Reachability Analysis

prof.dr.ir. Wil van der Aalst
www.vdaalst.com
Strong point of Petri nets

Math. foundation

Graphical notation

Compactness

Concurrency, locality

Analysis techniques

Tool support

Today’s lecture
Outline

• What do we want to analyze?
• The reachability graph revised
• Petri net standard properties
• Reachability analysis in CPN Tools
• Workflow nets and soundness
• Model checking
Design-time analysis
Design-time analysis vs run-time analysis

Design-time analysis

- People
- Processes
- Services
- Components
- Organizations

Run-time analysis

- Event logs
- Discovery
- Conformance
- Extension
- Models
- Analyzes

Validation

Verification

Performance analysis

- Design-time models
- E.g., process models represented in BPMN, BPEL, EPCs, Petri nets, UML AD, etc. or other types of models such as social networks, organizational networks, decision trees, etc.

- Run-time models
- E.g., systems like WebSphere, Oracle, TIBCO/Staffware, SAP, FLOWer, etc.

- Supports/controls
- Specifies configures implements analyzes
- Records events, e.g., messages, transactions, etc.
What do we want to analyze? – Design time

- **Goal:** Gain insight into the behavior of the information system to be designed and/or the underlying business processes
- Analyze process model that models its behavior
  - Check whether model conforms to specification
  - Ask questions about information system

Qualitative properties

Quantitative properties
Analysis techniques

- **Validation**
  - Model must reflect what it intends to represent
  - Simulation

- **Verification**
  - Investigate relation between specification and model
  - Reachability analysis
  - Structural analysis (e.g., invariants)

- **Performance analysis**
  - Make predictions about key performance indicators (e.g., flow time)
  - Simulation (for simple processes also Markov analysis)
What do we want to analyze? – **Runtime**

- Analyze the *running* information system
- Continuous improvement (redesign and organic growth)
- Use *event logs* for analysis

**Analysis techniques**
- Monitoring
- Data mining
- Process mining

Also see new Bachelor course on Business Process Intelligence (2IIE0) and Master Course on Process Mining (2II66)!
What is a Petri net?

- A graphical notion (model = picture?)
- A mathematical notion (model = graph?)
- A programming notion (model = program?)

- A solver independent medium
- Starting point for an abundance of analysis approaches
Petri net analysis

Mapping technique/use:
1. Reachability graph (validation, verification)
2. Structural analysis (validation, verification)
3. Simulation (validation, performance analysis)

- Each technique can be applied to CPNs.
- For (1) and (2), we restrict ourselves to Petri nets.
Outline

- What do we want to analyze?
- The reachability graph revised
- Petri net standard properties
- Reachability analysis in CPN Tools
- Workflow nets and soundness
- Model checking
Reachability graph

Recall mapping of a Petri net onto a transition system

$s = m: P \rightarrow N$

Reachability graph is the reachable portion of the transition system (reachable from initial marking)
Reachability graph algorithm

1) Label the initial marking $m_0$ as the root and tag it "new".

2) While "new" markings exist, do the following:
   a) Select a new marking $m$.
   b) If no transitions are enabled at $m$, tag $m$ "dead-end".
   c) While there exist enabled transitions at $m$, do the following for each enabled transition $t$ at $m$:
      i. Obtain the marking $m'$ that results from firing $t$ at $m$.
      ii. If $m'$ does not appear in the graph add $m'$ and tag it "new".
      iii. Draw an arc with label $t$ from $m$ to $m'$ (if not already present).

3) Output the graph
Example

Each state represented as a multiset

Alternatively, as a vector:

(1,0,0,1,1,0,0) for ordering
(r1,g1,o1,x,r2,g2,o2)
color Name = string;
color Address = string;
color DateOfBirth = string;
color PatientID = int;
color EmpNo = int;
color Experience = int;
color Gender = with male|female;
color Pat = product PatientID * Name * Address * DateOfBirth * Gender;
color Emp = product EmpNo * Experience;
color EP = product Emp * Pat;
var p:Pat;
var emp:Emp;
val Peter = (12345, "Peter", "Kerkstraat 10, Amsterdam", "13-Dec-1962", male);
val Sara = (23456, "Sara", "Rozenstraat 2, Amsterdam", "1-May-1974", female);
val Ann = (641112, 7);
Solution

Edge annotated by a binding

wait=[Peter, Sara], free=[Ann], busy=[], done=[]

