## Overview

<table>
<thead>
<tr>
<th>week</th>
<th>date</th>
<th>Type</th>
<th>topic</th>
<th>to prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>22-4-2013</td>
<td>Lect.</td>
<td>Introduction, transition systems, Petri nets (1)</td>
<td>Read Chapters 1-3 of book.</td>
</tr>
<tr>
<td></td>
<td>25-4-2013</td>
<td>Lect.</td>
<td>Petri nets (2)</td>
<td>Read Chapter 3 of book.</td>
</tr>
<tr>
<td></td>
<td>26-4-2013</td>
<td>Inst.</td>
<td>Transition systems, Petri nets</td>
<td>Make all exercises in Section 1 and part of the exercises in Section 2.</td>
</tr>
<tr>
<td>18</td>
<td>29-4-2013</td>
<td>Lect.</td>
<td>TU/e closed</td>
<td>Read Chapter 4 of book.</td>
</tr>
<tr>
<td></td>
<td>3-5-2013</td>
<td>Inst.</td>
<td>Modeling with Petri nets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9-5-2013</td>
<td></td>
<td>TU/e closed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10-5-2013</td>
<td></td>
<td>TU/e closed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17-5-2013</td>
<td>Exam.</td>
<td>Pre-exam focusing on classical Petri nets (1 point)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17-5-2013</td>
<td>Inst.</td>
<td>Explanation &quot;CPN assignment&quot; (3 points)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17-5-2013</td>
<td></td>
<td>TU/e closed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23-5-2013</td>
<td></td>
<td>Modeling in terms of CPN</td>
<td>Make all exercises in Section 3.</td>
</tr>
<tr>
<td></td>
<td>24-5-2013</td>
<td>Inst.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30-5-2013</td>
<td></td>
<td>No lecture</td>
<td>Make all exercises in Section 4.</td>
</tr>
<tr>
<td></td>
<td>31-5-2013</td>
<td></td>
<td>TU/e closed</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>3-6-2013</td>
<td>Lect.</td>
<td>Simulation (9)</td>
<td>Read Chapter 8.</td>
</tr>
<tr>
<td></td>
<td>6-6-2013</td>
<td></td>
<td>Reachability Analysis and basic properties (10)</td>
<td>Read Chapter 8.</td>
</tr>
<tr>
<td></td>
<td>6-6-2013</td>
<td>Ass.</td>
<td>Deadline Part I of &quot;CPN assignment&quot;</td>
<td>Hand-in assignment in time (see detailed instructions).</td>
</tr>
<tr>
<td></td>
<td>6-6-2013</td>
<td></td>
<td></td>
<td>Make all exercises in Section 5 and start with exercises in Section 6.</td>
</tr>
<tr>
<td></td>
<td>7-6-2013</td>
<td>Inst.</td>
<td>Conclusion of CPN modeling and Reachability Analysis and</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></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>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 Section 9.</td>
</tr>
<tr>
<td></td>
<td>23-6-2013</td>
<td>Ass.</td>
<td>Deadline Part II of &quot;CPN assignment&quot;</td>
<td>Hand-in assignment in time (see detailed instructions).</td>
</tr>
<tr>
<td></td>
<td>23-6-2013</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>4-7-2013</td>
<td>Exam.</td>
<td>Final exam (6 points)</td>
<td></td>
</tr>
</tbody>
</table>
Overview

“world”

business processes

people

services

components

organizations

validation

models

analyzes

supports/

controls

records events, e.g.,

messages, transactions,

etc.

verification

(configuration)

(software)

system

e.g., syste

WebSphere,

Oracle,

Staffware,

etc.

(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

analysis

e.g., dedicated formats

such as IBM’s

Common Event
Infrastructure (CEI) and
MXML or proprietary
formats stored in flat files
or database tables.

design-time

analysis

models

analyzes

conformance

extension

process

model

events, e.g.,

messages,

transactions,

etc.

