilidade da marca (ausente ou presente) durante a ocorrência de uma alteração. No caso de ramos mantenedores, a marca mantém-se disponível enquanto ocorre a alteração, contrariamente ao caso de ramos restauradores. Neste último caso, a marca é "restaurada" ao final da alteração mas não é "mantida" disponível durante a alteração.

Quando duas alterações estão habilitadas, elas podem manter dois tipos de relação — **independência** e **conflito** — definidas a seguir:

Definição 2 (independência de alterações)

Diz-se que duas alterações habilitadas são *independentes* quando não possuem marca de entrada ou de saída comum ligada às conexões por ramos alteradores ou restauradores, podendo existir marca de entrada ou de saída comum ligada às conexões por ramos mantenedores [figura 4].

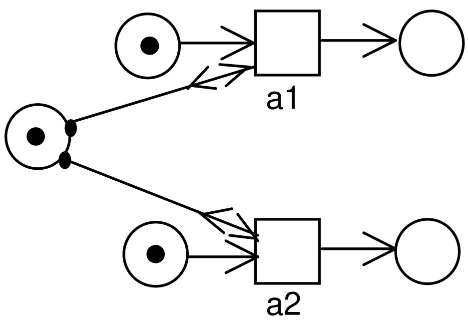


figura 4 - Representação gráfica de duas alterações independentes a1 e a2 em uma Rede de Petri Elementar

Definição 3 (conflito de alterações)

Diz-se que duas alterações habilitadas são *conflitantes* quando possuem ao menos uma marca de entrada ou de saída comum ligada às conexões por ramos alteradores ou restauradores [figura 5].

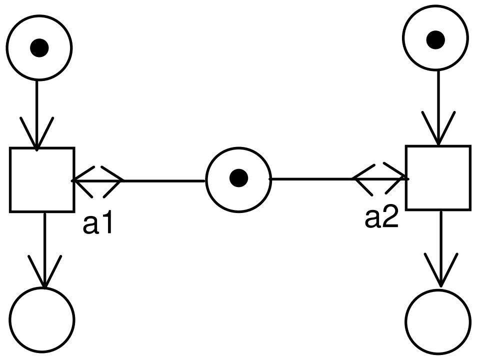


figura 5 - Representação gráfica de duas alterações conflitantes a1 e a2 em uma Rede de Petri Elementar

Utilizando os conceitos acima apresentados, pode-se definir como se dá a ocorrência combinada de alterações. Para esse fim, é apresentada a seguir a definição de **passo**, de efeito da ocorrência de um passo, e de **marcação alcançável**. Esses conceitos permitem compreender todo o comportamento de uma rede a partir de sua marcação inicial.

Definição 4 (passo)

Um conjunto de alterações é um *passo* frente a uma marcação, chamada marcação *precursora*, quando:

- todas as alterações do passo estão habilitadas frente à marcação precursora;
- as alterações do passo são independentes.

Definição 5 (efeito da ocorrência de um passo)

O *efeito* da ocorrência das alterações que definem um passo é a transição da marcação precursora para uma marcação sucessora de tal forma que:

- todas as marcas ligadas às conexões do passo por ramos alteradores de entrada *desapareçam* na marcação sucessora;
- todas as marcas ligadas às conexões do passo por ramos alteradores de saída *apareçam* na marcação sucessora.

Durante a ocorrência de um passo, no caso de uma alteração na qual marcas de entrada/saída estejam ligadas à conexão por ramos restauradores, o resultado da ocorrência da alteração é obtido como o efeito da seqüência de duas ações: uma que retira/coloca a ficha do/no lugar e outra que repõe/retira essa ficha no/do lugar; isto é, a presença/ausência da marca é afetada durante a ocorrência da alteração. Caso as ligações sejam feitas através de ramos mantenedores, a alteração comporta-se como uma ação de consulta a essas marcas; isto é, a presença/ausência das marcas não é afetada durante a ocorrência da alteração.

Definição 6 (marcação alcançável)

Frente a uma marcação inicial da rede, é qualquer marcação obtida pelo efeito da ocorrência de um passo sobre a marcação inicial ou sobre qualquer marcação alcançável em um ou mais passos a partir desta.

Para completar o conhecimento sobre o comportamento de uma rede de Petri elementar, são apresentadas a seguir algumas relações entre eventos modelados pela rede[1].

Definição 7 (seqüência de eventos)

Diz-se que dois eventos estão em *seqüência* quando, para que um evento ocorra, é necessário que outro evento tenha ocorrido antes. As redes abaixo modelam, através das conexões e1 e e2, eventos que estão em seqüência [figura 6]:

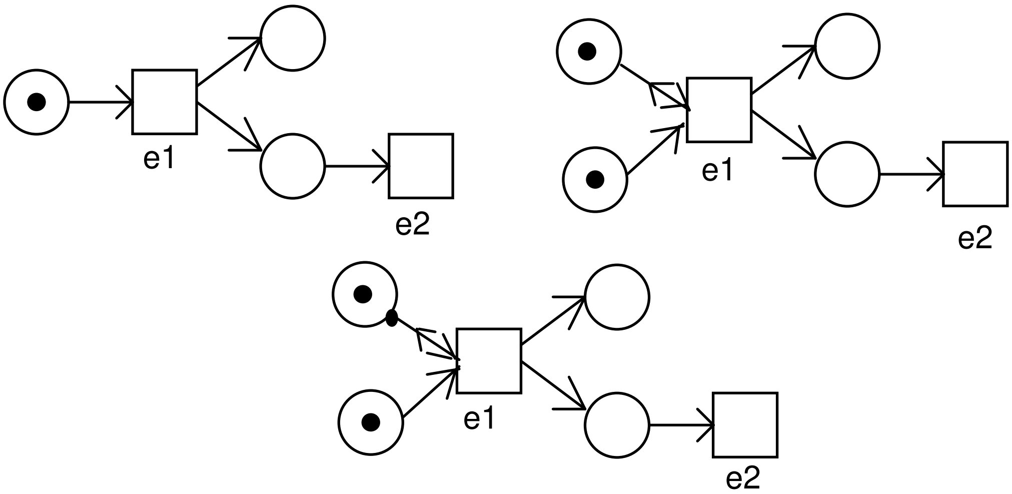


figura 6 - representação gráfica de Redes de Petri Elementares com dois eventos, e1 e e2, em seqüência

Definição 8 (independência de eventos)

Diz-se que dois eventos são *independentes* quando as conexões que os modelam não possuem nenhuma marca de entrada ou saída comum ligada por ramos alteradores ou restauradores, podendo possuir condição de entrada ou saída comum ligada por ramos mantenedores [figura 7]:

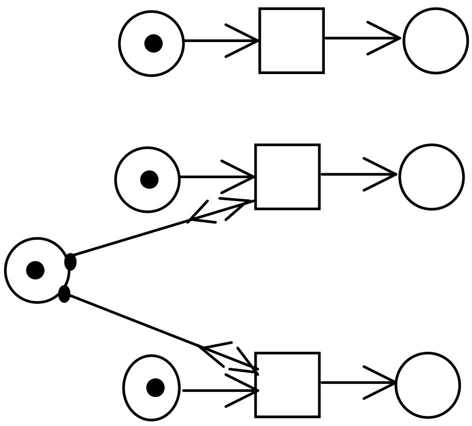


figura 7 - Representação gráfica de Redes de Petri Elementares com conexões que modelam eventos independentes

Definição 9 (conflito de eventos)

Diz-se que dois eventos habilitados são *conflitantes* quando as conexões que o modelam possuem alguma marca de entrada ou saída comum ligada por ramos alteradores ou restauradores [figura 8].

[1] O fato de um evento ocorrer ou não está diretamente relacionado às regras de habilitação.

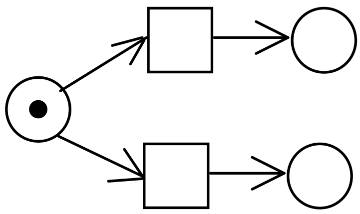


figura 8 - Representação gráfica de uma Rede de Petri Elementar contendo conexões que modelam eventos conflitantes

A seguir [tabela 2], são relacionados os elementos presentes em uma Rede de Petri Elementar e sua interpretação na modelagem condição/evento.

Rede Elementar	**Interpretação C/E**
marca	condição (estado local)
presença de marca	vigorar de condição
ausência de marca	não vigorar de condição
marca de entrada/saída	condição de entrada/saída
marcação da rede	estado global do sistema
alteração	evento
ocorrência de alteração	ocorrência de evento
ocorrência de passo	transição entre estados do sistema

tabela 2 - Interpretação dos elementos que constituem uma Rede de Petri Elementar na abordagem condição/evento

Originalmente, Redes de Petri não modelam a obrigatoriedade da ocorrência de um evento habilitado; modelam apenas que um evento habilitado **pode** ocorrer. Entretanto, os eventos tratados neste trabalho, associados à estimulação de atividades operacionais primitivas através de fluxos de dados/controle discretos, submetem-se a obrigatoriedade de ocorrência. Para modelar a ocorrência obrigatória de um evento habilitado, é aqui utilizada uma notação de ramo, chamada **ramo determinante**, representada graficamente por uma seta tracejada. A semântica desse ramo impõe a ocorrência de qualquer alteração que o utilize, no instante em que se produzir a marcação habilitadora da alteração.

Voltando à definição de Redes de Petri, observa-se que elas podem ter ramos (de entrada/saída) de três tipos: alteradores, restauradores e mantenedores. Será ilustrado agora a importância destes dois últimos ramos (restauradores e mantenedores) no comportamento da rede.

Muitas vezes, é necessário expressar que a habilitação de uma alteração depende não só das marcas modificadas pela sua ocorrência, mas também de outras marcas. Para exemplificar esse fato, pode ser considerada a Rede de Petri Elementar que modela o funcionamento de um semáforo, com o objetivo de descrever sua contínua abertura e fechamento, bem como a passagem de automóveis pelo mesmo. A rede que, aparentemente, modela essa situação está apresentada na figura 9.

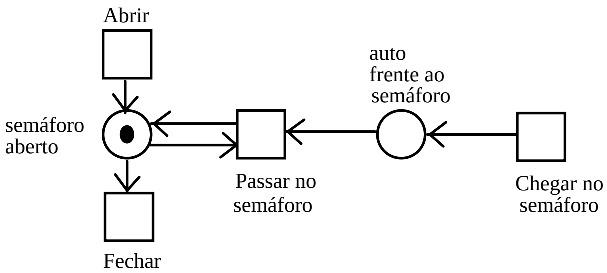


figura 9 - Rede de Petri Elementar que, aparentemente, modela o funcionamento de um semáforo (eventos nunca habilitados)

Nessa rede, observa-se os lugares *semáforo aberto*, modelando a condição do semáforo estar aberto ou fechado, e *auto frente ao semáforo*, modelando a condição do automóvel estar ou não em frente ao semáforo. As conexões *abrir* e *fechar* modelam, respectivamente, a abertura e fechamento do semáforo. A conexão *passar no semáforo* deveria modelar a passagem do automóvel pelo semáforo. No entanto, na rede acima apresentada, a alteração definida por essa conexão jamais estará habilitada. Para isso, é necessário que a passagem de um automóvel pelo semáforo dependa, para sua habilitação, da condição do semáforo (aberto), mas que sua ocorrência não altere essa condição. O modelo correto deve utilizar ramos restauradores, como apresentado na rede da figura 10.

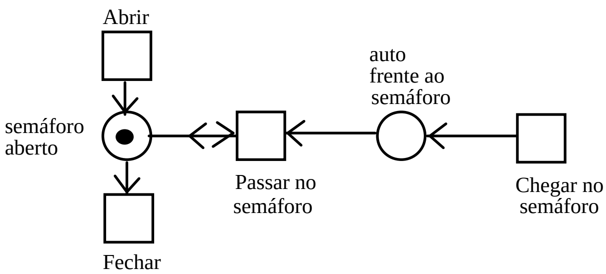


figura 10 - Rede de Petri Elementar que modela o funcionamento de um semáforo

Uma outra situação pode ocorrer ao modelar uma rua com mais de uma pista. Nesse caso, o uso de ramos restauradores redunda em conflito, o que impõe o uso de ramos mantenedores, caracterizando independência entre alterações. A rede seria [figura 11]:

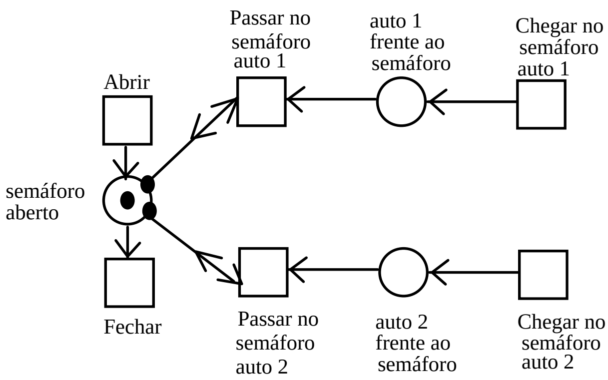


figura 11 - Rede de Petri Elementar que modela o funcionamento de um semáforo para duas pistas (ramos mantenedores)

Ramos restauradores e ramos mantenedores permitem modelar recursos que são compartilhados por mais de uma atividade. A diferença está no fato de que, no primeiro caso, os eventos modelados não podem ocorrer simultaneamente porque isso requereria o uso exclusivo da condição de habilitação modelada por algum lugar comum. Já no segundo caso, os eventos modelados podem ocorrer simultaneamente, pois a condição de habilitação modelada pelo lugar comum é replicada tantas vezes quantos forem os ramos mantenedores a ele ligados. Nas figuras 12 e 13, está definida a semântica desses ramos, respectivamente:

Ramo Restaurador

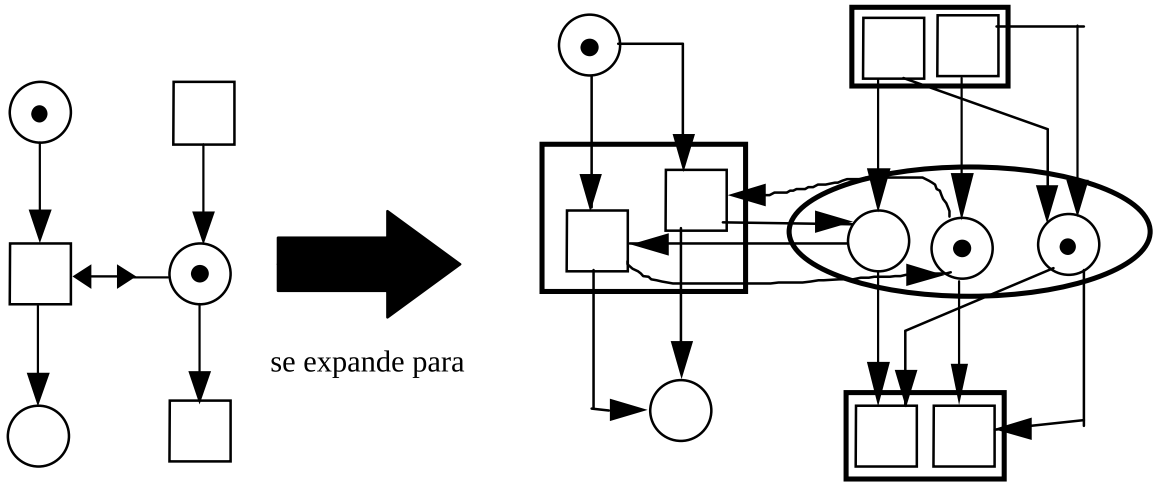


figura 12 - Representação gráfica do significado de um ramo restaurador

Ramo Mantenedor

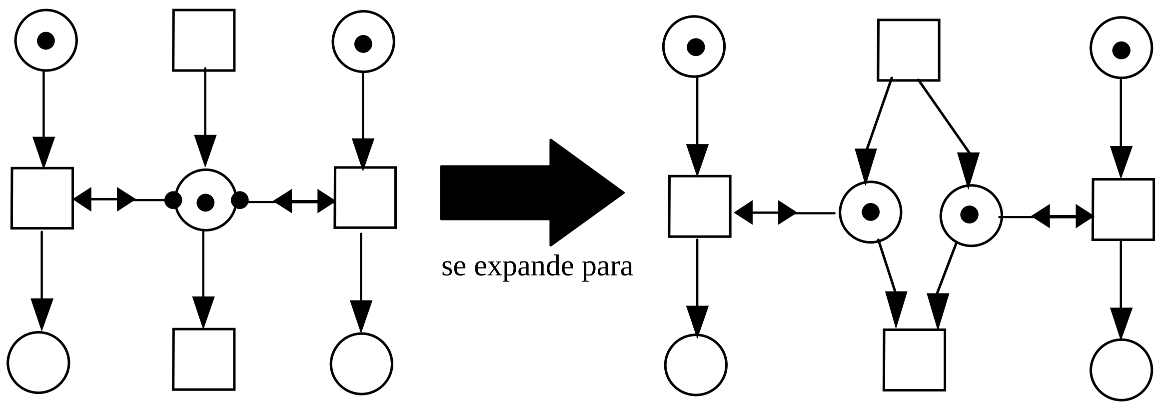


figura 13 - Representação gráfica do significado de um ramo mantenedor

2.3 Redes de Petri Compactas

Uma rede elementar descreve o comportamento de um número fixo de condições e eventos. A partir das redes elementares, podemos chegar ao conceito de rede compacta, que permite a modelagem de um número variável de condições e eventos. Por essa razão, redes compactas prestam-se melhor à modelagem de sistemas de informação. Nas redes compactas:

- cada lugar define um conjunto de marcas pertencentes ao domínio do lugar e cada marca definida modela uma condição do sistema;
- cada conexão define um conjunto de alterações e cada alteração definida modela um evento.

