PMR 5237
Modelagem e Design de Sistemas Discretos em Redes de Petri
Aula 9: Redes Coloridas e Extensões
Prof. José Reinaldo Silva
reinaldo@usp.br
Modelagem e análise em Redes Colorida/Alto Nível

- Problema
  - Modelagem em CPN/HLPN
    - Análise de atingibilidade (simulação)
  - Modelagem em rede P/T
    - Análise de atingibilidade (simulação)
    - Análise de invariantes
    - Análise de performance
It is *not necessary* for a *user* to know the formal definition of CP-nets:

- The correct *syntax* is checked by the CPN editor, i.e., the computer tool by which CP-nets are constructed.

- The correct use of the *semantics* (i.e., the enabling rule and the occurrence rule) is guaranteed by the CPN simulator and the CPN tools for formal verification.
Correct interpretation means that the “behavior” of practical system (described in some language) is a metaphor for the high level net (syntactically and semantically).

Even if environments can be an apprentice for that there is not guarantee that we get a good metaphor. Only going through the modeling, analysis and verification we can be certain "correctness".
Interpretations can also be:

abstract (like in requirements analysis)

concrete (like in PLC programming)
Use of Petri Nets in Design

1. Requirements
2. Requirements Analysis
3. Model Analysis
4. Design Model
5. Verification/Validation
6. Design
A concrete example
The Gas Station Problem


Requirements Analysis

All transitions must fire to get a complete cycle for using the gas station (it does not matter how many Pumps it has)
- Analysis is typically model-driven to allow e.g. what-if questions.
- Models of both operational processes and/or the information systems can be analyzed.
- Types of analysis:
  - validation
  - verification
  - performance analysis
1. Reachability/coverability graph
2. Structural techniques
   • Place and transition invariants
   • Marking equation
   • Traps, siphons, etc.
3. Simulation

   • Each can be applied to both classical and high-level Petri nets.
   • Nevertheless, for the second we restrict ourselves to classical Petri nets.

Mapping technique/use:
   • reachability graph (validation, verification)
   • invariants (validation, verification)
   • simulation (validation, performance analysis)
CPN Analysis

CPN Analysis follow the same procedures than the classical analysis and face the same problems, even if has new formal resources to include.

Again, we have a state/transition approach, in a discrete flavor, with the possibility of explosion of the number of combinations of states, that is, in the composition of processes.
Modelagem clássica

As redes possuem propriedades típicas dos esquemas que as tornam Uma excelente representação formal para sistemas (dinâmicos) discretos, Entre os quais figuram:

- o princípio da dualidade
- o princípio da localidade
- o princípio da concorrência
- o princípio da representação gráfica
- o princípio da representação algébrica
As in the classic net systems, we detached specific net configurations that constitute a challenge in the analysis process (or situations to be avoided, in order to have the desirable system). Some of this situations are

• Conflict

• Branching and synchronization

• Deadlocks
Distribution and concurrency

As before, everything is based on the concept of locality. According to that, individual states could be classified as independent, and, in such a case, they could be grouped in macro states called cases. Conversely, independent transitions could be also grouped in steps.

Thus, cases and steps could be arranged in a dual way, generating a more abstract net.
Classic and HL modeling

Concluding, the modeling in Classic P/T nets (taken as the abstract archetype of the classic nets) is very similar to the modeling in CPN (or in any high level net).

On the other hand, the inclusion of type theory, or even a simple distinction in the tokens should not be underestimated as good resources to analyze complex systems more comfortably. Such advances is what open new applications to PN and what could make the modeling, analysis (and more recently the verification process) reliable.
Extensions

Even extension elements are used in a similar way. So far we have seen two very important extension elements:

• Gates

• Hierarchy
Analysis in CPN Nets
Directed Graphs

Definition 37

A directed graph is tuple $DG = (V, A, N)$ such that:
(i) $V$ is a set of nodes or vertices;
(ii) $A$ is a set of arcs (or edges) such that $V \cap A = \emptyset$;
(iii) $N$ is a node function or mapping $A \rightarrow V \times V$.