1

2

3

4

5

6

10

11

12

13

(software)

system

supports/

controls

models

analyzes

records events, e.g.,

messages, transactions,

etc.

verification

(validation)

models

analyzes

conformance

extension

design-time

analysis

run-time

analysis

e.g., syste

WebSphere,

Oracle,

Staffware,

etc.

(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.)

(e.g., systems,

etc.)

(process)

model

events, e.g.,

messages,

transactions,

etc.

models

analyzes

conformance

extension

(e.g., systems,

etc.)

(process)

model

events, e.g.,

messages,

transactions,

etc.

models

analyzes

conformance

extension

(e.g., systems,

etc.)

(process)

model

events, e.g.,

messages,

transactions,
Reminder

- The final mark will be based on:
  - A “pre-exam” focusing on classical Petri nets (1 point). This written exam is scheduled on Friday 17-5-2013 from 14.00-15.30 (as part of the regular BIS instruction).
  - A “CPN assignment” (3 points) where a large CPN model is constructed and analyzed. There are two deadlines: 6-6-2013 (Part I) and 23-6-2013 (Part II).
  - A “final exam” (6 points). This written exam is scheduled on Thursday 4-7-2013 from 9.00-12.00.
- The “pre-exam” and “CPN assignment” are mandatory and will expire after the first “final exam” on 4-7-2013.
- The exam in the interim period (15-8-2013) will cover all material and therefore much more difficult to pass.
- Students are advised to not take any risk, and pass the first time (to avoid redoing the entire course).
What's next? Bachelor Level

Bachelor College Choice Coherent:

- **Preparation Master BIS**
  
  mms://TUESTREAM.campus.tue.nl/Videos-STU/preparation-master-bis.wmv
  http://w3.tue.nl/fileadmin/Onderwijs/TU_e_Bachelor_College/Keuzepakketten_PDFs/Preparation_master_BIS.pdf

- **Process Analytics and Systems**
  
  mms://TUESTREAM.campus.tue.nl/Videos-STU/process-analytics-systems.wmv
  http://w3.tue.nl/fileadmin/Onderwijs/TU_e_Bachelor_College/Keuzepakketten_PDFs/Process_analytics_and_systems.pdf

- **Business Process Intelligence (BPI) course**
  
  (2IIE0, Q3)
  
  This course starts with an overview of approaches and technologies that use event data to support decision making and business process (re)design. Then the course focuses on process mining as a bridge between data mining and business process modeling. The course is at an introductory level with various practical assignments.

- **DBL Information Systems**
  
  (2I071, Q4)
  
  The project has 2 phases. 1st phase: understanding the existing process implementation, analyzing problems, developing a plan to solve the problem. 2nd phase: redesigning a process, implementing and testing the designed process. Also focus on professional skills: cooperation, developing solutions of complex problems in groups, presentation and writing documents, reflection.
Master program choice

• Different Master programs BIS, CSE, ES, etc.

• Selection criteria:
  1. What do you like?
  2. What kind of job would you like to have in five years?
  3. Do such jobs exist and are they available?
  4. What are the areas where TU/e excels?
  5. In which group would you like to do a Master project?

• Master of Business Information Systems continues on the topics in this course
Business Information Systems Master

- Computer science is not just about IT systems but also about the processes supported by these systems.
- Interplay between computer science and industrial engineering.
- Most IT-related jobs are in the BIS area.
- TU/e has a strong position in this area (e.g., world-wide leader in areas such as business process management, workflow management, and process mining)
- More information? Ask me!
What's next? Master Level

- **Business Process Management Systems course** (Q3, 2II55)
- **Elective courses** (2013-2014)
  - Process Mining (Q4, 2II66)
  - Business Process Simulation (Q2, 2II75)
  - Meta Modeling and Interoperability (Q2, 2II65)
  - Seminar IS (Q2, 2II96)
- Capita Selecta AIS (2II95, only by invitation)
- Internships (for best students)
- Master projects
Do you know what happens at TU/e?
How much time do you think that academic staff spends on research/teaching?
Science is like sports!