As propriedades estáticas e dinâmicas do sistema são modeladas empregando-se anotações formais em lugares, conexões e ramos. Analogamente à apresentação das redes elementares, são apresentadas a seguir algumas definições que caracterizam uma Rede de Petri Compacta, bem como a relação com sua rede elementar subjacente.

Definição 10 (rede compacta)

Uma rede compacta é composta por:

1. a definição de uma *linguagem de anotação* (LA), com sintaxe bem semelhante à da lógica de predicados de primeira ordem [2];
2. a definição de um *conjunto de Universos de Discursos (UDs)*, que pode ser feita através da enumeração das entidades e correspondências do UD ou através de fórmulas da LA que especifiquem todos os UDs aceitáveis para a rede;
3. *anotações* associadas a *lugares*, *ramos* e *conexões*, efetuadas em termos da LA, da seguinte forma:
   - a cada conexão é associada uma fórmula da LA. Essa fórmula de conexão aparece, na representação gráfica da rede, dentro ou próxima do retângulo representativo da conexão. No caso especial da fórmula ter sempre o valor verdadeiro, ela pode ser omitida;
   - a cada lugar são associados dois termos constantes (termos sem variáveis livres) da LA, que devem designar, cada qual, um conjunto do UD. Um dos termos, o *domínio do lugar*, aparece, na representação gráfica, entre parênteses, após o identificador do lugar. O outro, o designador da *marcação inicial do lugar*, aparece dentro da circunferência representativa do lugar. Quando a marcação inicial for o conjunto vazio, este termo pode ser omitido;
   - a cada ramo é associado um termo. Esse termo de ramo deve designar um subconjunto do domínio do lugar ao qual esteja ligado e aparece, na representação gráfica, junto à seta representativa do ramo e próximo da conexão à qual esteja ligado.

Definição 11 (lugares da rede elementar subjacente)

Seja uma rede compacta e um UD definido para essa rede. Um lugar da rede compacta define um lugar da rede elementar subjacente para cada entidade do domínio do lugar da rede compacta. Assim sendo, é usual identificar cada lugar da rede elementar subjacente através do par (identificador do lugar da rede compacta, entidade do domínio da rede compacta).

Definição 12 (variáveis de uma conexão)

Seja uma conexão de uma rede compacta. O conjunto de variáveis da conexão é o conjunto de todas as variáveis livres que aparecem nos ramos da conexão.

Por se estar falando de variáveis e não de ocorrências de variáveis, não interessa quantas vezes a variável aparece nos ramos da conexão. Além disso, não é feita nenhuma diferença entre variáveis em ramos de saída e variáveis em ramos de entrada.

Definição 13 (conexões da rede elementar subjacente)

Seja uma rede compacta e um UD definido para essa rede. Uma conexão da rede compacta define uma conexão da rede elementar subjacente para cada valoração das variáveis livres da conexão compacta que obedeça aos seguintes requisitos:

- o conjunto de entidades designado pelo termo de cada ramo, sob a valoração em questão, deve fazer parte do domínio do lugar conectado;
- a fórmula da conexão, caso presente, deve resultar, sob a valoração em questão, no valor verdadeiro.

---

[2] Vide apêndice para uma definição dessa LA.

Essa conexão da rede elementar subjacente liga os lugares subjacentes definidos pelos termos dos ramos da conexão compacta sob a valoração em questão. A conexão subjacente é identificada por uma lista encabeçada pelo identificador da conexão compacta, seguido dos valores das variáveis livres da conexão compacta em ordem alfabética de nome de variável.

Definição 14 (marcas definidas por um lugar da rede compacta)

Seja uma rede compacta e um UD definido para essa rede. Um lugar da rede compacta define uma *marca* para cada entidade do domínio do lugar. É usual identificar a marca através do par (identificador do lugar, entidade do domínio).

Definição 15 (marcação)

O par formado pelos conjuntos que definem as marcas presentes e as marcas ausentes de uma rede compacta define a *marcação da rede*. Analogamente, o par formado pelos conjuntos que definem as marcas presentes a as marcas ausentes de um lugar compacto define a *marcação do lugar*.
Note-se que, por essa definição, qualquer par de conjuntos de marcas presentes e ausentes pode definir uma marcação, mesmo que esta não faça sentido frente à realidade modelada.

Definição 16 (alterações definidas por uma conexão da rede compacta)

Seja uma rede compacta e um UD definido para essa rede. Uma conexão da rede define uma alteração para cada valoração de suas variáveis que obedeça aos seguintes requisitos:

- o conjunto de entidades designado pelo termo de cada ramo, sob a valoração em questão, deve fazer parte do domínio do lugar conectado;
- a fórmula da conexão, caso presente, deve resultar, sob a valoração em questão, no valor verdadeiro.

Cada alteração é identificada por uma lista encabeçada pelo identificador da conexão compacta, seguido dos valores das variáveis da conexão em ordem alfabética de nome da variável.

Definição 17 (marcas ligadas a uma alteração)

As *marcas ligadas a uma alteração* são as marcas correspondentes às entidades designadas pelos ramos da conexão que define a alteração, sob a valoração que estabelece essa definição. Essas marcas podem ser classificadas em:

- marcas de entrada — marcas correspondentes às entidades designadas pelos termos dos ramos de entrada da conexão —;
- marcas de saída — marcas correspondentes às entidades designadas pelos termos dos ramos de saída da conexão —.

E em:

- marcas alteradas — marcas correspondentes às entidades designadas pelos termos associados aos ramos alteradores da conexão —;
- marcas restauradas — marcas correspondentes às entidades designadas pelos termos associados aos ramos restauradores da conexão —;
- marcas mantidas — marcas correspondentes às entidades designadas pelos termos associados aos ramos mantenedores da conexão —.

Analogamente à classificação dos ramos de uma conexão, podem ser feitas combinações desses dois sistemas de classificação das marcas de uma alteração (mantida de entrada, mantida de saída, restaurada de entrada, restaurada de saída, alterada de entrada, alterada de saída).

Definição 18 (regra de habilitação)

Uma alteração está *habilitada* frente a uma marcação quando:

- suas marcas de entrada, caso a alteração as tenha, estiverem presentes na marcação;
- suas marcas de saída, caso a alteração as tenha, estiverem ausentes na marcação.

Definição 19 (independência)

Diz-se que duas alterações habilitadas são *independentes* quando não possuírem:

- marca alterada de entrada comum;
- marca restaurada de entrada comum;
- marca alterada de saída comum;
- marca restaurada de saída comum.

Definição 20 (Conflito)

Diz-se que duas alterações habilitadas estão *em conflito* quando possuírem:

- marca alterada de entrada comum;
- marca restaurada de entrada comum;
- marca alterada de saída comum;
- marca restaurada de saída comum.

Definição 21 (passo)

Um conjunto de alterações é um passo frente a uma marcação, chamada marcação precursora, quando:

- todas as alterações do passo estiverem habilitadas dentro da marcação precursora;
- as alterações do passo não forem conflitantes entre si.

Definição 22 (efeito da ocorrência de um passo)

O efeito da ocorrência de um passo é a transição da marcação precursora para uma marcação sucessora de tal forma que:

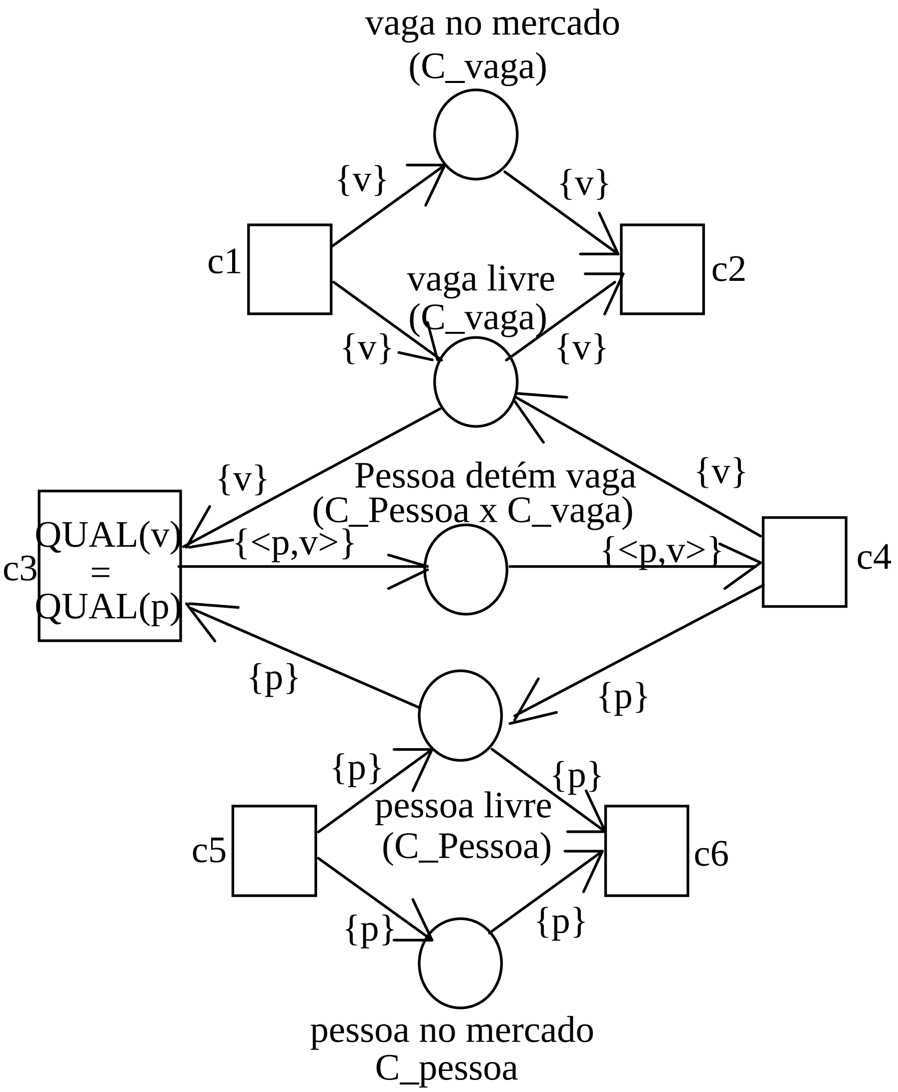
- todas as marcas alteradas de entrada das alterações do passo desapareçam da marcação sucessora;
- todas as marcas alteradas de saída das alterações do passo apareçam na marcação sucessora.

Definição 23 (marcação alcançável)

Frente a uma marcação inicial da rede, define-se marcação alcançável como sendo qualquer marcação obtida pelo efeito da ocorrência de um passo sobre a marcação inicial ou sobre marcações alcançáveis a partir desta.

Para melhor visualização das redes compactas, é apresentado na figura 14 a Rede de Petri Compacta que modela o sistema de um Mercado de Trabalho. A parte de sua rede elementar subjacente relacionada à conexão c3 referente a dois UDs distintos está apresentada nas duas figuras 15 e 16 que seguem a que ilustra o modelo.

Mercado de Trabalho



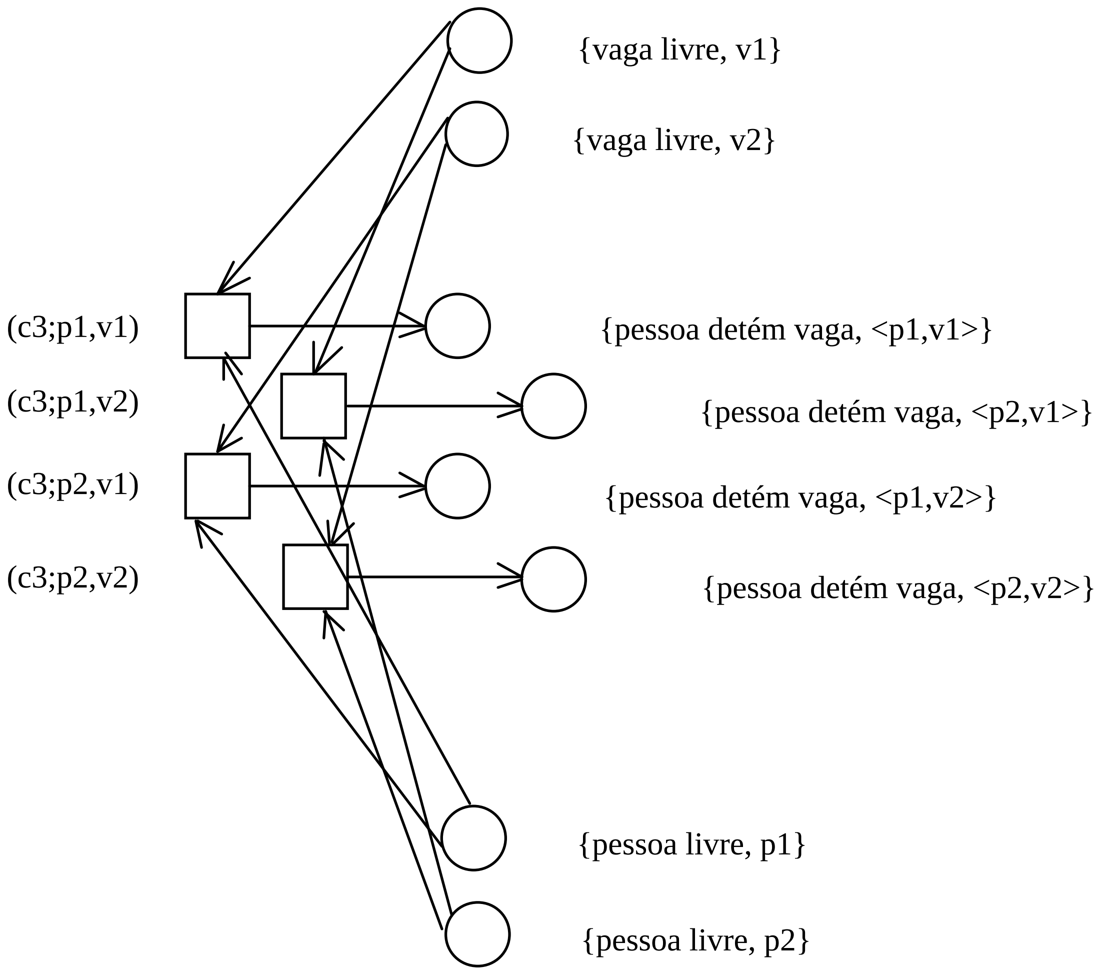
Definição dos Uds da LA

Função : QUAL (unário) $\leftrightarrow \forall$ x, y (y = QUAL (x) $\rightarrow$
(y $\in$ C_Qual $\wedge$ (x $\in$ C_Vaga $\vee$ x $\in$ C_Pessoa)))

