QUALITATIVE MODELLING AND ANALYSIS CONCURRENT SYSTEMS WITH PETRI NETS

Monika Heiner

BTU COTTBUS, COMPUTER SCIENCE

http://www-dssz.informatik.tu-cottbus.de

OUTLINE

1. MOTIVATION

2. INTRODUCTION INTO QUALITATIVE MODELLING

3. INTRODUCTION INTO QUALITATIVE ANALYSIS PROPERTIES
   REACHABILITY GRAPH
   [TRANSITION / PLACE INVARIANTS]

4. SUMMARY, OUTLOOK

5. REFERENCES
1. 

**MOTIVATION**

---

**WHY PETRI NETS?**

- a suitable intermediate representation for
  - different languages
  - different stages of certainty
- modelling power
  - partial order semantics
  - applicable on any abstraction level
  - specification of limited resources possible
- analysis power
  - various complementary analysis methods
  - reliable tools
- integration of qualitative and quantitative analyses
- **BUT:**
  - modelling power <-> analysis power
PETRI NETS, A BIT OF HISTORY

- Carl Adam Petri, 1962, PhD University of Technology Darmstadt
  -> basic ideas introduced

- early 1970's
  -> first papers contributing to Petri net theory

- Petri, 1976
  -> application to chemical networks mentioned

- early 1980's
  -> first monographs appear

- Reddy, 1993
  -> first paper on bio application

- late 1990's
  -> increasing interest in applying Petri nets for modelling and analysis of bio networks

2. INTRODUCTION INTO QUALITATIVE MODELLING
**PETRI NETS, STRUCTURE**

- **two types of nodes**
  - places: "passive elements", conditions, local states, chemical compounds
  - transitions: "active elements", events, actions, chemical reactions

- **arcs**
  - directed
  - never arcs between nodes of the same type
  - for any node, arbitrary number of pre-nodes and post-nodes

- **arc inscriptions**
  - arc weight / multiplicity
  - amount of units of the substances involved in the basic (re-) action

**PETRI NETS, SYSTEM STATE**

- **tokens**
  - moving objects, e.g. units of substances (e.g. Mole), ...
  - condition is not fulfilled
  - condition is (one times) fulfilled
  - condition is n times fulfilled

- **marking**
  - How many tokens are on each place?
  - system state
    - substance distribution
  - initial marking
    - initial substance distribution
PETRI NETS, BEHAVIOUR

- flow of tokens
  -> defined by firing rule

- an action **may** happen (fire), if
  -> all preconditions are fulfilled (corresponding to the arc weights);

- **if** an action happens (fires), then
  -> tokens are removed from all preconditions (corresponding to the arc weights), **and**
  -> tokens are added to all postconditions (corresponding to the arc weights);

- an action happens (firing of a transition)
  -> atomic
  -> time-less
TYPICAL BASIC STRUCTURES

- **CHAIN OF REACTIONS**
  - MB1 → r1 → MB2 → r2 → MB3

- **(FREE-CHOICE) BRANCHING / CONFLICT**
  - MB1 → r1 → MB2
  - MB1 → r2 → MB3

- **CONCURRENCY**
  - r1, r2, r3 are concurrent = independent

EXAMPLE, CARBON OXIDATION, BASIC REACTIONS

r1) $2 \text{C} + \text{O}_2 \rightarrow 2 \text{CO}$

r2) $\text{C} + \text{O}_2 \rightarrow \text{CO}_2$

r3) $\text{C} + \text{CO}_2 \leftrightarrow 2 \text{CO}$

- carbon monoxide
- carbon dioxide
- oxygen
**EXAMPLE, COMPOSITION**

**BASIC MODEL**

```
\[
\begin{align*}
\text{r1:} & \quad 2 \text{C} + \text{O}_2 \rightarrow 2 \text{CO} \\
\text{r3a:} & \quad \text{C} + \text{O}_2 \rightarrow \text{CO} \\
\text{r3b:} & \quad \text{C} + \text{CO}_2 \leftrightarrow 2 \text{CO} \\
\end{align*}
\]
```

**SYSTEM’S TOTAL EQUATION**

1) \(2 \text{C} + \text{O}_2 \rightarrow 2 \text{CO}\)
2) \(\text{C} + \text{O}_2 \rightarrow \text{CO}\)
3) \(\text{C} + \text{CO}_2 \leftrightarrow 2 \text{CO}\)

\(3 \text{C} + 2 \text{O}_2 \rightarrow 2 \text{CO} + \text{CO}_2\)

**MODEL OF THE SYSTEM’S TOTAL EQUATION**

```
\[
\begin{align*}
\text{C:} & \quad 3 \\
\text{CO}_2: & \quad 2 \\
\text{CO:} & \quad 2 \\
\text{O}_2: & \quad 2 \\
\end{align*}
\]
```