$DG$ is finite if $V$ and $A$ are finite.
O-Graph

**Definition 38**

The full occurrence graph of a CP-net, also called O-graph, is a directed graph $OG = \{ V, A, N \}$ where:

i) $V = |M_0\rangle$;

ii) $A = \{(M_1, b, M_2) \in (V \times BE \times V) | M_1 | b \rangle M_2 \}$,

iii) $\forall (M_1, b, M_2) \in AN(M_1, b, M_2) = (M_1, M_2)$. 
O Graph algorithm

**Proposition 6.3:** The following algorithm constructs the O-graph. The algorithm halts iff the O-graph is finite. Otherwise the algorithm continues forever, producing a larger and larger subgraph of the O-graph.

\[
\begin{align*}
W & := \emptyset \\
\text{Node}(M_0) \\
\text{repeat} \\
& \quad \text{select a node } M_1 \in W \\
& \quad \text{for all } (b,M_2) \in \text{Next}(M_1) \text{ do} \\
& \quad \quad \text{begin} \\
& \quad \quad \quad \text{Node}(M_2) \\
& \quad \quad \quad \text{Arc}(M_1,b,M_2) \\
& \quad \quad \text{end} \\
& \quad \text{remove } M_1 \text{ from } W \\
\text{until } W = \emptyset.
\end{align*}
\]
The invariant method

We first *construct* a set of place invariants.

Then we check whether they are *fulfilled*.

- This is done by showing that each occurring binding element *respects* the invariants.
- The *removed* set of tokens must be identical to the *added* set of tokens – when the weights are taken into account.

Finally, we use the place invariants to *prove* behavioural properties of the CP-net.

- This is done by a *mathematical proof*. 
Automating the invariant analysis

*Automatic calculation* of all place invariants:

- This is possible, but it is a very *complex* task.

- Moreover, it is difficult to represent the results on a *useful form*, i.e., a form which can be used by the system designer.

*Interactive calculation* of place invariants:

- The *user* proposes some of the weights.

- The *tool* calculates the *remaining weights* – if possible.

Interactive calculation of place invariants is *much easier* than a fully automatic calculation.
The invariant method in CPN

- The user needs some ingenuity to *construct* invariants. This can be supported by *computer tools* – interactive process.

- The user also needs some ingenuity to *use* invariants. This can also be supported by *computer tools* – interactive process.

- Invariants can be used to verify a system – without fixing the *system parameters* (such as the number of sites in the data base system).
Invariants are a very important feature in CPN Design. However, we should not expect to solve the design problem by just inserting invariant analysis.

Besides those inherent problems with invariants, the difficulty to apply this approach to large systems is still present.
CP-nets may be large

A typical *industrial application* of CP-nets contains:

- 10-200 *pages*.
- 50-1000 *places and transitions*.
- 10-200 *colour sets*.

This corresponds to *thousands/millions of nodes* in a Place/Transition Net.

Most of the industrial applications would be *totally impossible* without:

- Colours.
- Hierarchies.
- Computer tools.
A hierarchical CP-net contains a number of
interrelated subnets–called pages.
Hierarchy is not anything new and is actually connected with any kind of net, including the classical ones.

In design, hierarchy means to abstract the elements which properties are not relevant in an analysis phases.
Hierarquia em redes clássicas