Conjuntos de entidades : C_Qual, C_Vaga e C_Pessoa

figura 14 - Rede de Petri Compacta que modela o sistema de um mercado de trabalho

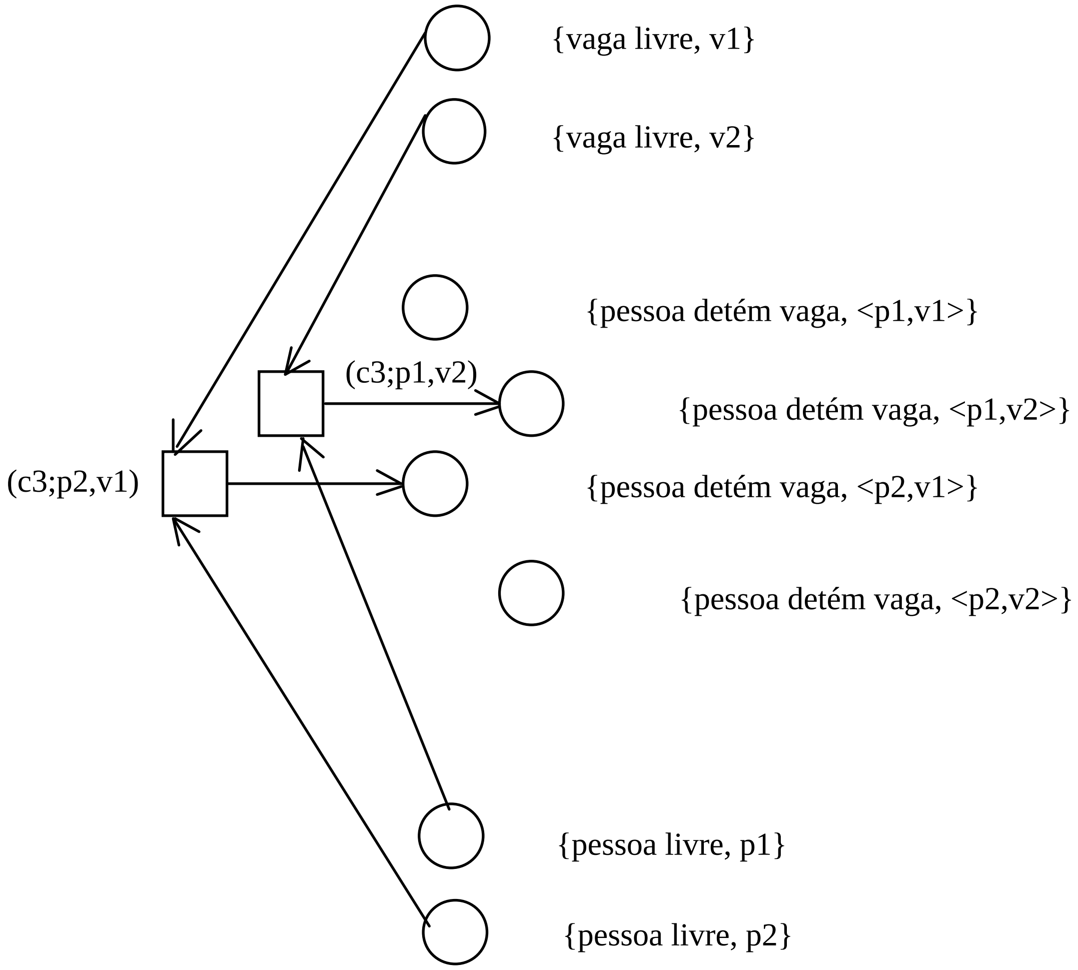
rede subjacente relacionada à conexão c3



Associada à rede compacta correspondente ao UD definido por:
C_QUAL = {q1} ; C_VAGA = {v1, v2} ; QUAL (v1) = QUAL(v2) = q1 ;
C_PESSOA = {p1, p2} ; QUAL (p1) = QUAL (p2) = q1

figura 15 - Rede de Petri Elementar subjacente relacionada a conexão c3 do modelo apresentado na figura 14
(a cada vaga do conjunto C_VAGA, relacionam-se as pessoas do conjunto C_PESSOA)

rede subjacente relacionada à conexão c3



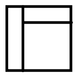
Associada à rede compacta correspondente ao UD definido por:
C_QUAL = {q1, q2} ;   C_VAGA = {v1, v2} ;   QUAL (v1) = q1 ;   QUAL(v2) = q2 ;
C_PESSOA = {p1, p2 ;   QUAL (p1) = q2 ;   QUAL (p2) = q1

figura 16 - Rede de Petri Elementar subjacente relacionada a conexão c3 do modelo apresentado na figura 14
(a cada vaga do conjunto C_VAGA, relaciona-se uma pessoa do conjunto C_PESSOA)

Conexões Mortas

Na perspectiva adotada até o momento, evidenciou-se a possibilidade de especificar propriedades dinâmicas da rede. Mas, é também possível especificar explicitamente propriedades estáticas apenas implicitamente representadas na perspectiva dinâmica. Para isso, necessita-se de um tipo de conexão chamado **conexão morta.** Esse tipo de conexão permite definir alterações que jamais estarão habilitadas na rede. Por não serem utilizadas neste trabalho, será indicado apenas o procedimento para incorporá-las a uma rede de Petri:

- identifica-se o conjunto de todas as marcações que violam a propriedade estática que se deseja expressar, ou seja, as marcações que se deseja excluir;
- incorpora-se à rede conexões que estejam habilitadas somente nas marcações a excluir, identificadas no item anterior;
- marca-se essa conexão como morta. Essa marcação é feita acrescentando, ao espaço interno do retângulo representativo da conexão, duas linhas perpendiculares obtendo o seguinte formato:



notação para uma conexão morta

Tratando conjuntos variáveis de entidades associadas a um lugar

Neste trabalho, necessita-se fazer referências a conjuntos variáveis de entidades que estão presentes ou ausentes em um determinado lugar da rede. Esses conjuntos podem conter entidades que satisfazem critérios específicos. Esse tipo de referência não pode ser obtido empregando a LA, pois essa linguagem só permite expressar conjuntos fixos, ou seja, conjuntos que não sofrem alterações durante a vida do sistema. Para ilustrar a solução deste problema, considera-se o exemplo apresentado a seguir.

EXEMPLO MÁQUINA DEFEITUOSA

Deseja-se uma conexão que defina alterações cuja ocorrência faça com que desapareçam, coincidentemente, todas as entidades (máquinas) presentes no lugar *Máquina Defeituosa*, como ilustra a Rede de Petri Compacta da figura 17.

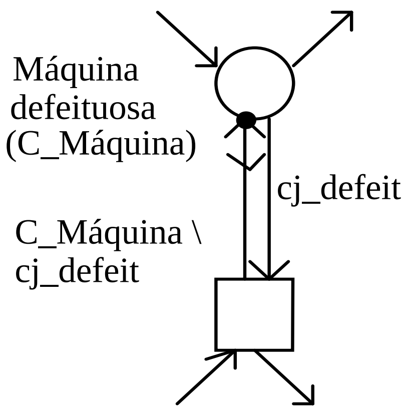


figura 17 - Rede de Petri Compacta que ilustra o exemplo *máquina defeituosa*

O conjunto de máquinas cj_defeit define as marcas de entrada da conexão e seu conjunto complemento em relação ao universo de máquinas C_Máquina define as marcas de saída. Supondo o conjunto C_Máquina = {'m1', 'm2', 'm3'}, pode ser verificado através da tabela 3 que a rede proposta constitui solução para o problema do exemplo. Nessa tabela são apresentados os valores dos termos do ramo de entrada e do ramo de saída da conexão, para cada alteração definida.

cj_defeit	C_Máquina \ cj_defeit
{ }	{'m1', 'm2', 'm3'}
{'m1'}	{'m2', 'm3'}
{'m2'}	{'m1', 'm3'}
{'m3'}	{'m1', 'm2'}
{'m1', 'm2'}	{'m3'}
{'m1', 'm3'}	{'m2'}
{'m2', 'm3'}	{'m1'}
{'m1', 'm2', 'm3''}	{ }

tabela 3 - Valores dos termos do ramo de entrada e do ramo de saída do exemplo *máquina defeituosa*

Observando a tabela, conclui-se que a única alteração habilitada é aquela em que cj_defeit é o conjunto de todas as máquinas **presentes** no lugar **Máquina Defeituosa**, pois somente para esta alteração verificam-se as condições de habilitação:

- todas as marcas de entrada presentes;
- todas as marcas de saída ausentes.

Como esse tipo de problema ocorre com freqüência, utiliza-se uma notação genérica abreviada para a solução. Essa notação, instanciada para este exemplo, está ilustrada na figura 18.

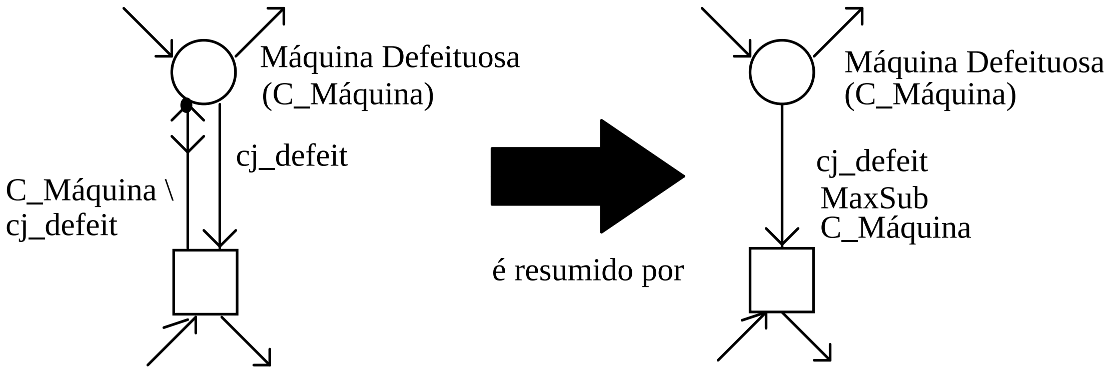


figura 18 - Notação genérica para a Rede de Petri Compacta relativa ao exemplo *máquina defeituosa*

A figura 19 apresenta a notação de forma genérica.

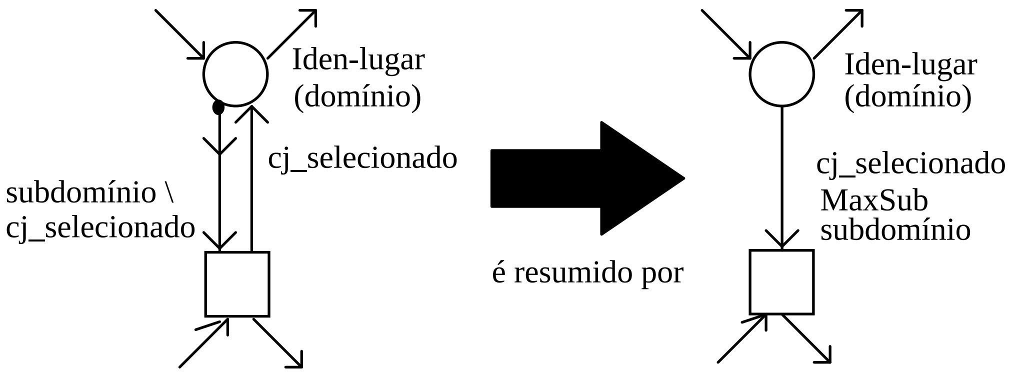


figura 19 - Notação genérica para uma Rede de Petri Compacta que trata um conjunto variável de entidades de um lugar

Essa notação utiliza um novo tipo de ramo, chamado **ramo de entrada de alteração total,** resultante da fusão dos dois existentes, rotulado por uma fórmula contendo dois termos:

- *cj_selecionado*, designando o conjunto máximo de entidades que, estando efetivamente presentes no lugar considerado, estão contidas no conjunto subdomínio. No exemplo, esse conjunto corresponde a cj_defeit;
- *subdomínio*, designando o subconjunto de entidades pertencentes ao domínio do lugar que satisfazem critérios específicos restringindo a quantidade que se deseja referenciar. No exemplo, esse subdomínio corresponde a C_Máquina e coincide com o domínio do lugar. Esse subdomínio poderia, em outro exemplo, restringir-se ao conjunto de máquinas defeituosas da cor verde.

De forma análoga à dessa notação, definida para ramos de entrada e visando à **exclusão** de entidades presentes em um lugar, pode-se introduzir uma notação para:

- **inclusão** de entidades, ausentes em um lugar, pertencentes a um dado conjunto selecionado;
- **verificação** da presença/ausência em um lugar de todas as entidades pertencentes a um dado conjunto selecionado, sem, no entanto, alterar o conteúdo do lugar.

A tabela 4, apresentada a seguir, define essas notações.

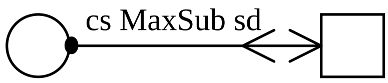
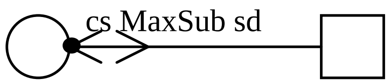
Nome e Propósito do Ramo	**Representação**	**valor do conjunto selecionado (cs)**
Ramo de Entrada de Alteração Total: para excluir todos os elementos do sd (subdomínio) que estão presentes no lugar		conjunto de todos os elementos do sd (subdomínio) que estão presentes no lugar
Ramo de Saída de Alteração Total: para incluir todos os elementos do sd (subdomínio) que estão ausentes no lugar		conjunto de todos os elementos do sd (subdomínio) que estão ausentes no lugar
Ramo de Entrada de Manutenção Total: para verificar a presença de todos os elementos do sd (subdomínio) que estão presentes no lugar		conjunto de todos os elementos do sd (subdomínio) que estão presentes no lugar
Ramo de Saída de Manutenção Total: para verificar a ausência de todos os elementos do sd (subdomínio) que estão presentes no lugar		conjunto de todos os elementos do sd (subdomínio) que estão ausentes no lugar

tabela 4 - Representação dos ramos de uma Rede de Petri Compacta que trata um conjunto variável de entidades de um lugar

3 ESTRUTURAS CONCEITUAIS BÁSICAS

Nesta seção são apresentados procedimentos para especificar graficamente, através de uma Rede de Petri Compacta, cada atividade operacional primitiva contida em um esquema de atividades destinado a modelar o comportamento de um sistema. São tratadas algumas **estruturas conceituais básicas** (**conceptual patterns**) típicas, representativas das possíveis formas sob as quais atividades operacionais primitivas podem apresentar-se. É construída, para cada uma dessas estruturas, a Rede de Petri Compacta que especifica declarativamente a atividade considerada.

A estrutura conceitual básica (ECB) aqui considerada está inserida no **arcabouço conceitual** (**conceptual framework**) definido pelo **Modelo da Essência** e caracteriza um mecanismo de estímulo-resposta primitivo presente no comportamento de sistema modelado. Esse mecanismo estará sendo modelado como membro de um conjunto de mecanismos de mesmo tipo, representado pelo **esquema de atividades** do Modelo da Essência. Para isso, usa-se um diagrama contendo apenas uma atividade operacional primitiva e cada ECB modela a relação matemática, que associa uma resposta (saída) à entrada correspondente, estabelecida como resultado da execução da atividade. Em cada ECB, a entrada da atividade conterá, sempre e apenas, um estímulo (modelado por um fluxo de dados) que dispara sua execução.

