Recall reachability analysis

Petri net standard properties

- Boundedness
- Terminating
- Deadlock freedom
- Dead transitions
- Liveness
- Home-markings

General reachability questions

- Marking $m$ reachable?

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 marking $m$ is a home-marking if from any reachable marking we can reach $m$. 

\[
\begin{align*}
&\text{or1} \quad \text{[o1,r2]} \\
&\text{go1} \quad \text{[g1,r2]} \\
&\text{rg1} \\
\end{align*}
\]

\[
\begin{align*}
&\text{or2} \quad \text{[r1,o2]} \\
&\text{rg2} \quad \text{[r1,g2]} \\
\end{align*}
\]

\[
\begin{align*}
&\text{or} \quad \text{[r1,r2,x]} \\
&\text{go2} \\
\end{align*}
\]
When reachability analysis fails: Infinite state spaces

**Idea:**
Construct a **finite** abstraction of the reachability graph that can be used to analyze some properties
Strong point of Petri nets

Math. foundation

Graphical notation

Compactness

Concurrency, locality

Analysis techniques

Tool support
Outline

- The coverability graph
- Fairness
Coverability graph (idea)

• Cannot calculate reachability graph for unbounded Petri nets

• Calculate a **finite** abstraction instead

• Represent **infinite parts** of reachability graph, where some places become unbounded, in a **finite manner**

\[ \omega \text{ denotes that } p_3 \text{ is unbounded} \]
Observation:
• \([p1,p3]\) is greater than \([p1]\);
• therefore, \(\langle t1,t2 \rangle\) can also be fired from \([p1,p3]\)
• resulting markings are the same, except for an additional token on \(p3\)
• by repeatedly executing \(\langle t1,t2 \rangle\), \(p3\) can grow in an unbounded fashion
• denote this by \(\omega\)
A more formal intuition

- There exists an infinite run \( \langle t_1 t_2 t_1 t_2 \ldots \rangle \)
- Only the number of tokens in place \( p_3 \) is increasing
- use that firing rule is **monotonous**: every transition that is enabled at a marking \( m \) can also fire at any marking that contains more tokens than \( m \)
- Denote that \( p_3 \) can grow unbounded by \( \omega \)
- Infinite run becomes **finite**
A multiset \( m_1 \) is greater than or equal to a multiset \( m_2 \) if for all elements \( e \) in \( m_1 \) holds:

\[ m_1(e) \geq m_2(e). \]

Example

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

\[
[p1,p3] \geq [p1] \\
[p2,2p3] \geq [p2,p3] \\
[p1] \not\geq [p2,2p3] \\
[p1,p2] \not\geq [p1,p3]
\]
Mathematics (2): Abstract markings

- Abstract marking $\omega$ denotes that a place is unbounded (i.e., it denotes its limit)
- Extend marking: $m: P \rightarrow \mathbb{N} \cup \{\omega\}$
  with $\omega + n = \omega$ and $\omega - n = \omega$, $n \in \mathbb{N}$.

→ Idea: once unbounded, always unbounded
Constructing the coverability graph

- Calculate the reachability graph
- Apply the following single modification
- For every calculated marking $m'$ and for every marking $m \neq m'$ on a path from the initial marking $m_0$ to $m'$, if $m' \geq m$, then set $m'(p) = \omega$, for all $p \in P$ with $m'(p) > m(p)$
Algorithm

1) Label the initial marking $m_0$ as the root and tag it "new".
2) While "new" markings exists, 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. For every marking $m'' \neq m'$ on a path from the initial marking $m_0$ to $m'$, if $m' \geq m''$, then set $m'(p) = \omega$, for all $p \in P$ with $m'(p) > m''(p)$
      iii. If $m'$ does not appear in the graph add $m'$ and tag it "new".
      iv. Draw an arc with label $t$ from $m$ to $m'$ (if not already present).
3) Output the graph
Example revisited

\[ \text{[p1]} \xrightarrow{t_1} \text{[p2]} \xrightarrow{t_2} \text{[p1, \omega p3]} \xrightarrow{t_1} \text{[p2, \omega p3]} \xrightarrow{t_3} \text{[\omega p3]} \]
Using ProM
Analyze Petri net using ProM
Select analysis

### Behavioral Analysis Plug-in

**Select Analysis Plugins**

To select a plugin, click the name of the plugin on the list below. To deselect it, click again the name of the plugin. If a plugin depends on other plugins, these plugins will automatically be selected.