(start, <emp=Ann,p=Peter>)

wait=[Sara], free=[], busy=[(Ann, Peter)], done=[]

(stop, <emp=Ann,p=Peter>)

wait=[Sara], free=[Ann], busy=[], done=[Peter]

(start, <emp=Ann,p=Sara>)

wait=[Peter], free=[], busy=[(Ann, Sara)], done=[]

(stop, <emp=Ann,p=Sara>)

wait=[Peter], free=[], busy=[], done=[Sara]

(start, <emp=Ann,p=Sara>)

wait=[Sara], free=[Ann], busy=[], done=[Peter]

(stop, <emp=Ann,p=Sara>)

wait=[Sara], free=[Ann], busy=[(Ann, Sara)], done=[Peter]

(stop, <emp=Ann,p=Peter>)

wait=[], free=[], busy=[(Ann, Sara)], done=[Peter]

(stop, <emp=Ann,p=Peter>)

wait=[], free=[], busy=[(Ann, Peter)], done=[Sara]

(stop, <emp=Ann,p=Peter>)

wait=[Sara], free=[Ann], busy=[], done=[Peter, Sara]
Reachability graph (summary)

- Can be automatically computed if the number of states is **finite**
- Adding time and data often yields an **infinite** reachability graph, making analysis intractable
- Hierarchy does not increase expressiveness of a model
  → **Compute reachability graph of the flattened model**
- Next: use reachability graph to determine properties of a Petri net
Outline

- What do we want to analyze?
- The reachability graph revised
- Petri net standard properties
- Reachability analysis in CPN Tools
- Workflow nets and soundness
- Model checking
Boundedness

• **Intuition**: Ensure limited capacity of some place

• A Petri net is \( k \)-bounded (or bounded for short) if and only if no place \( p \) in \( P \) contains more than \( k \) tokens in any reachable marking. Otherwise, the net is unbounded. If \( k=1 \), the Petri net is safe.

• A bounded Petri net has finitely many reachable markings
Bounded?

Reachability graph has infinitely many nodes
Terminating

• **Intuition**: The Petri net always reaches a terminal marking

• A Petri net is **terminating** if every run is finite.

• Petri net with a **finite and acyclic reachability graph** is terminating

• A terminating Petri net has only **finitely many runs**
Terminating?

One run

Infinitely many runs

[Image of a Petri net with transitions and states labeled 'wait', 'start', 'stop', 'free', 'busy', 'done', and tokens 'p1', 'p2', 'p3', 't1', 't2', 't3'].

- One run:
  - Start with 'wait', then transition to 'free', then 'done'.
  - Transition to 'busy', then 'stop', then 'wait, free, done', then 'start', then 'busy, done', then 'stop', then 'free, 2-done'.

- Infinitely many runs:
  - Start with 'p1', then 't1', then 'p2', then 't2', then 'p1,p3', then 't1', then 'p2,p3', then 't2', then 'p1,2-p3', then 't1', then 'p2,2-p3', then 't2', then '...',
  - Includes cycles and multiple states.

[Image of aPetri net with transitions and states labeled 'wait', 'start', 'stop', 'free', 'busy', 'done', and tokens 'p1', 'p2', 'p3', 't1', 't2', 't3'].

- One run:
  - Start with 'wait', then transition to 'free', then 'done'.
  - Transition to 'busy', then 'stop', then 'wait, free, done', then 'start', then 'busy, done', then 'stop', then 'free, 2-done'.

- Infinitely many runs:
  - Start with 'p1', then 't1', then 'p2', then 't2', then 'p1,p3', then 't1', then 'p2,p3', then 't2', then 'p1,2-p3', then 't1', then 'p2,2-p3', then 't2', then '...',
  - Includes cycles and multiple states.
Deadlock freedom

- **Intuition**: The Petri net can reach a terminal marking

- A Petri net is **deadlock free** if at least one transition is enabled at every reachable marking
Deadlock free?

\[
\begin{align*}
\text{[p1]} & \xrightarrow{t1} \text{[p2]} & \xrightarrow{t2} \text{[p1,p3]} & \xrightarrow{t1} \text{[p2,p3]} & \xrightarrow{t2} \text{[p1,2\cdot p3]} & \xrightarrow{t1} \text{[p2,2\cdot p3]} & \xrightarrow{t2} \ldots \\
\end{align*}
\]
Dead transition

• **Intuition**: A transition can in principle occur (its implemented functionality can be used)

• A transition $t$ of a Petri net is **dead** if $t$ is not enabled at any reachable marking.
No dead transitions?
Liveness

• **Intuition**: A transition can always become enabled again

• A transition $t$ is **live** if from every reachable marking $m$ there is a marking $m'$ reachable such that $t$ is enabled at $m'$. A Petri net is **live** if every transition $t$ in $T$ is live.