Entradas possíveis são:

- fluxo de dados (correspondendo ao estímulo)
- acesso de entrada/leitura (consulta a conteúdo) a depósito de dados
- fluxo de dados + acesso de entrada/leitura (consulta a conteúdo) a depósito de dados

Saídas (respostas) possíveis são:

- fluxo de dados
- acesso de saída/escrita (edição: eliminação, inserção e alteração ou conteúdo) a depósito de dados
- fluxo de dados + acesso de saída/escrita (edição: eliminação, inserção e alteração ou conteúdo) a depósito de dados

Utilizando os conceitos apresentados relativos a Redes de Petri Compactas, pode-se especificar declarativamente, ainda em termos gráficos, as ECBs contidas em um Esquema de Atividades. A figura 20 ilustra genericamente uma estrutura conceitual típica, apresentando os elementos funcionais passivos (fluxo de dados de entrada/saída e depósito de dados submetido a acesso de entrada/saída) com os quais a atividade operacional primitiva de uma ECB pode interagir.

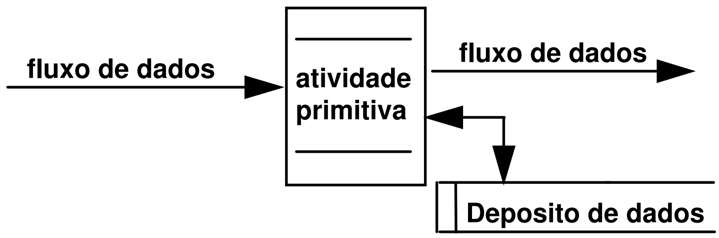


figura 20 - Estrutura conceitual básica típica de um Esquema de Atividades

São apresentados a seguir algumas das estruturas conceituais básicas (ECBs) mais relevantes.

ECB I — fluxo de entrada (estímulo) $\rightarrow$ fluxo de saída (resposta)

Apresenta a atividade operacional primitiva mais elementar, envolvendo apenas um fluxo de dados de entrada e um fluxo de dados de saída [figura 21].

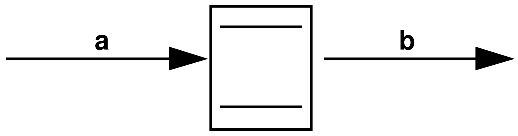


figura 21 - Diagrama de atividade contendo um fluxo de dados de entrada e um fluxo de dados de saída

Para construir uma Rede de Petri que especifique a atividade deve-se:

1. Identificar os componentes de **a** e **b**, os quais serão modelados por elementos que definem os domínios dos lugares da rede que modelam **a** e **b.** Se o fluxo **a** não possuir estrutura, o domínio do lugar que o modela será um conjunto constante C_a especificando todos os valores que podem constituir instâncias de **a**. Se o fluxo **a** possuir estrutura, $\mathbf{a_1 + a_2 + ... + a_n}$, o domínio do lugar que o modela será um conjunto constante C_a_1 x C_a_2 x ... x C_a_n especificando todos os valores que podem constituir instâncias da n-tupla $<\mathbf{a_1, a_2, ... , a_n}>$, ou seja, de **a**. No caso particular de um fluxo sem estrutura, n = 1. Cada lugar receberá um nome que o identifica. Esse nome corresponderá ao descritor do fluxo e definirá um símbolo "predicativo" de aridade n. Os valores da n-tupla que resultam no valor verdadeiro para o predicado definem a marcação do lugar. A anotação de qualquer ramo associado ao lugar designará, de acordo com a teoria de Redes de Petri Compactas, um subconjunto do domínio.

2. Identificar a fórmula associada a cada conexão da rede. As conexões da rede deverão permitir a especificação de todos os possíveis valores de saída produzidos pela atividade, resultantes do processamento de cada instância possível da entrada, evidenciando-se a relação existente entre os valores de saída e a correspondente instância da entrada.

No caso de fluxo sem estrutura, a Rede de Petri Compacta correspondente ao diagrama de atividade acima seria [figura 22]:

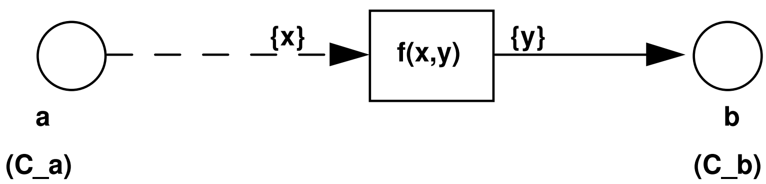


figura 22 - Rede de Petri Compacta correspondente ao diagrama de atividade ilustrado na figura 21

O ramo determinante acima especificado é necessário porque, naquele diagrama de atividade, quem determina a ativação da atividade é a ocorrência do fluxo **a.** Isto é, a rede deve modelar o fato de que, ao se tornar disponível, a entrada do tipo fluxo, a atividade deve imediatamente iniciar sua execução. As variáveis x e y, que rotulam os ramos na rede, designam, respectivamente, elementos dos conjuntos C_a e C_b.

ECB II — fluxo de entrada (estímulo) → fluxo de saída (resposta) composto

Neste caso, associam-se à entrada **a** duas saídas **b** e **c**, como ilustra o diagrama da figura 23.

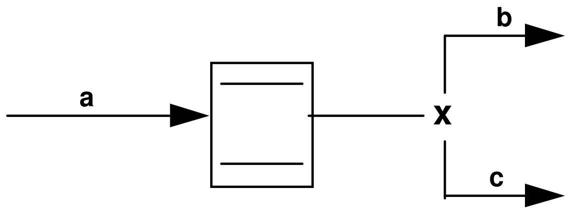


figura 23 - Diagrama de atividade contendo uma entrada associada a duas saídas

Para construir a rede correspondente a esse diagrama, utiliza-se os procedimentos já definidos para o caso I, estabelecendo a presença de três lugares cujos domínios representam **a, b** e **c** [figura 24].

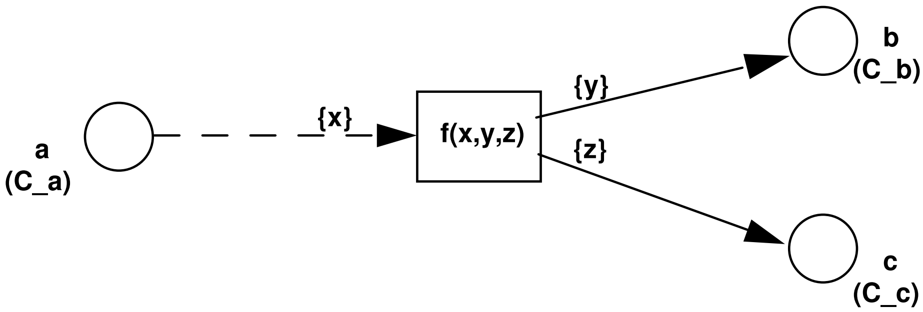


figura 24 - Rede de Petri Compacta correspondente ao diagrama de atividade ilustrado na figura 23

ECB III — fluxo de entrada (estímulo) → fluxo de saída (resposta) alternativo

Neste caso, associa-se à entrada **a** uma saída alternativa, **b** ou **c,** como ilustra o diagrama da figura 25.

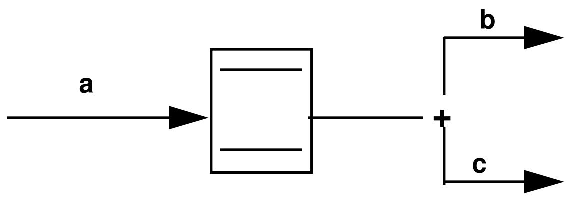


figura 25 - Diagrama de atividade contendo uma entrada associada a duas saídas alternativas

A rede correspondente a esse diagrama, com a possibilidade de alternativa sendo modelada através de uma relação de conflito de alterações, está ilustrada na figura 26.

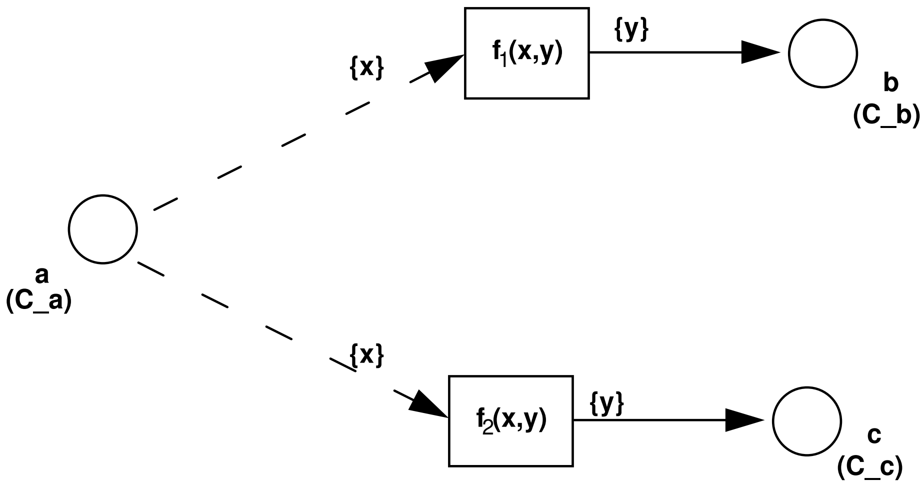


figura 26 - Rede de Petri Compacta correspondente ao diagrama de atividade ilustrado na figura 25

ECB IV — (fluxo de entrada (estímulo) + acesso de entrada a depósito de dados) → fluxo de saída (resposta)

Neste caso, foi acrescentado ao diagrama do caso I um depósito de dados identificado por ENT, ao qual a atividade realiza acesso de entrada (leitura) para efetuar uma consulta. Por consulta, entende-se o uso sem consumo de um elemento de informação presente no depósito [figura 27].

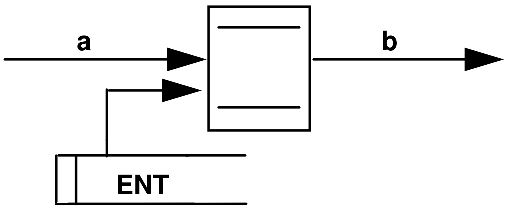


figura 27 - Diagrama de atividade contendo um acesso de entrada (leitura) a um depósito

A Rede de Petri Compacta correspondente a esse diagrama, ilustrada na figura 28, é construída de acordo com o procedimento relativo ao caso I, apenas acrescentando-se um lugar para modelar cada elemento de informação contido no depósito e um ramo restaurador para modelar a consulta. Esse ramo deve ser restaurador quando se desejar inibir consultas simultâneas, por atividades distintas, ao mesmo elemento de informação do depósito. O ramo será mantenedor quando a modelagem de consulta simultânea for desejada.

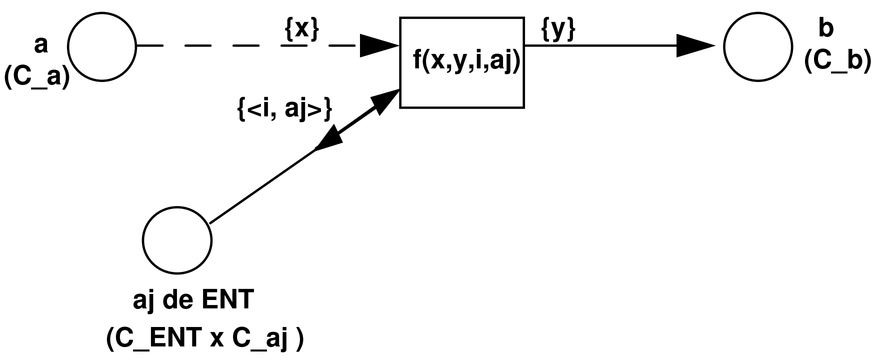


figura 28 - Rede de Petri Compacta correspondente ao diagrama de atividade ilustrado na figura 27, contendo um ramo restaurador de entrada que modela consulta ao valor de um atributo registrado no depósito

De acordo com a notação adotada, o ramo restaurador, rotulado por {<i, a_j>}, modela a consulta ao valor do atributo j do registro i do depósito ENT. Esse registro contém os valores de todos os atributos armazenados no depósito e define uma entidade do Modelo Entidades-Relacionamentos associado ao Esquema de Atividades apresentado no diagrama de atividade da figura 27.

Quando se deseja modelar o fato de que um dado atributo de uma entidade não possui valor registrado no depósito ENT, utiliza-se um ramo restaurador de saída conforme a Rede de Petri Compacta ilustrada na figura 29.

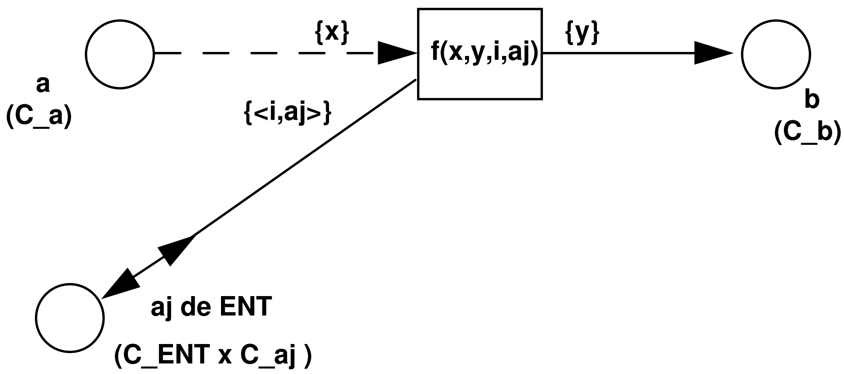


figura 28 - Rede de Petri Compacta correspondente ao diagrama de atividade ilustrado na figura 27, contendo um ramo restaurador de saída que modela consulta a valor de atributo não registrado no depósito

ECB V — fluxo de entrada (estímulo) → acesso de saída a depósito de dados (resposta)

Neste caso, considera-se uma edição efetuada no depósito ENT, ilustrada na figura 30. Por edição entende-se uma operação de INCLUSÃO, de EXCLUSÃO ou de ALTERAÇÃO.

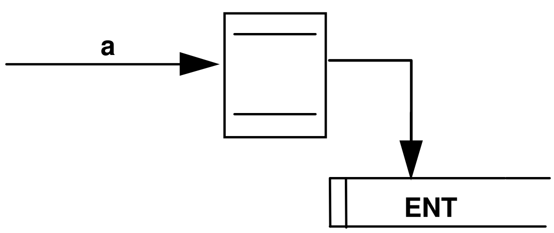


figura 30 - Diagrama de atividade modelando uma operação de edição sobre um depósito

As três operações de edição são modeladas em rede através de ramos específicos ligados aos lugares associados aos elementos definidos pelo depósito ENT. A Rede de Petri Compacta correspondente a esse diagrama é construída de forma semelhante à apresentada no caso IV, acrescentando-se os elementos para a modelagem dos procedimentos de edição no depósito ENT.

Para especificar uma INCLUSÃO de registro em ENT a rede seria [figura 31]:

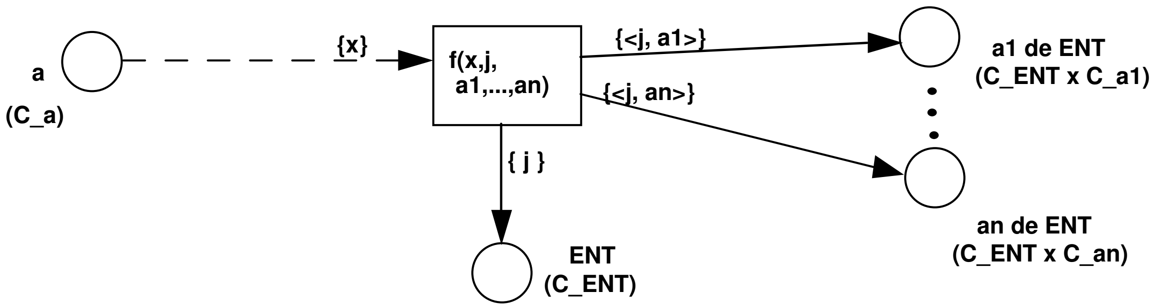


figura 31 - Rede de Petri Compacta correspondente ao diagrama de atividade da figura 30, especificando uma operação de inclusão de registro em depósito

Nessa rede, os ramos alteradores de saída associados aos lugares **ENT**, $\mathbf{a_1}$ **de ENT**, **...**, $\mathbf{a_n}$ **de ENT** modelam a inclusão do registro j em **ENT**.