**ENVIRONMENT BEHAVIOUR**

- strong assumptions about quantitative relations of input / output compounds
- 'inverse' total equation
- there are no boundary nodes
CARBON OXIDATION,
SYSTEM MODEL,

```
carbon2.spped
```

3. INTRODUCTION INTO QUALITATIVE ANALYSIS
# Petri Net Properties, Overview / INA

## 1. Simple Structural Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORD</td>
<td>ordinary (1-multiplicity of all arcs)</td>
</tr>
<tr>
<td>HOM</td>
<td>homogeneous (all output arcs of a given place have the same multiplicity)</td>
</tr>
<tr>
<td>NBM</td>
<td>non-blocking multiplicity (for each place applies: MIN multiplicity of input arcs (\geq) MAX multiplicity of output arcs)</td>
</tr>
<tr>
<td>PUR</td>
<td>pure (no side conditions)</td>
</tr>
<tr>
<td>CSV</td>
<td>conservative (any firing preserves token amount)</td>
</tr>
<tr>
<td>SCF</td>
<td>static conflict free</td>
</tr>
<tr>
<td>CON</td>
<td>connected</td>
</tr>
<tr>
<td>SC</td>
<td>strongly connected</td>
</tr>
<tr>
<td>Ft0</td>
<td>there is a transition without pre-place</td>
</tr>
<tr>
<td>tF0</td>
<td>there is a transition without post-place</td>
</tr>
<tr>
<td>Fp0</td>
<td>there is a place without pre-transition</td>
</tr>
<tr>
<td>pF0</td>
<td>there is a place without post-transition</td>
</tr>
<tr>
<td>MG</td>
<td>marked graph (synchronization graph)</td>
</tr>
<tr>
<td>SM</td>
<td>state machine</td>
</tr>
<tr>
<td>FC</td>
<td>free choice net</td>
</tr>
<tr>
<td>EFC</td>
<td>extended free choice net</td>
</tr>
<tr>
<td>ES</td>
<td>extended simple net</td>
</tr>
</tbody>
</table>

## 2. More Expensive Structural Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTP</td>
<td>deadlock trap property</td>
</tr>
<tr>
<td>SMC</td>
<td>state machine coverable (covered with SM components)</td>
</tr>
<tr>
<td>SMD</td>
<td>state machine decomposable (covered with SCSM components)</td>
</tr>
<tr>
<td>SMA</td>
<td>state machine allocatable</td>
</tr>
<tr>
<td>CPI</td>
<td>covered with place invariants</td>
</tr>
<tr>
<td>CTI</td>
<td>covered with transition invariants</td>
</tr>
<tr>
<td>SB</td>
<td>structurally bounded</td>
</tr>
</tbody>
</table>

## 3. Behavioural Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>bounded</td>
</tr>
<tr>
<td>REV</td>
<td>reversible (the initial state (m_0) can be reached again from all reachable states: home state)</td>
</tr>
<tr>
<td>DSt</td>
<td>dead states (a state where no transition is enabled)</td>
</tr>
<tr>
<td>BSt</td>
<td>bad states (a state where a fact is enabled)</td>
</tr>
<tr>
<td>DTr</td>
<td>dead transitions (at the initial state)</td>
</tr>
<tr>
<td>DCF</td>
<td>dynamically conflict free</td>
</tr>
<tr>
<td>L</td>
<td>live</td>
</tr>
<tr>
<td>LV</td>
<td>live, excepted transitions dead at the initial marking</td>
</tr>
<tr>
<td>L&amp;S</td>
<td>live &amp; safe (1-bounded)</td>
</tr>
</tbody>
</table>
**BEHAVIOURAL NET PROPERTIES, OVERVIEW**

- **MARKABILITY of places**
  - markable (place liveness)
  - k-bounded (1-bounded / safe)
  - unbounded

- **LIVENESS of transitions**
  - zero times firing ($m_0$-dead)
  - finite times firing (dead, non-live)
  - infinite times firing, probably (live)
  - infinite times firing, definitely (livelock free)

- **REACHABILITY of states**
  - dead states
  - reproducibility
  - reversibility ($m_0$ - home state)
  - bad states (facts)
  - user-specified states

- **NET INVARIANTS**
  - transition invariants
  - place invariants
  - temporal relationship of logic formulae
    - safety properties
    - progress properties

**QUALITATIVE ANALYSIS METHODS, OVERVIEW**

- **NET REDUCTION**

- **STRUCTURAL PROPERTIES**

- **LINEAR PROGRAMMING**
  - place / transition invariants
  - state equation
  - trap equation

- **STATE SPACE ANALYSIS**
  - (complete) reachability graph

- **MODEL CHECKING**
  - compressed state spaces
    - BDDs, NDDs, ..., xDDs
    - Kronecker products
  - reduced state spaces
    - coverability graph
    - symmetry
    - stubborn sets
  - branching process