• Liveness and terminating exclude each other
• A live Petri net does not have dead transitions
Basic idea of liveness

all reachable markings

markings where t is enabled
Live?

[Diagram of a Petri net with transitions and places labeled t1, t2, t3, p1, p2, p3, and actions looks_for_documents, puts_away_documents, performs_actions, reads_documents. The diagram shows a correct sequence of transitions and a faulty sequence marked with an 'X'.]
• **Intuition**: A marking can always be reached again

• A marking $m$ is a **home-marking** if from any reachable marking we can reach $m$. A Petri net is **reversible** if its initial marking is a home-marking.

• A Petri net is reversible if and only if its reachability graph is strongly connected.
Home-markings?

All reachable markings are home-markings; net is also reversible.

Not a home marking, however, if there is a single unavoidable dead marking, it will be a home marking.
Home-markings?

[Diagram showing a state transition diagram with states including `wait`, `busy`, `done`, `start`, and `stop`.

- From `wait`, transitions to `start` and `stop`.
- From `start`, transitions to `stop`.
- From `stop`, transitions to `done`.
- From `busy`, transitions to `done`.

Transitions:
- `wait` to `free`.
- `free` to `busy`.
- `done` to `free`.
- `stop` to `busy`.
- `busy` to `done`.
- `start` to `stop`.
- `stop` to `start`.

States:
- `2·wait, free`
- `wait, busy`
- `wait, free, done`
- `busy, done`
- `free, 2·done`]

✓
### Live (L), Deadlock Free (DF), Bounded (B), Reversible (R)

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

provide marked Petri net for each cell (if possible)
Liveness implies deadlock-freeness

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td>![Diagram]</td>
<td>![Cross]</td>
<td>![Question Mark]</td>
<td></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td>![Diagram]</td>
<td>![Cross]</td>
<td>[Diagram]</td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td>![Diagram]</td>
<td>![Cross]</td>
<td>![Question Mark]</td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td>![Diagram]</td>
<td>![Cross]</td>
<td>[Diagram]</td>
<td></td>
</tr>
</tbody>
</table>
Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td>?</td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
</tr>
<tr>
<td>Conditions</td>
<td>Deadlock Free (DF)</td>
<td>Not Deadlock Free (NDF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
<td>------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Live and Deadlock Free (L,DF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Live and not Deadlock Free (L,NDF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Live and Deadlock Free (NL,DF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Live and Not Deadlock Free (NL,NDF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Reversible (R)</th>
<th>Not Reversible (NR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td>?</td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td></td>
</tr>
</tbody>
</table>
### Give marked Petri net

<table>
<thead>
<tr>
<th>Condition</th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td><img src="diagram1.png" alt="Diagram" /></td>
<td><img src="diagram2.png" alt="Diagram" /></td>
<td><img src="diagram3.png" alt="Diagram" /></td>
<td><img src="diagram4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td><img src="diagram5.png" alt="Diagram" /></td>
<td><img src="diagram6.png" alt="Diagram" /></td>
<td><img src="diagram7.png" alt="Diagram" /></td>
<td>?</td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td><img src="diagram8.png" alt="Diagram" /></td>
<td><img src="diagram9.png" alt="Diagram" /></td>
<td><img src="diagram10.png" alt="Diagram" /></td>
<td><img src="diagram11.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td><img src="diagram12.png" alt="Diagram" /></td>
<td><img src="diagram13.png" alt="Diagram" /></td>
<td><img src="diagram14.png" alt="Diagram" /></td>
<td>?</td>
</tr>
</tbody>
</table>
Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bounded and Reversible (B,R)</strong></td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>Bounded and Not Reversible (B,NR)</strong></td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>Not Bounded and Reversible (NB,R)</strong></td>
<td><img src="image9" alt="Diagram" /></td>
<td><img src="image10" alt="Diagram" /></td>
<td><img src="image11" alt="Diagram" /></td>
<td><img src="image12" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>Not Bounded and Not Reversible (NB,NR)</strong></td>
<td><img src="image13" alt="Diagram" /></td>
<td><img src="image14" alt="Diagram" /></td>
<td><img src="image15" alt="Diagram" /></td>
<td><img src="image16" alt="Diagram" /></td>
</tr>
</tbody>
</table>
## Give marked Petri net