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Live Transitions</td>
<td>✅</td>
</tr>
<tr>
<td>Unbounded Sequences</td>
<td>✅</td>
</tr>
<tr>
<td>Liveness Info</td>
<td>✅</td>
</tr>
<tr>
<td>Relaxed soundness info (using LeLa for Windows)</td>
<td></td>
</tr>
<tr>
<td>Unbounded Places</td>
<td></td>
</tr>
<tr>
<td>Non-relaxed sound transitions (using LeLa for Windows)</td>
<td></td>
</tr>
<tr>
<td>Home Marking</td>
<td></td>
</tr>
<tr>
<td>Dead Transitions</td>
<td></td>
</tr>
<tr>
<td>Unbounded Info</td>
<td></td>
</tr>
</tbody>
</table>

- ✅: Selected
- ❌: Deselected
Example
Coverability graph

\[ [p1] \xrightarrow{t1} [p2] \xrightarrow{t2} [p1, \omega \ p3] \xrightarrow{t1} [p1, p3] \]
Is the coverability graph unique?

• Construction depends on the transition that we fire first (more than one transition can be enabled)
• So intermediate states are calculated in different order, which may result in different coverability graphs
Example showing non-determinism
CPN Tools and ProM Analysis
Coverability Graph in ProM
Is there a minimal coverability graph?

- The answer is yes:
- Idea: build a tree rather than a graph and give criteria to collapse (not necessarily equivalent) branches on-the-fly
- Remark: different algorithms exist in the literature; differ when to insert an omega and whether a graph or a tree is build
Properties for coverability graphs

1. The reachability graph and the coverability graph of a bounded Petri nets are equivalent.
2. The coverability graph of a Petri net is always finite.
3. A transition $t$ of a Petri net is dead if and only if it does not appear in the coverability graph.
4. A place $p$ of a Petri net is $k$-bounded if and only if $p$ does not contain more than $k$ tokens in any marking of the coverability graph.
5. Every run of a Petri net can be mimicked in the coverability graph (but not the other way around).
Why is the construction always finite? (Sketch)

- Infinite coverability graph implies a run that visits infinitely many states.
- Such a run must contain monotonous parts (Dickson’s Lemma).
- Thus, it contains coverable markings.
- Thus, we could add new omegas ($\omega$).
- As omegas never disappear, and the set of places is finite, we can only add finitely many omegas ($\omega$).
- Thus, construction is finite.
Dickson's Lemma (1874-1954)

Let $P$ be a finite set and let $\varphi_1, \varphi_2, \varphi_3, \ldots$ be an infinite sequence of mappings from $P$ to $\mathbb{N} \cup \{\omega\}$. There exists an infinite sequence of indices $i_1, i_2, i_3, \ldots$ which is strongly monotonic (i.e., $i_1 < i_2 < i_3 < \ldots$) such that, for each $p$ in $P$, $\varphi_{i_1}(p) \leq \varphi_{i_2}(p) \leq \varphi_{i_3}(p) \leq \ldots$.
Properties: boundedness

- A place $p$ of a Petri net is $k$-bounded if and only if $p$ does not contain more than $k$ tokens in any marking of the coverability graph.
• Every run of a Petri net can be mimicked in the coverability graph (but not the other way around).

\[ \langle t_1 \ t_3 \ t_4 \ t_4 \rangle \text{ is a counterexample} \]
\[
\rightarrow \text{ Shows inability to decide liveness}
\]
Reachability graph vs coverability graph (2)

- Several Petri nets may have the same coverability graph (because of the abstraction)
Relation $\omega$-markings and normal markings

Let $b=5$. There is a marking reachable with 1 token in $p_1$ and at least 5 tokens in $p_2$. 

$[p_1, 6 \cdot p_2]$
Example: Coverability graph analysis

- Dead transitions?✗
- Are p1 and p2 bounded?✓
- And p3?✗
- Is [p1,p2] reachable?✗
- Is [p2,5p3] reachable??

Only know: [p2,x⋅p3] reachable and x is at least 5
The coverability graph is finite but ...