Para especificar uma EXCLUSÃO de registro, os elementos da Rede de Petri Compacta serão os mesmos que os da rede que modela a inclusão, conforme ilustrado na figura 32. A diferença reside apenas no tipo dos ramos ligados aos lugares que representam conteúdo do depósito ENT, que passa a ser alterador de entrada.

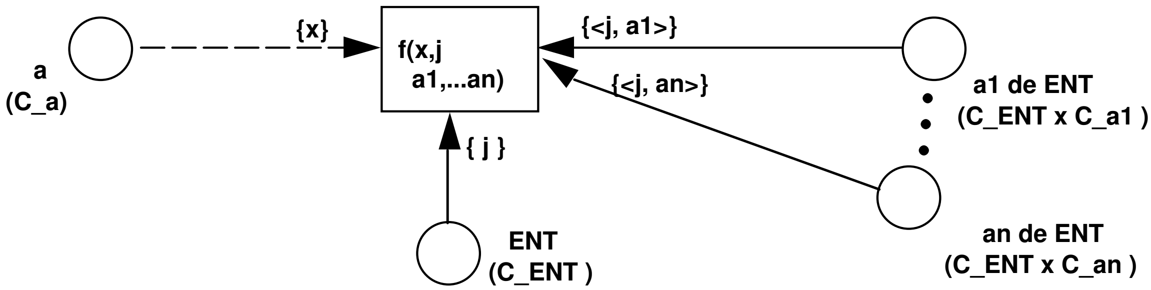


figura 32 - Rede de Petri Compacta correspondente ao diagrama de atividade da figura 30, especificando uma operação de exclusão de registro em depósito

Para especificar uma ALTERAÇÃO no valor de um atributo de um registro, liga-se a conexão ao lugar através de dois ramos: um ramo alterador de entrada, cuja anotação deve designar o valor corrente do atributo a ser alterado, e um ramo alterador de saída, cuja anotação deve designar o novo valor do atributo. A figura 33 ilustra a operação de alteração de valor para apenas um atributo, $\mathbf{a_i}$**,** do registro **j** de ENT, sem no entanto alterar os valores dos demais elementos presentes em ENT.

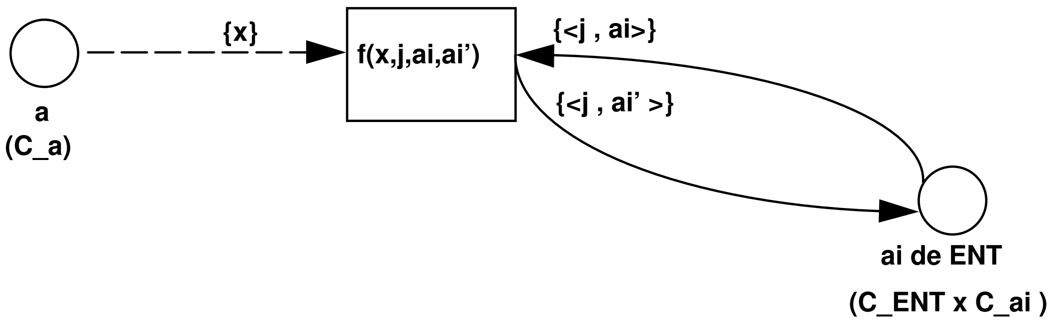


figura 33 - Rede de Petri Compacta correspondente ao diagrama de atividade da figura 30, especificando uma operação de alteração no valor de um atributo de um registro em depósito

ECB VI — (fluxo de entrada (estímulo) + acesso de entrada a depósito de dados) → (fluxo de saída (resposta) + acesso de saída a depósito de dados (resposta))

Neste caso, considera-se a execução de CONSULTA e EDIÇÃO ao depósito ENT, ilustrada na figura 34.

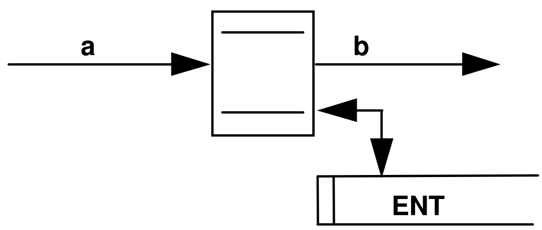


figura 34 - Diagrama de atividade modelando operações de consulta e edição a um depósito

A construção da Rede de Petri correspondente a esse diagrama segue os mesmos procedimentos vistos nos casos anteriores, salientando que, neste caso, haverá a participação simultânea dos ramos restauradores, que especificam o processo de consulta, e dos ramos alteradores, que especificam inclusão, exclusão ou alteração, ligados a lugares associados ao depósito ENT. São apresentados a seguir alguns modelos de Rede de Petri Compacta correspondentes ao diagrama da figura 34, considerando casos de inclusão, alteração e exclusão. O lugar denotado por b será interpretado como uma mensagem de confirmação ou de rejeição.

Considere-se que o diagrama acima modele um processo de **inclusão** de um registro **i** em ENT, onde b modela uma mensagem de confirmação ou de rejeição. Para especificar essa inclusão, a Rede de Petri Compacta ilustrada na figura 35 apresenta-se com duas conexões: a primeira modelando que o registro **i** já pertence ao depósito ENT através de um ramo restaurador de entrada, sendo **b** uma mensagem de rejeição expressa por um ramo alterador de saída anotado por {y}; a segunda, modelando a inclusão do registro **i** em ENT através de ramos alteradores de saída associados aos lugares **ENT**, **a_1 de ENT, ..., a_n de ENT** e a emissão da mensagem de confirmação de inclusão, através do ramo alterador de saída também anotado por {y}. O fato de y representar uma mensagem de confirmação ou rejeição é expresso na fórmula da conexão.

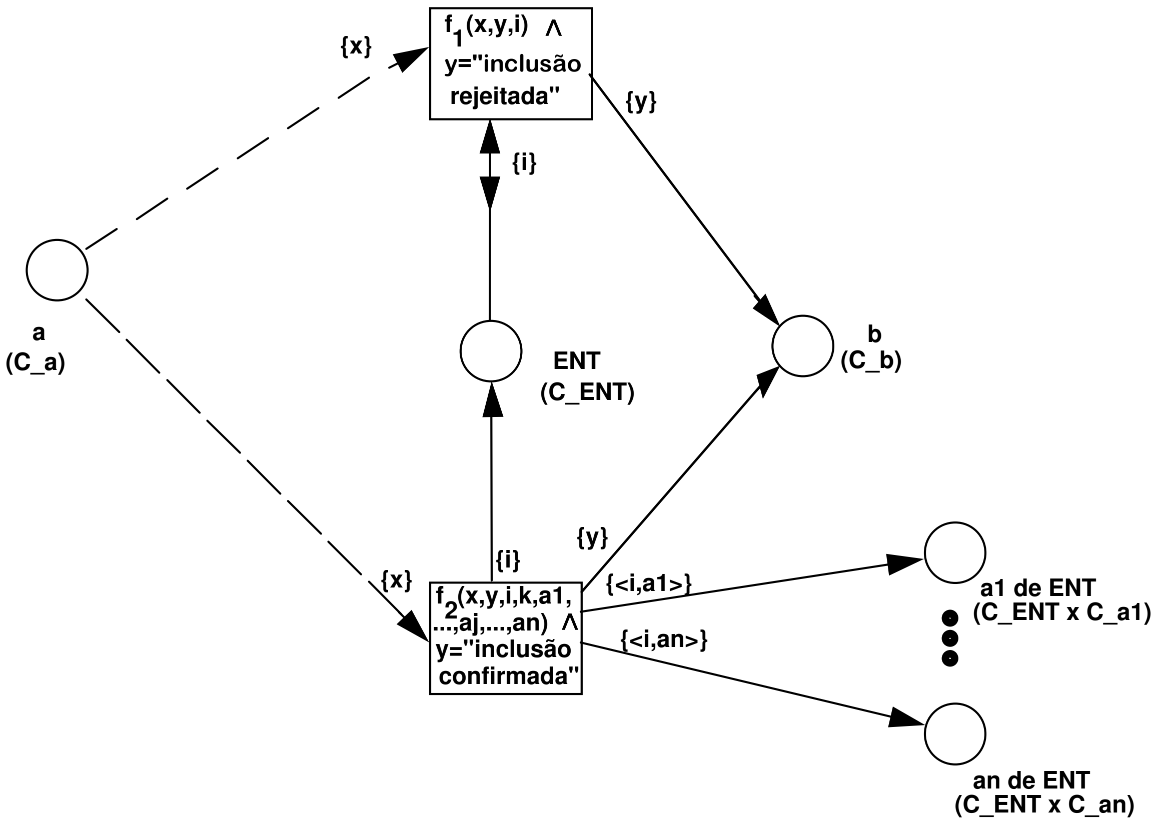


figura 35 - Rede de Petri correspondente ao diagrama de atividades da figura 34, especificando um procedimento de inclusão de um registro em depósito

Caso o diagrama represente um processo de exclusão de um registro **i** de ENT, a Rede de Petri associada também possuirá duas conexões. Uma conexão envolvendo um ramo restaurador de saída, que verifica a ausência do registro **i** em ENT, e um ramo alterador de saída anotado por {y}, que designa uma mensagem de rejeição; e uma conexão contendo ramos alteradores de entrada, associados aos lugares **ENT, a_1 de ENT,..., a_n de ENT**, modelando a exclusão do registro i, além de um ramo alterador de saída, anotado por {y}, designando a mensagem de confirmação da exclusão. A figura 36 ilustra esse especificação.

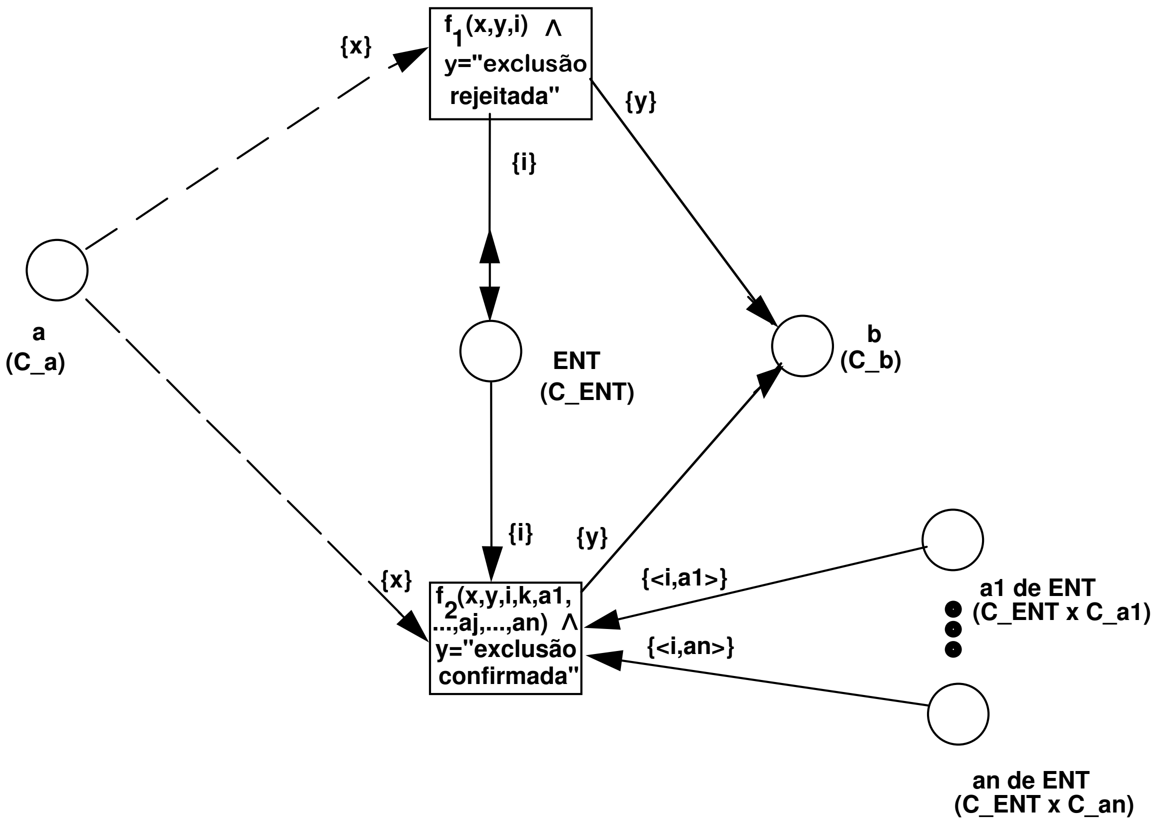


figura 36 - Rede de Petri Compacta correspondente ao diagrama de atividade da figura 34, especificando uma operação de exclusão de um registro em depósito

Para especificar uma **alteração** no valor de um atributo **ak** de um registro **i**, a Rede de Petri Compacta ilustrada na figura 37 possui uma conexão ligada ao lugar **ak de ENT**, que contém o valor do atributo **ak** do registro i que se deseja alterar, através de dois ramos: um ramo alterador de entrada, cuja anotação designa o valor corrente do atributo a ser alterado, e um ramo alterador de saída, cuja anotação designa o novo valor do atributo. Essa rede também possui uma outra conexão ligada a um ramo restaurador de saída associado ao lugar **ENT** que modela a ausência do registro **i** que se deseja alterar; essa condição dará origem a uma mensagem de rejeição. Cada conexão também está ligada ao lugar **b** através de um ramo alterador de saída, anotado por {y}, que designará, respectivamente, uma mensagem de confirmação ou de rejeição, expressa na fórmula da conexão.

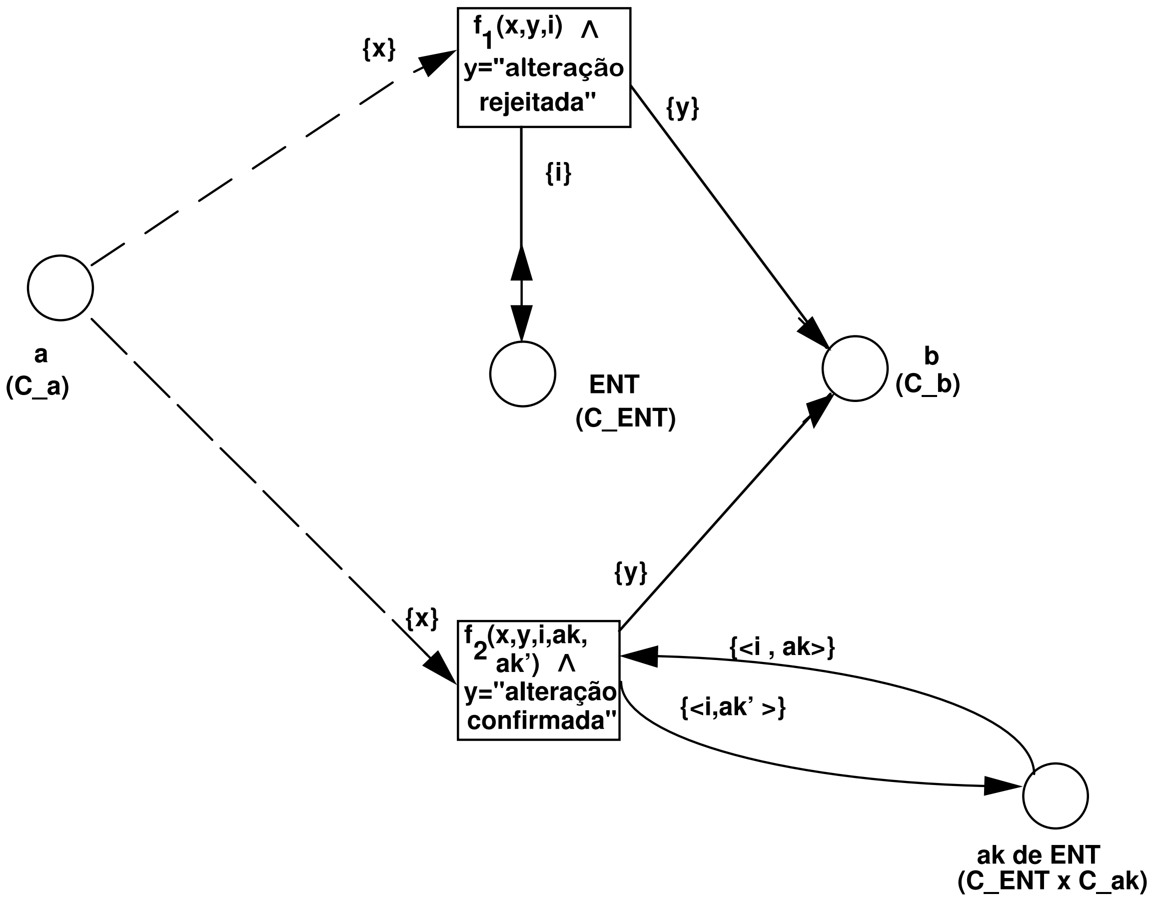


figura 37 - Rede de Petri Compacta correspondente ao diagrama de atividade da figura 34, especificando uma operação de alteração do valor de um atributo de um registro em depósito

Além dos casos apresentados, relativos às estruturas conceituais básicas mais comuns, pode-se especificar através de uma Rede de Petri Compacta qualquer atividade operacional primitiva que execute uma combinação das operações anteriormente consideradas. Para isso, a rede correspondente deve ser construída com base nos casos aqui tratados, considerando-se os elementos do diagrama de atividade que define a combinação desejada.