• Competition at various levels: reviews, evaluations, citation indices, funding of projects, H-index, etc.
• Example: H-index based on Google Scholar (see http://scholar.google.com).
Try to look beyond the class room!

- Talk to academic staff and actively think about your future:
  - Which master program?
  - Which specialization?
  - Suitable topic for master project?
  - What kind of job do I want?
  - What kind of job can I get?
Overview AIS

Process Mining

- Process discovery
- Conformance checking
- Simulation
- Verification

PAIS Technology

- BPM/WFM/SOA systems
- Workflow patterns

Process Modeling/Analysis

Perfect score in evaluation of all Dutch Computer Science groups (2010):
5-5-5-5-5
http://tinyurl.com/43h5zfo

Wil van der Aalst

Eindhoven University of Technology

Business Process Management - Process Mining - Petri nets - Information Systems - Business Intelligence

Verified email at tue.nl

http://scholar.google.com
a) **Internal assignments**  
(in areas mentioned before)

b) **External assignments within organizations such as**

- Pallas Athena (NL): process mining, simulation, case handling, and process configuration
- Futura Process Intelligence (NL): process mining and process discovery
- Philips Healthcare (NL): process mining based on event logs of medical devices
- IBM Research (Switzerland/US): workflow patterns and analysis
- IBM Development (Germany/US): case handling and process mining in WebSphere
- SAP AG (Germany/Australia): semantic process mining of ERP systems
- Océ (NL): Petri-net-based modeling and analysis of copiers
- Thales (NL): adapter generation and interface discovery in systems of systems
- IDS Scheer (Germany): process mining and social network analysis
- Academisch Medisch Centrum (NL): workflow management and process mining for hospitals
- ING Group (NL): process redesign and analysis in investment banking
- ILOG/IBM (France): optimization and planning
- Deloitte (NL): IT support for auditing using process mining and process modeling
- Gemeente Harderwijk (NL): process mining and business process modeling
- APG (NL): process mining, workflow management and business process modeling
- PwC (NL): “business process forensics” based on process mining
Master Projects

Architecture of Information Systems
November 2012


this issue
- Research lines P.2
- About the AIS group P.3
- Impact and societal relevance P.4
- Staff involved P.5
- Example projects P.6
- Procedure for Master projects P.8

Are you looking for a Master’s thesis topic that
Interested?

w.m.p.v.d.aalst@tue.nl

+31 40 247.4295/2733

www.vdaalst.com
Old Exam
Business Information Systems
(2II05, 25-1-2010)
Assignment 1 (3.5 points)

- Consider the Brisbane CityCat system, i.e., an urban transportation system using catamarans to quickly move people along the Brisbane River. Let us assume that there are four stops named A, B, C, and D. CityCats move from one stop to the other, first upstream (A,B,C,D) and then downstream (D,C,B,A). There are 10 CityCats. Initially, all CityCats are in a dedicated harbor denoted by X. Depending on the workload, CityCats are put into service (moved from harbor X to stop A) or taken out of service (moved from stop A to harbor X). Note that the number of active CityCats (i.e. in service) may vary between 0 and 10.
CityCats
A CityCat will move as indicated in the figure: e.g., X,A,B,C,D,C,B,A,B,C,D,C,B,A,X or X,A,B,C,D,C,B,A,X or X,A,B,C,D,C,B,A,B,C,D,C,B,A,B,C,D,C,B,A,X, etc. The minimal time (i.e., excluding queuing time) to move from one stop to another is 5 or 10 minutes depending on the type of CityCat. Moreover, each stop takes 5 minutes to allow passengers to embark or disembark. The stops have a capacity of one, i.e., only one CityCat can dock at a particular stop at a time. The capacity of the river is large enough to fit all CityCats in-between any two stops. Note that during rush hours several CityCats may be queuing for the same stop.
Model the Brisbane CityCat system in terms of a classical Petri net (including its initial state). There is no need to distinguish individual CityCats or to model time. (1 point)
A-free