<table>
<thead>
<tr>
<th>Bounded and Reversible (B,R)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Net" /></td>
<td><img src="image2" alt="Net" /></td>
<td><img src="image3" alt="Net" /></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image4" alt="Net" /></td>
<td><img src="image5" alt="Net" /></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image6" alt="Net" /></td>
<td><img src="image7" alt="Net" /></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image8" alt="Net" /></td>
<td><img src="image9" alt="Net" /></td>
<td></td>
</tr>
</tbody>
</table>
Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Not Live and Deadlock Free (L,NDF)</th>
<th>Not Live and Not Deadlock Free (NL,DF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
</tbody>
</table>

Does not exist: initial marking cannot be dead (otherwise not unbounded) and we can always return to this non-dead initial marking (net is reversible). Hence, the model has to be deadlock free thus creating a contradiction.
### Give marked Petri net

<table>
<thead>
<tr>
<th>State</th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td><img src="image9" alt="Diagram" /></td>
<td><img src="image10" alt="Diagram" /></td>
<td><img src="image11" alt="Diagram" /></td>
<td><img src="image12" alt="Diagram" /></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td><img src="image13" alt="Diagram" /></td>
<td><img src="image14" alt="Diagram" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Give marked Petri net

<table>
<thead>
<tr>
<th>Bounded and Reversible (B,R)</th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Give marked Petri net

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded and Reversible (B,R)</td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
</tr>
<tr>
<td>Bounded and Not Reversible (B,NR)</td>
<td><img src="image4" alt="Diagram" /></td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
</tr>
<tr>
<td>Not Bounded and Reversible (NB,R)</td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
<td><img src="image9" alt="Diagram" /></td>
</tr>
<tr>
<td>Not Bounded and Not Reversible (NB,NR)</td>
<td><img src="image10" alt="Diagram" /></td>
<td><img src="image11" alt="Diagram" /></td>
<td><img src="image12" alt="Diagram" /></td>
</tr>
</tbody>
</table>
## Overview

<table>
<thead>
<tr>
<th></th>
<th>Live and Deadlock Free (L,DF)</th>
<th>Live and not Deadlock Free (L,NDF)</th>
<th>Not Live and Deadlock Free (NL,DF)</th>
<th>Not Live and Not Deadlock Free (NL,NDF)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bounded and Reversible (B,R)</strong></td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>Bounded and Not Reversible (B,NR)</strong></td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>Not Bounded and Reversible (NB,R)</strong></td>
<td><img src="image9" alt="Diagram" /></td>
<td><img src="image10" alt="Diagram" /></td>
<td><img src="image11" alt="Diagram" /></td>
<td><img src="image12" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>Not Bounded and Not Reversible (NB,NR)</strong></td>
<td><img src="image13" alt="Diagram" /></td>
<td><img src="image14" alt="Diagram" /></td>
<td><img src="image15" alt="Diagram" /></td>
<td><img src="image16" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Another Example

• The marked Petri net is:
  ✓ deadlock free
  ✓ live
  ✓ bounded
  ✓ safe
  ✓ reversible
  ✓ all markings are home markings
• What do we want to analyze?
• The reachability graph revised
• Petri net standard properties
• Reachability analysis in CPN Tools
• Workflow nets and soundness
• Model checking
Analysis in CPN Tools

- State-space analysis
- Generate report in text file
- State-space visualization

Steps:
1. Calculating state space (incl. generating ML code)
2. Calculating the SCC graph (to calculate home states and fairness)
3. Save/view state space report
Example
Create report

Statistics

State Space
Nodes: 24
Arcs: 44
Secs: 0
Status: Full

Scc Graph
Nodes: 24
Arcs: 44
Secs: 0

Boundedness Properties

Best Integer Bounds

<table>
<thead>
<tr>
<th></th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>main'database 1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>main'deposit 1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>main'withdraw 1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Best Upper Multi-set Bounds

main'database 1  1'(1,0)++
1'(2,(-5))++
1'(2,(-1))++
1'(2,0)++
1'(2,3)++
CPN Tools state space report for:
/cygdrive/D/courses/BIS-2013/CPN-files/state-space/bank.cpn