---

*monika.heiner@informatik.tu-cottbus.de*
**REACHABILITY GRAPH (RG)**

- **nodes** - system states
- **arcs** - the (single) firing transition
- **example** - carbon oxidation, environment style 3

```
* init

start

r1

r2

r3a

r3b

O2

C

CO

CO2

repeat
```

carbon2.ssped

---

**RG (CARBON OXIDATION)**

```
\[\text{s1: (init)}\]

\[\text{s2: (2 O}_2\text{, 3 C)}\]

\[\text{s3: (O}_2\text{, 2 C, 2 CO)}\]

\[\text{s4: (O}_2\text{, 2 C, CO}_2\text{)}\]

\[\text{s5: (2 CO, CO}_2\text{)}\]

\[\text{s6: (C, 2 CO}_2\text{)}\]

\[\text{concurrency}\]

\[\text{repeat}\]

\[\text{conflict}\]
```

\[\rightarrow \text{ interleaving description of the whole system behaviour}\]
EXAMPLE: RG AND THREE BASIC PN PROPERTIES

- no concurrency
  - $\text{rg}(\text{pn}) = \text{pn}$

- $\text{rg}$ - finite
  - bounded $\text{pn}$

- $\text{rg}$ - not sc
  - $\text{pn}$ not reversible

- no dead states, but liveness?

- condensed $\text{rg}$
  - node - sc component (scc)
    - scc: maximal set of sc nodes;
      - a terminal scc
        - possible terminal system behaviour
        - must contain all transitions in a live $\text{pn}$

- not all terminal scc contain all transitions
  - $\text{pn}$ is not live

BASIC PROPERTIES & RG, SUMMARY

- How many tokens may reside at most in a given place . . .
  - $(0, 1, k, \infty)$?
    - boundedness
    - $\text{rg}$ is finite

- How often may a transition fire . . .
  - $(0\text{-times, } n\text{-times, } \infty\text{-times})$?
    - liveness
    - every terminal scc contains all transitions

- Is the initial system state . . .
  - always reachable again ?
    - reversibility
    - $\text{rg}$ is sc (consists of one scc)
RG(CARBON OXIDATION),
EVALUATION

RG is finite
->  BND

1 Strongly Connected Component (SCC)
->  REV

the only SCC contains all transitions
->  LIVE

PROCEDURE rg (IN Net pn, IN Marking m0,  
OUT MSet nodes, OUT ArcSet arcs);

MSet $U = \{m0\}$,  // unprocessed markings
N $= \emptyset$;  // rg nodes
ArcSet $E = \emptyset$;  // rg arcs (pre, post, t)
Marking $m'$;  // successor marking
Transition $t$;

WHILE $U \neq \emptyset$ DO
    choose one $m \in U$;
    $U = U \setminus \{m\}$;  $N = N \cup \{m\}$;

    FOR ALL $t$ enabled at $m$ DO
        $m' = m + \Delta t$;
        IF $m' \notin N \cup U$  // new marking
            THEN $U = U \cup \{m'\}$
            ENDIF;
        $E = E \cup \{(m, m', t)\}$
    ENDFOR

ENDWHILE;

nodes $= N$; arcs $= E$;

ENDPROC rg.
REACHABILITY GRAPH, OBSERVATIONS

- **unbounded** Petri net
  - $\Rightarrow$ the RG is **infinite**
- **bounded** Petri net
  - $\Rightarrow$ the RG is **finite**

- simple construction algorithm
  - $\Rightarrow$ **single step firing rule**

- concurrency
  - $\Rightarrow$ **enumeration of all interleaving sequences**

- branching arcs in the RG
  - $\Rightarrow$ **conflict OR concurrency**

- RG tend to be very large
  - $\Rightarrow$ **automatic evaluation necessary**

- worst case: over-exponential growth
  - $\Rightarrow$ **alternative analyses techniques ?**

---

5.

**SUMMARY AND OUTLOOK**
**PETRI NETS**

**MODEL CLASSES**

**PLACE/TRANSITION PETRI NET**
- validation by Petri net theory

**TIME-DEPENDENT PN**
- *DISCRETE* *)
  - worst-case evaluation
- CONTINUOUS PETRI NET
  - behaviour prediction
- STOCHASTIC PETRI NET
  - reliability prediction

*) DISCREETLY TREATABLE

**REFERENCES**

**PETRI NETS, THEORY**
- Baumgarten, B.: Petri Nets, Principles and Applications (in German); Wissenschaftsverlag 1990.
- Lautenbach, K.: Exact Liveness Conditions of a Petri Net Class (in German); Berichte der GMD 82, Bonn 1973.
- Starke, P. H.: Analysis of Petri Net Models (in German); B. G. Teubner 1990.

For more related books and papers see [http://www.daimi.au.dk/PetriNets/ -> Petri Net Bibliography](http://www.daimi.au.dk/PetriNets/)

**PETRI NETS, TOOLS**