- some information gets lost in case of unbounded behavior, and
- it may be huge and impossible to construct.
Restrictions of the coverability graph

- Construct the coverability graph!

```plaintext
color INT = int;
var i:INT;

number = [1]
(counter, (i=1))

number = [2]
(counter, (i=2))

number = [3]
(counter, (i=3))

number = [4]
...  

One token, but different values

Abstraction limited to unboundedness;
Here, data abstraction needed.
Outline

• The coverability graph
• Fairness
The two traffic lights revisited

- There exists a run where the second traffic light never turns green
- Starvation problem
- Can be caused by the abstraction (we abstract from precise scheduling mechanism)
- Analysis should detect this
Distribution of mail in an office

- There exists an infinite run where transition work2 never fires
- This means that empl2 never works
- Run caused by the abstraction (model abstracts from the precise distribution mechanism)
- Analysis should detect this
Motivation

- Abstract model can allow unwanted behavior
- Detect those transitions that cannot fire infinitely often while being enabled infinitely often
- Defined only on infinite runs
Fairness notions

• Transition $t$ is impartial if $t$ occurs infinitely often in every infinite run of the net.

• Transition $t$ is fair if $t$ occurs infinitely often in every infinite run of the net where $t$ is enabled infinitely often.

• Transition $t$ is just if $t$ occurs infinitely often in every infinite run of the net where $t$ is continuously enabled from a marking onward.

• Transition $t$ is not fair if $t$ is not just; that is, there is an infinite run of the net where $t$ is continuously enabled from a marking onward and does not fire any more.
Fairness notions (cont.)

- **Impartial**: considers all infinite runs
- **Fair**: considers those infinite runs where $t$ has to be enabled infinitely often
- **Just**: considers those infinite runs where $t$ has to be continuously enabled infinitely often

Impartial implies fair implies just
Determining fairness

- Consider only **reachable, infinite runs**
- For every transition $t$,
  - Check if $t$ is impartial
    - $t$ impartial $\rightarrow$ done
    - $t$ not impartial (i.e., counterexample exists), check if $t$ is fair
    - $t$ fair $\rightarrow$ done
    - $t$ not fair, then check if $t$ is just
    - $t$ just $\rightarrow$ done
    - else $t$ has no fairness
- Counterexamples can often be used for other transitions or to disprove an even weaker fairness condition
Example (1)

All transitions are **impartial**; that is, they occur infinitely often in every infinite run.
Example (2)

\[ p \in \{ p_1, p_2, p_3, p_4, p_5, p_6 \} \]

\[ t \in \{ t_1, t_2, t_3, t_4, t_5, t_6 \} \]

\[ \begin{align*}
  t_1: & \text{ fair (not impartial:} \\
  & \langle t_1 \ t_3 \ t_2 \ t_3 \ t_2 \ t_3 \ldots \rangle \\
  t_2: & \text{ no fairness (not just:} \\
  & \langle t_1 \ t_3 \ t_5 \ t_4 \ t_5 \ t_4 \ldots \rangle \\
  t_3, t_4, t_5: & \text{ no fairness.} \\
  t_6: & \text{ just (not fair:} \\
  & \langle t_1 \ t_3 \ t_5 \ t_2 \ t_4 \ t_3 \ t_5 \ldots \rangle \\
\end{align*} \]
State Space
Nodes: 5
Arcs: 10
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
Results

Fairness Properties
----------------------------

Impartial Transition Instances
None

Fair Transition Instances
New_Page't1 1

Just Transition Instances
New_Page't6 1

Transition Instances with No Fairness
New_Page't2 1
New_Page't3 1
New_Page't4 1
New_Page't5 1

<table>
<thead>
<tr>
<th>Transition</th>
<th>Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Fair</td>
</tr>
<tr>
<td>t2</td>
<td>No Fairness</td>
</tr>
<tr>
<td>t3</td>
<td>No Fairness</td>
</tr>
<tr>
<td>t4</td>
<td>No Fairness</td>
</tr>
<tr>
<td>t5</td>
<td>No Fairness</td>
</tr>
<tr>
<td>t6</td>
<td>Just</td>
</tr>
</tbody>
</table>
More results

Fairness Properties

- New_Page't1 1 Fair
- New_Page't2 1 No Fairness
- New_Page't3 1 No Fairness
- New_Page't4 1 No Fairness