4 EXEMPLOS DE ESPECIFICAÇÃO DECLARATIVA GRÁFICA UTILIZANDO REDES DE PETRI COMPACTAS

Nesta seção, utilizando as estruturas conceituais básicas tratadas na seção precedente, são apresentados alguns exemplos ilustrativos de como especificar declarativamente uma atividade operacional primitiva utilizando uma Rede de Petri Compacta. Cada exemplo apresenta também a especificação declarativa da atividade baseada em Lista de Pré/Pós-Condições, instrumento textual de modelagem semi-formal freqüentemente empregado.

EXEMPLO 1 — CALCULA RAIZ QUADRADA

A atividade operacional primitiva considerada, dado um número maior que ou igual a zero, efetua o cálculo da raiz quadrada. Se o número for menor que zero, emite mensagem de rejeição [figura 38].

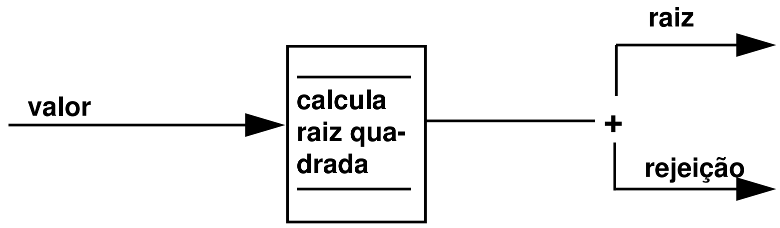


figura 38 - Diagrama de atividade para o exemplo C*alcula Raiz Quadrada*

pré-condição 1: valor $\geq$ 0
pós-condição 1: raiz TALQUE (raiz^2 = valor) $\wedge$ $\neg$ rejeição

pré-condição 2: valor < 0
pós-condição 2: rejeição indicando que não existe raiz real $\wedge$ $\neg$ raiz

A Rede de Petri Compacta correspondente é apresentada na figura 39.

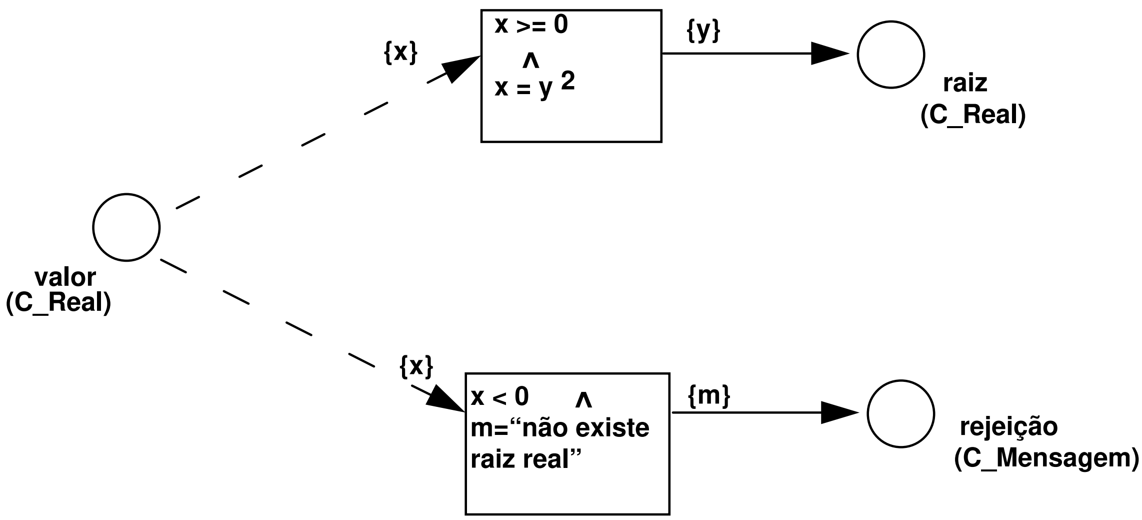


figura 39 - Rede de Petri Compacta correspondente ao exemplo C*alcula Raiz Quadrada*

Neste exemplo, *x*, *y* e *m* representam, respectivamente, elementos dos conjuntos representados pelos lugares *valor*, *raiz* e *rejeição*. Cada alteração só estará habilitada quando a fórmula da conexão que a define for verdadeira. Vale também observar a presença de ramos determinantes nesta rede. Isto significa que, em cada instante em que existir uma marcação habilitando uma alteração, esta ocorrerá obrigatoriamente.

EXEMPLO 2 — EMITE HISTÓRICO ESCOLAR

A atividade operacional primitiva considerada, dado o número de matrícula do aluno, verifica se aluno consta no cadastro e se há disciplinas associadas a ele. Em caso afirmativo, obtém dados do aluno e emite seu histórico. Em caso negativo, emite uma mensagem de rejeição [figura 40].

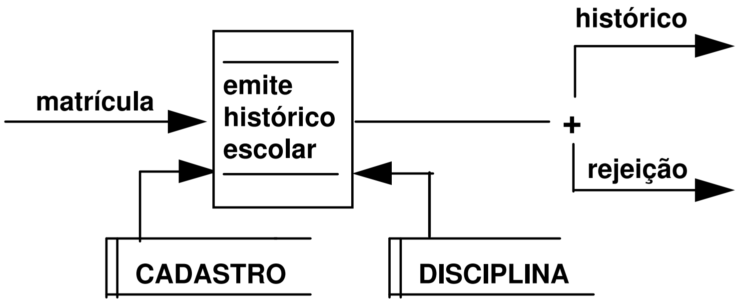


figura 40 - Diagrama de atividade para o exemplo *Emite Histórico Escolar*

pré-condição 1: matrícula $\in$ CADASTRO $\wedge$ matrícula $\in$ DISCIPLINA
pós-condição 1: histórico TALQUE (valor histórico =
valor (CADASTRO (matrícula).nome +
DISCIPLINA (matrícula).{<codmat, nota>})) $\wedge$ $\neg$ rejeição

pré-condição 2: matrícula $\notin$ CADASTRO
pós-condição 2: rejeição indicando que aluno não está cadastrado $\wedge$ $\neg$ histórico

pré-condição 3: matrícula $\in$ CADASTRO $\wedge$ matrícula $\notin$ DISCIPLINA
pós-condição 3: rejeição indicando que o aluno não cursou disciplina alguma $\wedge$
$\neg$ histórico

A Rede de Petri Compacta correspondente é apresentada na figura 41.

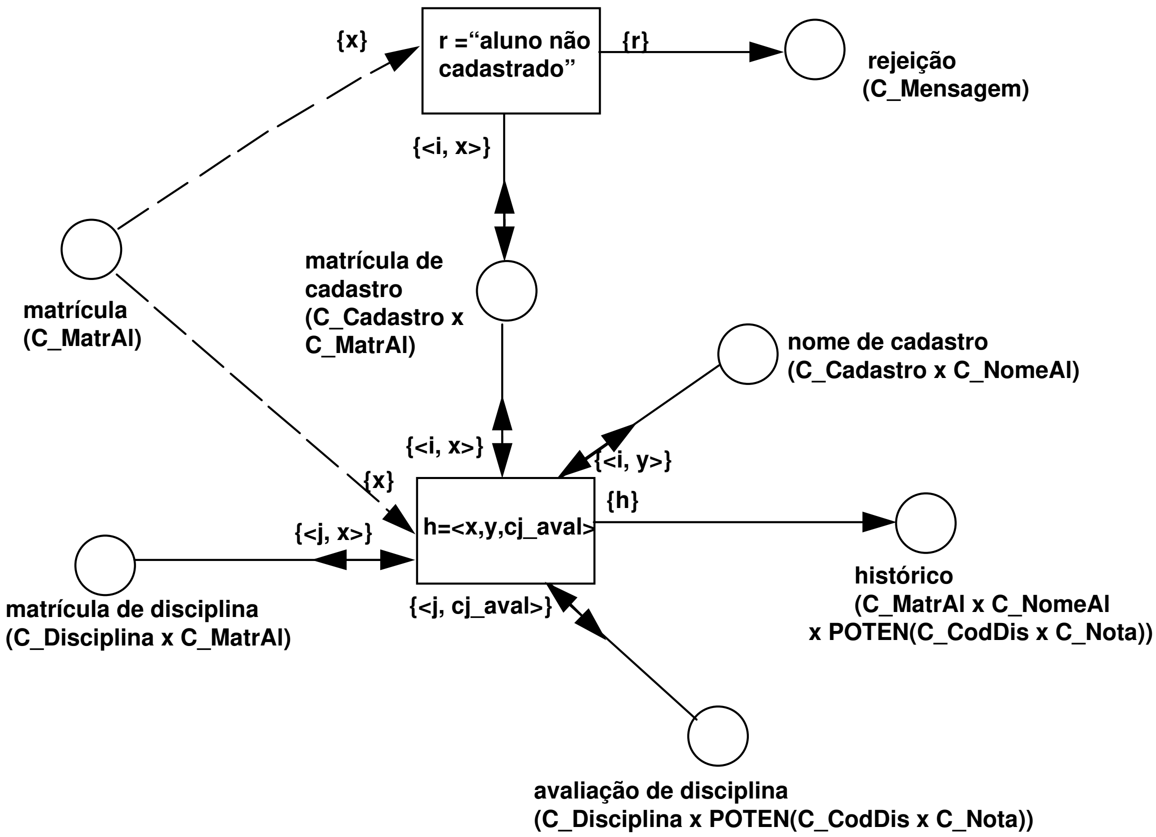


figura 41 - Rede de Petri Compacta correspondente ao exemplo *Emite Histórico Escolar*

O diagrama de atividade do exemplo contém dois depósitos de dados CADASTRO e DISCIPLINA. Conforme indicado anteriormente, cada atributo de registro em depósito corresponde a um lugar na rede cujo domínio é definido por um conjunto de pares que modelam o valor do atributo em cada registro **i** do depósito. Neste exemplo, *cj-aval* designa o conjunto de pares contendo o código da disciplina e a nota do aluno, para cada aluno identificado pelo número de matrícula no depósito de dados DISCIPLINA.

EXEMPLO 3 — ALTERA NOTA DE ALUNO

A atividade operacional primitiva considerada, dado o número de matrícula do aluno, o código da disciplina e a nova nota, verifica se a matrícula do aluno consta no arquivo de disciplinas. Em caso negativo, emite mensagem de rejeição. Em caso positivo, verifica se o código da disciplina está presente no registro correspondente ao aluno. Caso esteja, efetua a alteração da nota e emite mensagem de confirmação. Caso contrário, emite mensagem de rejeição [figura 42].

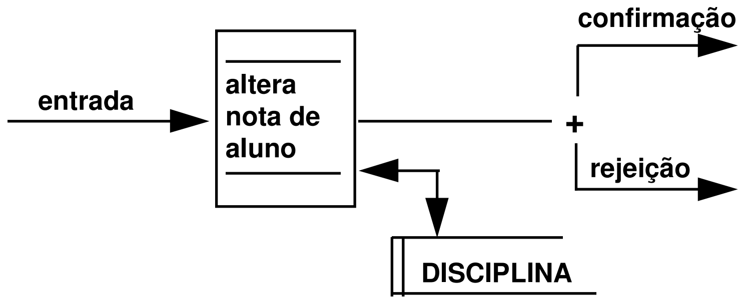


figura 42 - Diagrama de atividade para o exemplo *Altera Nota de Aluno*

pré-condição 1: entrada (matricula $\notin$ DISCIPLINA)
pós-condição 1: mensagem de rejeição indicando que a matrícula do aluno não consta no arquivo de disciplinas

pré-condição 2: entrada (matrícula $\in$ DISCIPLINA) $\wedge$ $\neg\exists$ código da disciplina associado à matrícula, em DISCIPLINA
pós-condição 2: mensagem de rejeição indicando que a disciplina não consta no arquivo de disciplinas

pré-condição 3: entrada (matrícula $\in$ DISCIPLINA) $\wedge$ $\exists$ código da disciplina associado a matrícula, em DISCIPLINA
pós-condição 3: nota alterada, em DISCIPLINA, do aluno cuja matrícula consta na "entrada", associada ao código que consta na "entrada" $\wedge$ mensagem de confirmação indicando que nota foi alterada

A Rede de Petri Compacta apresentada na figura 43 especifica a atividade *altera nota do aluno*. Nela, está apresentada a modelagem da alteração do valor, pertencente a uma n-tupla, de um atributo, que é a nota do aluno.