Statistics

State Space
Nodes: 24
Arcs: 44
Secs: 0
Status: Full

Scc Graph
Nodes: 24
Arcs: 44
Secs: 0
## Boundedness Properties

### Best Integer Bounds

<table>
<thead>
<tr>
<th></th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>main'database 1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>main'deposit 1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>main'withdraw 1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

### Best Upper Multi-set Bounds

- **main'database 1**: $1'(1,0)++ 1'(2, -5)++ 1'(2, -1)++ 1'(2, 0)++ 1'(2, 3)++ 1'(2, 4)++ 1'(2, 7)++ 1'(2, 8)++ 1'(2, 11)++ 1'(2, 12)++ 1'(2, 15)++ 1'(2, 16)++ 1'(2, 20)++ 1'(3, -9)++ 1'(3, 0)
- **main'deposit 1**: $5'(2, 4)
- **main'withdraw 1**: $1'(2, 5)++ 1'(3, 9)

### Best Lower Multi-set Bounds

- **main'database 1**: $1'(1,0)$
- **main'deposit 1**: empty
- **main'withdraw 1**: empty
Report (3)

Home Markings
[24]

Liveness Properties

Dead Markings
[24]

Dead Transition Instances
None

Live Transition Instances
None

Fairness Properties

No infinite occurrence sequences.
A path from node 1 to node 24 is: [1, 2, 5, 9, 13, 18, 23, 24]
val it = true : bool

Reachable'(1,24)

There is no path from 2 to 1
val it = false : bool

Reachable'(2,1)
See also the Web page
Dining philosophers
Home Markings
All

Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All

How many reachable states?
Home markings?
Dead markings?
Dead transitions?
Live transitions?

State Space
Nodes: 11
Arcs: 30
Secs: 0
Status: Full

Scc Graph
Nodes: 1
Arcs: 0
Secs: 0
How many reachable states?
Home markings?
Dead markings?
Dead transitions?
Live transitions?

State Space
Nodes: 11
Arcs: 30
Secs: 0
Status: Full

Scc Graph
Nodes: 1
Arcs: 0
Secs: 0

Home Markings
All

Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All
### Best Integer Bounds

<table>
<thead>
<tr>
<th>Page/Name</th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page'Eat 1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Page'Think 1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Page'Unused_Chopsticks 1</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
Non-atomic take/release

State Space
Nodes: 1473
Arcs: 6270
Secs: 0
Status: Full

Scc Graph
Nodes: 3
Arcs: 10
Secs: 0

Home Markings
None

Dead Markings
[292,503]

Dead Transition Instances
None

Live Transition Instances
None

How many reachable states?
Home markings?
Dead markings?
Dead transitions?
Live transitions?

no home markings, however, if there is a single always reachable dead marking, it will be a home marking
Dead Markings [292,503]
Dead Markings

[292,503]
Non-atomic take/release (take right first)

State Space
Nodes: 392
Arcs: 1415
Secs: 0
Status: Full

Scc Graph
Nodes: 2
Arcs: 5
Secs: 0

How many reachable states?
Home markings?
Dead markings?
Dead transitions?
Live transitions?

Home Markings [97]
Dead Markings [97]
Dead Transition Instances None
Live Transition Instances None
How many reachable states?
Home markings?
Dead markings?
Dead transitions?
Live transitions?

State Space
Nodes: 14158
Arcs: 81848
Secs: 34
Status: Full

Scc Graph
Nodes: 2
Arcs: 8
Secs: 0

Home Markings
[2944]

Dead Markings
[2944]

Dead Transition Instances
None

Live Transition Instances
None

How about 8 philosophers?
Outline

- What do we want to analyze?
- The reachability graph revised
- Petri net standard properties
- Reachability analysis in CPN Tools
- Workflow nets and soundness
- Model checking
Modeling processes as workflow nets

Well-defined starting point

Nodes are connected

Well-defined ending point
A Petri net $N = (P, T, F)$ is a workflow net if

1. **Object creation**: $N$ contains an input place $i$ (the source place) such that $\bullet i = \emptyset$.

2. **Object completion**: $N$ contains an output place $o$ (the sink place) such that $o^* = \emptyset$.