**Definition 39**

Seja uma estrutura de rede \( N = (S, T; F) \). Seja \( X = S \cup T \) e um sub-cojunto \( Y \subseteq X \). Definimos uma borda de \( N \) como o conjunto \( \partial(N) = \{ y \in Y \, | \, \exists x \notin Y. x \in loc(y) \} \).
Substituição de uma sub-rede

transition bounded substitution

**Definition 40**

Um sub-conjunto de elementos $Y$ da rede $N = (S, T; F)$ é dito limitado por lugar (place bounded) ou aberto, se e somente se $\partial(Y) \subseteq S$. Similarmente, um sub-conjunto $Y$ desta rede é dito limitado por transição (transition bounded), se e somente se $\partial(Y) \subseteq T$. 
place bounded substitution
Se em uma rede com estrutura \( N = (S, T; F) \) existe uma sub-rede \( Y \) limitada por transição, a substituição desta sub-rede \( Y \) gera uma rede \( N' = (S', T'; F') \) onde:

(i) \( S' = S \setminus Y \);

(ii) \( T' = T \setminus Y \cup \{t_y\} \), onde \( t_y \) é o novo elemento que substitui \( Y \);

(iii) \( F' = F \setminus \text{Int}(Y) \) onde \( \text{Int}(Y) \) é o conjunto dos arcos internos de \( Y \).

Similaresmente, se a sub-rede \( Y \) é limitada por lugar,

(i) \( S' = S \setminus Y \cup \{s_y\} \), onde \( s_y \) é o novo elemento que substitui \( Y \);

(ii) \( T' = T \setminus Y \);

(iii) \( F' = F \setminus \text{Int}(Y) \) onde \( \text{Int}(Y) \) é o conjunto dos arcos internos de \( Y \).
Elementos próprios

Seja $x_y$ um elemento genérico (instanciável por $t_y$ ou por $p_y$). Este elemento é dito *próprio* se e somente se é limitado por transição (lugar), tem somente dois elementos de borda, com pelo menos um processo vivo entre eles.

Se os elementos abstratos são próprios as propriedades da rede subjacente se conservam a menos de um termo aditivo. (J. R. Silva, On The Property Analysis of Abstract and Hierarchical Nets, to appear).
Hierarchy is a good abstraction feature. However, the real challenge is to associate that with the property analysis, so that the abstract net preserve the same properties than the expanded one.

The proper requirement is a key issue for that.
A page may contain one or more substitution transitions.

- Each substitution transition is related to a page, i.e., a subnet providing a more detailed description than the transition itself.
- The page is a subpage of the substitution transition.
There is a *well-defined interface* between a substitution transition and its subpage:

- The places surrounding the substitution transition are *socket places*.
- The subpage contains a number of *port places*.
- Socket places are *related* to port places – in a similar way as actual parameters are related to formal parameters in a procedure call.
- A socket place has always the *same marking* as the related port place. The two places are just *different views* of the same place.
Substitution transitions work in a similar way as the refinement primitives found in many system description languages – e.g., SADT diagrams.
Fundamentos do método estruturado

Um bloco é um conjunto genérico de instruções de programa, onde uma dada instrução é identificada como a entrada do bloco e outra (diferente da primeira) é identificada como a saída.

Se A e B são blocos de um mesmo programa, então A e B são ditos independentes se e somente se A ∩ B = ∅.

Se A e B são tais que A ∩ B ≠ ∅ então (A ⊆ B) OU (A ⊆ B)
Constituintes próprios e primos

Elementos próprios indivisíveis são chamados primos. Um conjunto LI de elementos primos pode constituir uma base e portanto pode descrever qualquer programa.
Decomposição por refinamentos: o método estruturado

Top-down

Bottom-up
Estrutura Nassi-Schneiderman
The great advantage in the structured method is its accumulative nature in the forward (top down) development. Conversely, it adds compositionality in the bottom up constructive development. Such a combination is the key issue that makes the structured approach a good feature in modeling and design.

Petri Nets has been criticized by its lack of compositionality. The inclusion of hierarchy (in both classic and high level nets) is the best answer to that critic.
Para além da estruturação

Redes clássicas
Hierárquicas e de alto Nível

→

Redes Orientadas a Objetos
Cronograma de atividades:

faltam mais 2 aulas (21 e 28 de maio)

mais uma lista de exercícios (lista 5)

Entrega do artigo final: 11 de junho
Fim