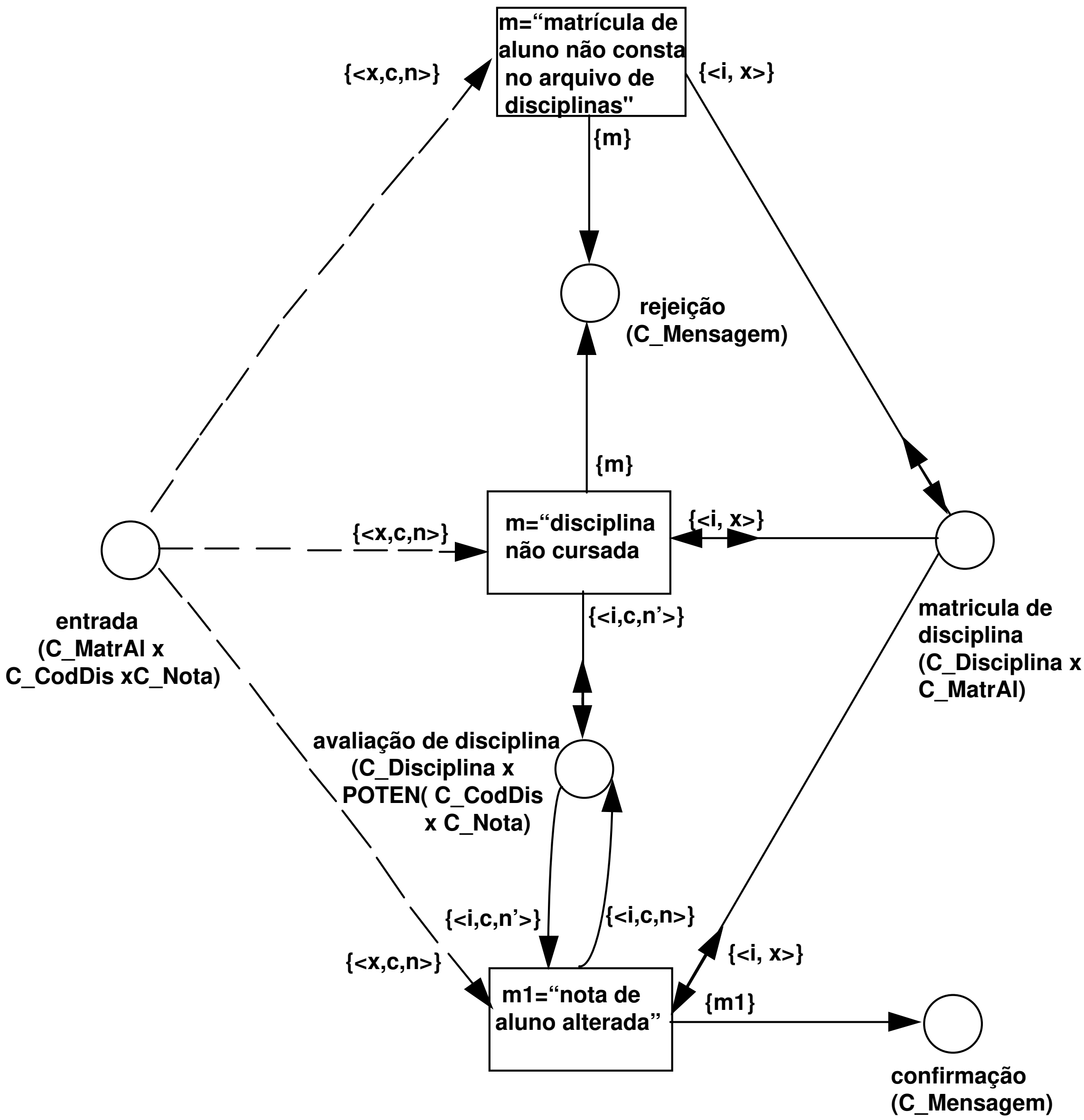


figura 43 - Rede de Petri Compacta correspondente ao exemplo *Altera Nota de Aluno*

EXEMPLO 4 — SEPARA MENSAGEM

A atividade operacional primitiva considerada, dada uma mensagem com valores consistentes, identifica se trata-se de uma mensagem comum ou de uma mensagem especial e separa a respectiva mensagem. Em caso de valores inconsistentes, emite mensagem de rejeição [figura 44].

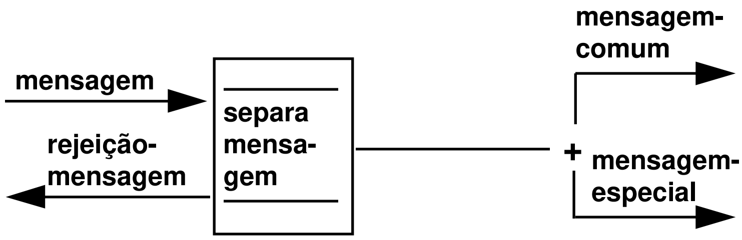


figura 44 - Diagrama de atividade para o exemplo *Separa Mensagem*

pré-condição 1: mensagem (∃ valor para texto ∧ valor para tipo indica mensagem - comum ∧ ¬ ∃ valor para classe)
pós-condição 1: mensagem-comum ∧ ¬ rejeição-mensagem ∧
¬ mensagem-especial

pré-condição 2: mensagem (∃ valor para texto ∧ valor para tipo indica mensagem-especial ∧ ∃ valor para classe)
pós-condição 2: mensagem-especial ∧ ¬ rejeição-mensagem ∧
¬ mensagem-comum

pré-condição 3: mensagem (¬ ∃ valor para (texto ∨ tipo) ∨ (∃ valor para texto ∧ valor para tipo indica mensagem especial ∧ ¬ ∃ valor para classe) ∨ (∃ valor para texto ∧ valor para tipo indica mensagem comum ∧
∃ valor para classe))
pós-condição 3: rejeição-mensagem ∧ ¬ mensagem-comum ∧
¬ mensagem-especial

A Rede de Petri Compacta correspondente é apresentada na figura 45.

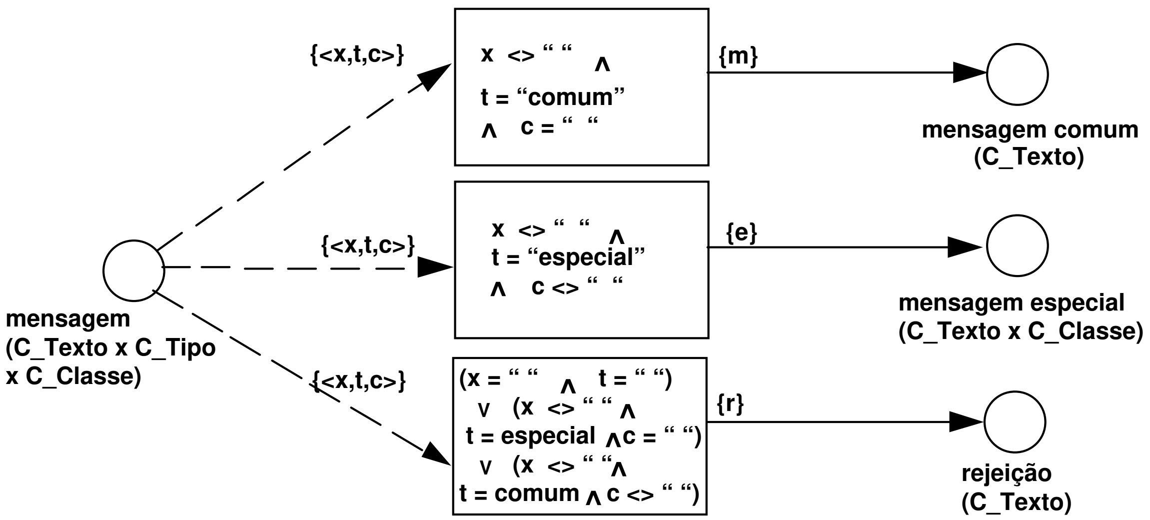


figura 45 - Rede de Petri Compacta correspondente ao exemplo S*epara Mensagem*

EXEMPLO 5 — DECODIFICA MENSAGEM COMUM

A atividade operacional primitiva considerada, dada uma mensagem comum, decodifica essa mensagem [figura 46]. A aplicação do código ao texto da mensagem é especificada através da função DECOD-COMUM.

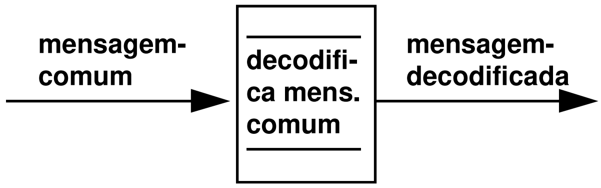


figura 46 - Diagrama de atividade para o exemplo *Decodifica Mensagem Comum*

pré-condição 1: mensagem-comum
pós-condição 1: mensagem-decodificada (mensagem-decodificada.texto =
DECOD-COMUM (mensagem-comum.texto))

A Rede de Petri Compacta correspondente é apresentada na figura 47.

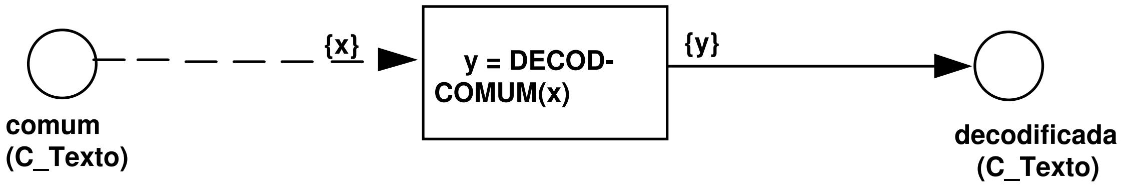


figura 47 - Rede de Petri Compacta correspondente ao exemplo D*ecodifica Mensagem Comum*

EXEMPLO 6 — DECODIFICA MENSAGEM ESPECIAL

A atividade operacional primitiva considerada, dado uma mensagem especial, verifica se existe no depósito DI 2 o código correspondente à classe da mensagem. Em caso positivo, decodifica o texto da mensagem especial, utilizando a função DECOD-ESPECIAL, sem realizar nenhum acesso ao depósito MENSAGEM ESPECIAL. Em caso negativo, inclui a mensagem no depósito MENSAGEM ESPECIAL [figura 48].

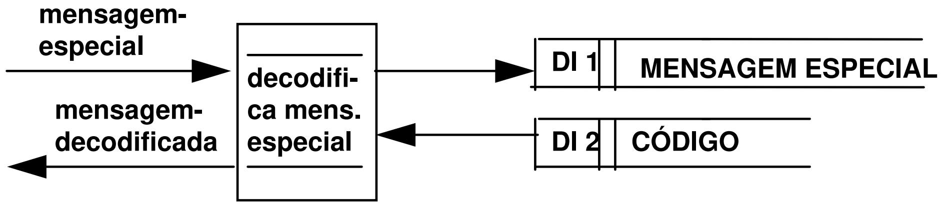


figura 48 - Diagrama de atividade para o exemplo D*ecodifica Mensagem Especial*

pré-condição 1: mensagem-especial $\wedge$ $\exists$i (CÓDIGO (i).classe = mensagem-especial.classe)
pós-condição 1: mensagem-decodificada (mensagem-decodificada.texto = DECOD-ESPECIAL(i) (mensagem-especial.texto)) $\wedge$ $\neg$ acesso a DI 1

pré-condição 2: mensagem-especial $\wedge$ $\neg$ $\exists$i (CÓDIGO(i).classe = mensagem-especial.classe)
pós-condição 2: $\neg$ mensagem-decodificada $\wedge$ $\exists$j (MENSAGEM ESPECIAL(j) = mensagem-especial)

A atividade *decodifica mensagem especial* é especificada pela Rede de Petri Compacta apresentada na figura 49, a qual inclui uma conexão que não é anotada (implicitamente, corresponde a uma anotação pela constante T que designa o valor verdadeiro).

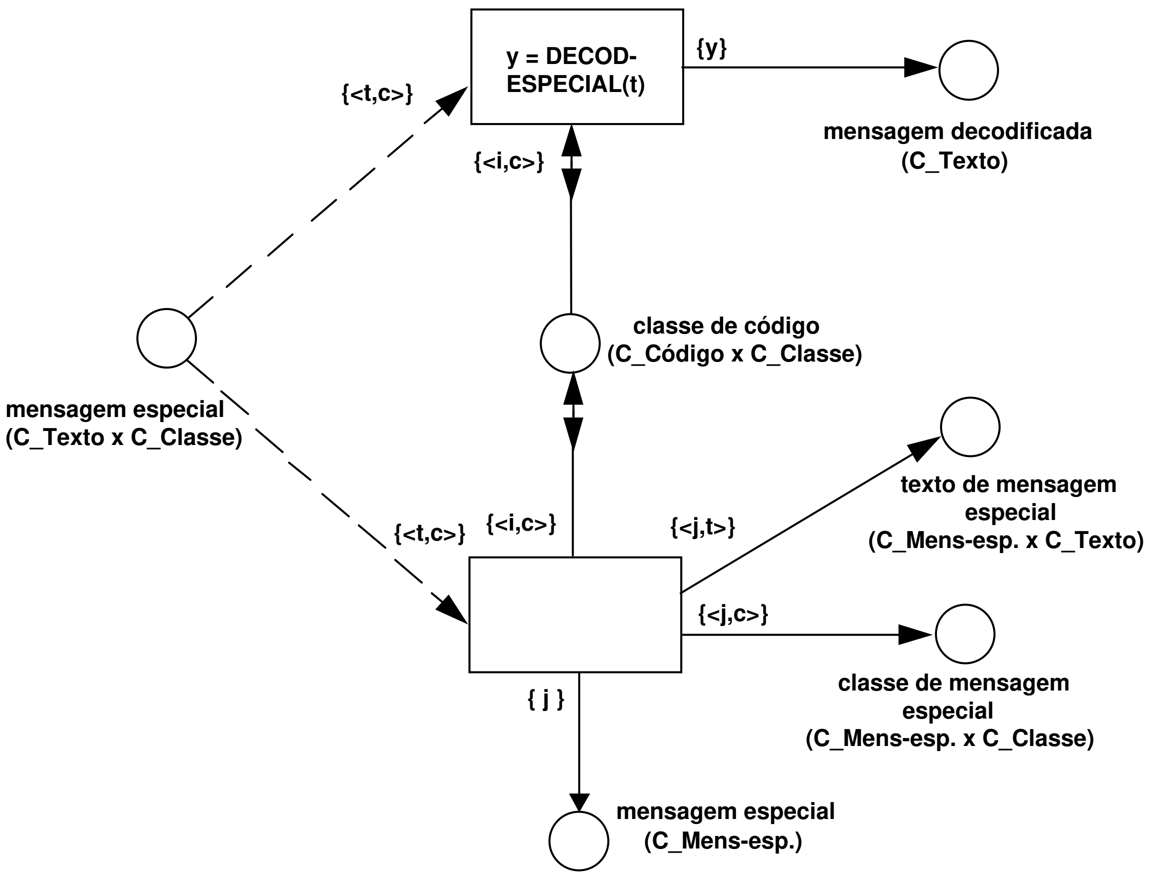


figura 49 - Rede de Petri Compacta correspondente ao exemplo *Decodifica Mensagem Especial*

EXEMPLO 7 — TRATA CÓDIGO

A atividade operacional primitiva considerada, dado o código, com valores atribuídos a todos os seus componentes, verifica em DI 1 a existência de mensagens da mesma classe. Caso existam, decodifica todas aplicando ao texto de cada uma a função DECOD-ESPECIAL e gera a saída mensagem-decodificada, além de armazenar a entrada em DI 2 e excluir de DI 1 cada mensagem tratada. Caso não existam mensagens da classe correspondente ao código, é produzida apenas a resposta de armazenamento da entrada em DI 2. Haverá rejeição caso esteja faltando algum componente da entrada [figura 50].

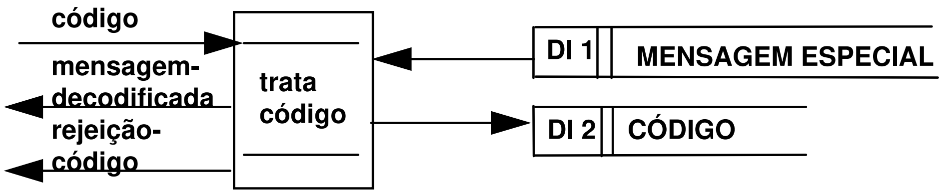


figura 50 - Diagrama de atividade para o exemplo *Trata Código*

pré-condição 1: código ($\exists$ (texto + classe)) $\wedge$ $\exists$i (MENSAGEM ESPECIAL(i).classe = código.classe)
pós-condição 1: mensagem-decodificada (mensagem-decodificada.texto = DECOD-ESPECIAL (mensagem-especial(i).texto)) $\wedge$ $\exists$j (CÓDIGO(j) = código) $\wedge$ $\neg$ $\exists$i (MENSAGEM ESPECIAL(i).classe = código.classe) $\wedge$ $\neg$ rejeição-código

pré-condição 2: código ($\exists$ (texto + classe)) $\wedge$ $\neg$ $\exists$i (MENSAGEM ESPECIAL(i).classe= código.classe)
pós-condição 2: incluído (DI 2, código) $\wedge$ $\neg$ rejeição-código $\wedge$ $\neg$ mensagem-decodificada

pré-condição 3: código ($\neg$ $\exists$ (texto $\vee$ classe))
pós-condição 3: rejeição-código $\wedge$ $\neg$ mensagem-decodificada $\wedge$ $\neg$ acesso-escrita(CÓDIGO)

A Rede de Petri Compacta correspondente é apresentada na figura 51.