3. **Connectedness**: Every node in $N$ is on a path from $i$ to $o$. 
Soundness

• One of the most established correctness criteria for workflow nets
• Soundness guarantees that the net has always the possibility to complete (from every reachable marking (from [i]) there is a path to marking [o])

![Workflow Net Diagram]
A workflow net $N = (P, T, F)$ with input place $i$ and output place $o$ and initial marking $[i]$ is **sound** if

1. **Option to complete:** For any reachable marking $m$, it is possible to reach marking $[o]$.
2. **Proper completion:** The only reachable marking that contains a token in place $o$ is the marking $[o]$.
3. **Absence of dead activities:** There are no dead transitions.
Deciding soundness

Theorem:
A workflow net is sound if and only if its short-circuit net is live and bounded.
Sound?

-start register
-send_bill
-receive_payment
-archive
-ship_goods
-check_availability
-replenish
-update
-reminder
-out_of_stock_no_repl
-out_of_stock_repl
-in_stock
-receive_payment
-end
-c1
-c2
-c3
-c4
-c5
-c6
-c7
-c8
-c0
-out_of_stock_no_repl
-out_of_stock_repl
-in_stock
-ship_goods
-archive
-end

Start
-register
-update
-check_availability
-c0

Start
-register
-update
-check_availability
-c0

Start
-register
-update
-check_availability
-c0

Start
-register
-update
-check_availability
-c0
Short-circuit workflow net

The net can deadlock!
Repair
Outline

- What do we want to analyze?
- The reachability graph revised
- Petri net standard properties
- Reachability analysis in CPN Tools
- Workflow nets and soundness
- Model checking
More specific questions

- Is it possible to have a token in both p2 and p5?
- Will t3 always take place?
- Is it possible to execute t1 after t4?
- Can both p4 and p5 be empty at the same time?
General state space analysis (Model Checking)

+ Question

property (in temporal logic)

Model Checker

✗

✓

Counterexample (run that shows the violation)
Model Checking

- Goal: automatically prove whether a model conforms to a specification
- Counterexamples provide valuable insight
- Many success stories in hardware and software verification

- Model checking and thus reachability analysis suffers from the state explosion problem
State explosion problem

- Even small systems have far more reachable states than a computer can handle

\[50^{15} = +/- 30,517,578,125,000,000,000,000,000,000,000\] states
Cause for state explosion: Concurrency

\[3 \times 3 \times 3 + 2 = 29\] markings
Reduction

3 * 3 * 3 + 2 = 27

Check for deadlock freedom:

Reduction possible: Use that all paths from [2,3,4] to [8,9,10] are only permutations

→ Partial order reduction
Cause for state explosion: Data

- Huge color sets result in a huge state space
- Often are color sets even infinite, resulting in an infinite state space

How to deal with?
- Abstract color set to a smaller/finite domain
- Apply other tricks
Cause for state explosion: Unboundedness

Ininitely many reachable states

Finite abstraction (coverability graph)

Next lecture
Simulation vs Verification

- Flexibility
- Analyze a multitude of properties possible
- Time-consuming
- Formal proof
- Applicable for large software systems
- Usage requires experience
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lecturer</th>
<th>Topic</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>3-6-2013</td>
<td>Lect.</td>
<td>Simulation (9)</td>
<td>continuation tasks.</td>
</tr>
<tr>
<td></td>
<td>6-6-2013</td>
<td>Lect.</td>
<td>Reachability Analysis and basic properties (10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6-6-2013</td>
<td>Ass.</td>
<td>Deadline Part I of “CPN assignment”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(23.59)</td>
<td></td>
<td>Conclusion of CPN modeling and Reachability Analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7-6-2013</td>
<td>Inst.</td>
<td>+ Explanation of Part I of the assignment.</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>10-6-2013</td>
<td>Lect.</td>
<td>Coverability and fairness (11)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13-6-2013</td>
<td>Lect.</td>
<td>Structural Analysis and Petri Net Subclasses (12)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Make all exercises in Section 4.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Read Chapter 8.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hand-in assignment in time (see detailed instructions).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Make all exercises in Section 5 and start with exercises in Section 6.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Read Chapter 8 and supplementary material.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Read Chapter 8 and supplementary material.</td>
<td></td>
</tr>
</tbody>
</table>
After this lesson you should be able to

- Explain the difference between qualitative and quantitative questions.
- Calculate the reachability graph of a Petri net and answer questions by inspecting this graph.
- Explain the Petri net standard properties.
- Determine whether a Petri net standard property holds for a given Petri net.
- Explain workflow nets and soundness.
- Explain the state explosion problem and its cause.
- Use CPN Tools to create a state space report.
- Explain the difference between simulation and verification.