- New_Page't5 1 No Fairness
- New_Page't6 1 Just
- New_Page't7 1 Fair
Fairness Properties

<table>
<thead>
<tr>
<th>New_Page't1</th>
<th>1</th>
<th>No Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>New_Page't2</td>
<td>1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>New_Page't3</td>
<td>1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>New_Page't4</td>
<td>1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>New_Page't5</td>
<td>1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>New_Page't6</td>
<td>1</td>
<td>No Fairness</td>
</tr>
<tr>
<td><strong>New_Page't7</strong></td>
<td><strong>1</strong></td>
<td>Fair</td>
</tr>
<tr>
<td>New_Page't8</td>
<td>1</td>
<td>No Fairness</td>
</tr>
</tbody>
</table>
Example (3)

- **t1, t2**: impartial

- **t3**: just (not fair: \langle t2 t1 t2 t1... \rangle)

- **t4**: fair, (t4 occurs infinitely often in every infinite run where t4 is enabled infinitely often—there is no such run)
Results CPN Tools

Fairness Properties

Impartial Transition Instances
main't1 1
main't2 1

Fair Transition Instances
main't4 1

Just Transition Instances
main't3 1

Transition Instances with No Fairness
None
Example (4)

t1, t4, t5: fair (not impartial, e.g., for t1 \langle t2 \ t4 \ t2 \ t4... \rangle )

t2, t3: just, (not fair, e.g., for t2 \langle t3 \ t5 \ t3 \ t5... \rangle )
Fairness Properties
-----------------------------------

Impartial Transition Instances
main't3 1

Fair Transition Instances
None

Just Transition Instances
main't1 1
main't2 1

Transition Instances with No Fairness
None
Fairness Properties

Impartial Transition Instances
main't1 1
main't7 1
main't8 1

Fair Transition Instances
None

Just Transition Instances
main't2 1
main't3 1
main't4 1
main't5 1
main't6 1

Transition Instances with No Fairness
None
Can CPN Tools analyze this model? Fair?

t6, t7, t8, t9, and t10 can be continuously enabled without ever firing. Hence, no fairness.

T1 and t5 are impartial

T2 and t4 are just (cannot be continuously enabled)

T3 is fair
CPN Tools stops after preset period/nof states
Fairness?

Impartial Transition Instances
None

Fair Transition Instances
None

Just Transition Instances
None

Transition Instances with No Fairness
Page'Put_Down_Chopsticks1 1
Page'Put_Down_Chopsticks2 1
Page'Put_Down_Chopsticks3 1
Page'Put_Down_Chopsticks4 1
Page'Put_Down_Chopsticks5 1
Page'Take_Chopsticks1 1
Page'Take_Chopsticks2 1
Page'Take_Chopsticks3 1
Page'Take_Chopsticks4 1
Page'Take_Chopsticks5 1
**Exercise!**

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Type</th>
<th>Topic</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>10-6-2013</td>
<td>Lect.</td>
<td>Coverability and fairness (11)</td>
<td>Read Chapter 8 and supplementary material.</td>
</tr>
<tr>
<td></td>
<td>13-6-2013</td>
<td>Lect.</td>
<td>Structural Analysis and Petri Net Subclasses (12)</td>
<td>Read Chapter 8 and supplementary material.</td>
</tr>
<tr>
<td></td>
<td>14-6-2013</td>
<td>Inst.</td>
<td>Reachability, coverability, and net properties.</td>
<td>Make all exercises in Section 6 and Section 7.</td>
</tr>
<tr>
<td>25</td>
<td>17-6-2013</td>
<td>Lect.</td>
<td>Process mining: the Alpha-algorithm (13)</td>
<td>Read Chapter 8 and supplementary material.</td>
</tr>
<tr>
<td></td>
<td>21-6-2013</td>
<td>Inst.</td>
<td>Invariants and process mining</td>
<td>Make all exercises in Section 8 and</td>
</tr>
</tbody>
</table>
After this lecture you should be able to:

• Construct a coverability graph of a Petri net.
• Tell which properties can(not) be derived from the coverability graph.
• Understand the limitations of the coverability graph (e.g., loss of information, inability to decide liveness).
• Derive conclusions from a concrete coverability graph.
• Determine whether a transition is impartial, fair, or just, both by hand and by using CPN Tools.
• Construct nets that have transitions that satisfy certain fairness properties, e.g., a net containing impartial, fair, and just transitions.