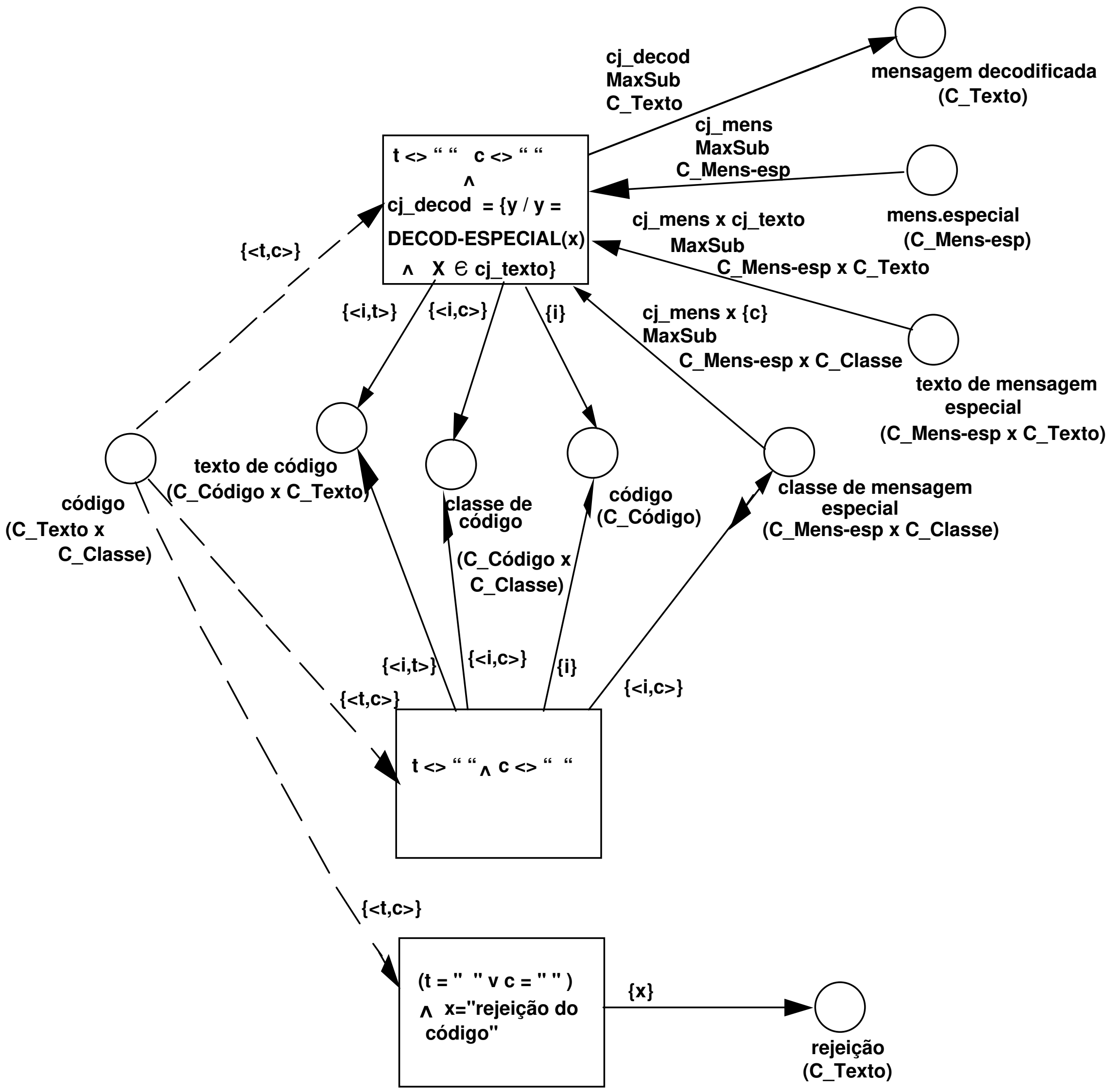


figura 51 - Rede de Petri Compacta correspondente ao exemplo *Trata Código*

Neste exemplo, foi utilizada a notação específica para selecionar, em um lugar da rede, um conjunto de entidades que possuem uma certa característica em comum. O conjunto representado por *cj_mens*, contém todas as mensagens presentes no lugar *mens. especial* que possuem a mesma classe *c* e é designado pelo ramo determinante de entrada anotado por *{<t,c>}*. O conjunto *cj_decod* contém todos os textos das mensagens presentes em cj_mens, que foram decodificados através da função DECOD-ESPECIAL. O conjunto *cj_texto* contém os textos de mensagem especial que são de uma mesma classe selecionadas no conjunto cj_mens.

5 CONCLUSÃO

Neste trabalho, foram apresentados vários procedimentos, cada um correspondendo a estrutura conceitual básica (conceptual pattern) de ocorrência freqüente na modelagem de sistemas baseada no Modelo da Essência, visando uma especificação declarativa em termos de Redes de Petri Compactas. Mais precisamente, objetivou-se especificar declarativamente cada atividade operacional primitiva contida no esquema de atividades associado à parte do Modelo da Essência que define o comportamento de um sistema, contribuindo para a formalização desse tipo de especificação e preservando a forma gráfica de representação.

Convém salientar que, utilizando Redes de Petri Compactas, é possível capturar toda a dinâmica inerente à atividade primitiva contida na estrutura conceitual básica que está sendo focalizada. Por definição, um ramo de entrada/saída de uma conexão de Rede de Petri Compacta indica a presença ou ausência de entidades em cada lugar ligado a uma conexão. Assim sendo, ao permitir a modelagem de conjuntos de entidades com cardinalidade variável de acordo com a dinâmica da rede, os lugares de uma Rede de Petri Compacta permitem a modelagem de qualquer elemento funcional passivo (fluxo ou depósito) presente em esquemas de atividades contidos na modelagem da essência do sistema.

Além disso, a abordagem proposta formaliza apenas a relação bem definida que deve existir entre a entrada e a saída da atividade especificada e, dessa forma, presta-se eficazmente à especificação declarativa dessa atividade pois não apresenta qualquer viés procedimental que indicaria como, numa dada alternativa de implementação, essa relação matemática seria estabelecida.

Finalmente, vale indicar que essa abordagem permite:

- graduar a complexidade da representação associada a uma especificação formal de sistemas de grande porte baseada em Redes de Petri Compactas.

  É bastante razoável supor que, utilizando unicamente Redes de Petri Compactas, mesmo empregando as formas de agregação propostas para esse tipo de linguagem, dar-se-ia origem a uma representação de grande complexidade, difícil de validar e manter consistente. Por outro lado, esquemas de atividades tais como os encontrados na modelagem da essência de sistemas, possuem estruturas conceituais básicas simples e intuitivas por se situarem em nível de abstração mais elevado do que o das Redes de Petri Compactas equivalentes. Assim sendo, de acordo com a abordagem aqui preconizada, o comportamento de um sistema de grande porte possuiria uma representação formal abstrata mais simples e intuitiva através desses esquemas e o detalhamento formal que restaria a fazer ficaria circunscrito às atividades operacionais primitivas neles contidas e seria obtido empregando-se os procedimentos indicados no presente trabalho;
- através do reúso das especificações formais associadas a essas estruturas conceituais básicas, aumentar a produtividade da etapa de especificação do sistema a ser construído, bem como aprimorar a qualidade dos produtos nela gerados.

  Sem dúvida, um dos aspectos mais complexos e consumidores de tempo durante a modelagem da essência de um sistema refere-se à sub-etapa de especificação detalhada das atividades operacionais primitivas, que compõem a solução abstrata proposta para o problema a ser resolvido através da construção do sistema. Essa é uma sub-etapa que, inclusive, proporciona a melhor oportunidade para detectar possíveis erros e omissões cometidos durante a fase de elaboração do enunciado desse problema. O reúso de especificações formais depuradas e refinadas ao longo tempo, associadas a estruturas conceituais básicas bem definidas e de ocorrência freqüente, acredita-se, contribuirá para aperfeiçoar, globalmente, a execução da etapa de especificação.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Gane, C. e Sarson T., *Structured Systems Analysis*. Prentice Hall, 1979.

[2] DeMarco, T., *Structured Essential Systems Analysis*. Prentice Hall, 1984.

[3] McMenamin, S. e Palmer J., *Structured Systems Analysis and System Specification*. Prentice Hall, 1979.

[4] Ward, P. T., *Systems Development Without Pain.* Yourdon Press, 1984.

[5] Ward, P. T. e Mellor S. J., *Structured Development of Real-Time Systems*. Yourdon Press, 1985.

[6] Yourdon, E., *Modern Structured Analysis*. Yourdon Press, 1989.

[7] Maffeo, B., *Engenharia de Software e Especificação de Sistemas*. Campus, 1992.

[8] Tao, Y. e Kung C., *Formal Definition and Verification of Data Flow Diagrams*. The Journal of Systems Software (16)1, 29 (1991).

[9] Richter, G. e Maffeo B., *Towards a Rigorous Interpretation of ESML - Extended Systems Modeling Language*. IEEE Transactions on Software Engineering (19)2, 165 (1993).

[10] Sanchez, M. L. e Maffeo B., *Modelagem Conceitual de Sistemas de Tempo-Real — Um Estudo de Caso —*. Monografias em Ciência da Computação do Departamento de Informática da PUC-Rio, 18/96.

[11] Heuser, Carlos A., *Modelagem Conceitual de Sistemas*. Edição Preliminar, V EBAI - Nova Friburgo, Brasil.

[12] Heuser, Carlos A. e Richter G., Constructs for Modeling Information Systems with Petri Nets. *Aplication and Theory of Petri Nets*. Springer-Verlag, 1992 (224 - 243).

[13] Heuser, Carlos A., Peres, Eduardo M. e Richter, G., Towards a complete conceptual model: Petri nets and entity-relationship diagrams. *Information Systems,* volume 18, número 5 (1993).

APÊNDICE

Sintaxe da Linguagem de Anotação (LA)

A sintaxe das LA obedece ao esquema abaixo apresentado em BNF.

```
<fórmula>                 ::=  Verdadeiro |
                               Falso
                               <simb. relação> (<termo>,...) |
                               (<termo> <simb. relação binária>
                               <termo>)|
                               (<fórmula> ∧ <fórmula>) |
                               (<fórmula> ∨ <fórmula>) |
                               ¬ <fórmula> |
                               ((<fórmula> → <fórmula>) |
                               ∀ <var> (<fórmula>) |
                               ∃ <var> (<fórmula>) |
                               (<fórmula>)
<simb. relação>           ::=  (de acordo com o modelo)
<simb. relação binária>   ::=  Elem | Sub | = |
                               (outros de acordo com o modelo)
<termo>                   ::=  <constante> |
                               <var> |
                               <simb. função binária> <termo> ) |
                               {<termo>,...} |
                               {<var> | <fórmula>} |
                               <<termo>,...> |
                               (<termo>)
<simb. função>            ::=  POTEN |
                               (outros de acordo com o modelo)
<simb. função binária>    ::=  x |
                               (outros de acordo com o modelo)
```

Observações sobre esta sintaxe:

- <simb. relação> ou <simb. relação binária> é uma palavra, iniciada por letra maiúscula e podendo conter o símbolo _. Excepcionalmente o símbolo da relação binária pode ser um símbolo especial, como os símbolos =, >, <.
- <simb. função> ou <simb. função binária> é uma palavra, composta exclusivamente de letras maiúsculas e podendo conter o símbolo _. Excepcionalmente o símbolo da função binária pode ser um símbolo especial, como os símbolos +, -, = e x.
- <var> é uma palavra composta exclusivamente de letras minúsculas e podendo conter o símbolo _.

- <constante> é uma cadeia de caracteres, delimitada por apóstrofos. No caso especial de uma constante numérica, os apóstrofos devem ser suprimidos. No caso especial de uma constante que designa conjunto de entidades <constante> é iniciada pela letra C_.
- Como é usual, podem ser suprimidos os parênteses ao redor de expressões para as quais valham a associatividade e a comutatividade, como as expressões com $\wedge$, $\vee$, x.

A sintaxe definida acima não é completa, pois falta a definição de símbolos de relação e de símbolos de função específicos de cada modelo (apenas alguns símbolos estão definidos). A sintaxe define portanto uma classe de linguagens de anotação. Para que uma linguagem específica seja definida dentro dessa classe, é suficiente, portanto, que esses símbolos sejam definidos, juntamente com sua "aridade", isto é, juntamente com o número de argumentos da relação ou função representada pelo símbolo.

Semântica das LA

termos

- <constante>
  Uma constante simboliza, designa uma entidade do UD. As entidades simbolizadas pelas constantes variam de modelo para modelo.
- <var>
  Como já foi mencionado, um termo contendo variáveis só possui significado caso às variáveis estejam atribuídas entidades.
- <símb. função> (<termo>,...)

  O *símb. função* representa uma função entre entidades do UD. O *termo* designa a entidade que é associada, através da função, às entidades designadas pelos demais termos da lista entre parênteses (os parâmetros ou argumentos da função).

Todos os demais termos designam conjuntos, podendo ser considerados como notações especiais de símbolos de funções.

fórmulas

- Verdadeiro | Falso
  Designam os valores lógicos verdadeiro e falso respectivamente. São fórmulas constantes, pouco usadas na prática
- <simb. relação> (<termo>,...) |(<termo> <simb. relação binária> <termo>)
  Um <simb. relação> representa uma relação entre entidades no UD
- (<fórmula> $\wedge$ <fórmula>) |
  (<fórmula> $\vee$ <fórmula>) |
  $\neg$ <fórmula> |
  (<fórmula> $\rightarrow$ <fórmula>)

O significado da A $\rightarrow$ B corresponde ao significado, na linguagem natural, de "se A verdadeiro, então B verdadeiro".

- ∃ <var> (<fórmula>)

  A fórmula é verdadeira se existir ao menos uma entidade no UD que, atribuída à variável <var>, torne a <fórmula> verdadeira.

- ∀ <var> (<fórmula>)

  A fórmula é verdadeira se toda entidade do UD, atribuída à variável <var>, torna a <fórmula> verdadeira.

termos - conjuntos de entidades:

- {<termo,...}
  O termo designa o conjunto, composto pelas entidades designadas pelos termos da lista de termos.

- {<var> | <fórmula>}

  O termo designa o conjunto de todas as entidades que, atribuídas à variável <var>, tornam a fórmula <fórmula> verdadeira.

- < <termo>,...>

  Este termo não designa um conjunto. Ele se encontra nesta lista, apenas por ser usado em conexão como termo do próximo item. O termo designa uma entidade, que consta de um par de entidades, no caso da lista conter dois termos, uma tripla de entidade, no caso da lista conter três entidades, uma quádrupla no caso da lista conter quatro entidades, e assim por diante. A uma entidade desse tipo damos a denominação de n-tupla (dupla, tripla, quádrupla,...), ou tupla.

- (<termo>$_1$ x <termo>$_2$)

  <termo>$_1$ e devem ambos designar conjuntos. O termo designa o conjunto de todos os pares, que podem ser formados, tomando um elemento qualquer do conjunto designado pelo <termo>$_1$ e um elemento qualquer do conjunto designado pelo <termo>$_2$.

- POTEN (<termo>)

  O símbolo de função POTEN representa a função que associa a cada conjunto o seu *conjunto potência*. O conjunto potência de um conjunto é o conjunto de todos os seus conjuntos partes. O conjunto potência é definido pela fórmula:

  POTEN (conj) = {subconj | subconj ⊆ conj }

fórmulas - conjuntos de entidades

- <termo>$_1$ ∈ <termo>$_2$

  A fórmula é verdadeira quando <termo>$_2$ designa um conjunto, e o <termo>$_1$ designa um elemento desse conjunto.

- <termo>$_1$ ⊆ <termo>$_2$

  A fórmula é verdadeira quando ambos termos designam conjuntos, e quando <termo>$_1$ ⊆ {<termo>$_2$}.