A

X

B

C

D

10 tokens

XtoA

AtoX

to AtoB

from BtoA

AtoB

BtoA

B-up

B-down

B-free

to BtoC

from CtoB

to DtoC

BtoC

from CtoB

from AtoB

from AtoB

from DtoC

from AtoB

from AtoB

from CtoB

from DtoC
b)

• Is the modeled system bounded? If possible, use a place invariant to prove this. (0.25 points)

YES, Invariants:

\[ X + A + A_{\text{to B}} + B_{\text{up}} + B_{\text{to C}} + C_{\text{up}} + C_{\text{to D}} + D + D_{\text{to C}} + C_{\text{down}} + C_{\text{to B}} + B_{\text{down}} + B_{\text{to A}} (= 10) \]
\[ A + A_{\text{free}} (=1) \]
\[ B_{\text{up}} + B_{\text{down}} + B_{\text{free}} (=1) \]
\[ C_{\text{up}} + C_{\text{down}} + C_{\text{free}} (=1) \]
\[ D + D_{\text{free}} (=1) \]

The sum of these semi-positive invariants yields an invariant which assigns a positive weight to all places.

\[ X + 2^*A + A_{\text{to B}} + 2^*B_{\text{up}} + B_{\text{to C}} + 2^*C_{\text{up}} + C_{\text{to D}} + 2^*D + D_{\text{to C}} + 2^*C_{\text{down}} + C_{\text{to B}} + 2^*B_{\text{down}} + B_{\text{to A}} + A_{\text{free}} + B_{\text{free}} + C_{\text{free}} + D_{\text{free}} (= 14) \]

Hence the net is bounded.
\[ X + A + A_{\text{toB}} + B_{\text{up}} + B_{\text{toC}} + C_{\text{up}} + C_{\text{toD}} + D + D_{\text{toC}} + C_{\text{down}} + C_{\text{toB}} + B_{\text{down}} + B_{\text{toA}} \ (= 10) \]

\[ A + A_{\text{free}} \ (=1) \]

\[ B_{\text{up}} + B_{\text{down}} + B_{\text{free}} \ (=1) \]

\[ C_{\text{up}} + C_{\text{down}} + C_{\text{free}} \ (=1) \]

\[ D + D_{\text{free}} \ (=1) \]

\[ X + 2A + A_{\text{toB}} + 2B_{\text{up}} + B_{\text{toC}} + 2C_{\text{up}} + C_{\text{toD}} + 2D + D_{\text{toC}} + 2C_{\text{down}} + C_{\text{toB}} + 2B_{\text{down}} + B_{\text{toA}} + A_{\text{free}} + B_{\text{free}} + C_{\text{free}} + D_{\text{free}} \ (= 14) \]
• Is the system reversible? (0.25 points)
YES, one can always go back to the initial state.

• Is the system live? (0.25 points)
YES, one can always go back to the initial state and from this state one can enable any transition by selecting an appropriate execution path.

• Is the Petri net free-choice? (0.25 points)
NO, for example transitions XtoA and fromBtoA share an input place but have different input sets (respectively \{A-free,X\} and \{A-free,BtoA\}).
Model the Brisbane CityCat system in terms of a colored Petri net (including its initial state). (1.5 points)

Use the CPN notation used in CPN Tools or the notation used in the lecture material. Moreover, take the following aspects into account:

- **Distinguish individual CityCats** (1, 2, … 10).
- There are **two types of CityCats**: slow and fast ones. CityCats 7, 8, 9 and 10 are of a newer generation; they only need 5 minutes to move from one stop to another. The older CityCats (1, 2, … 6) need 10 minutes.
- **CityCats are not allowed to overtake one another!** Therefore, a slower CityCat may slow down a faster one. For example, in the state shown, CityCat 7 cannot overtake CityCat 5 although it is faster. Moreover, there can be a queue of CityCats in front of a stop. In this case, a First-Come-First-Served (FCFS) queuing discipline is used.
- **Upstream CityCats have priority over downstream CityCats!** For example, if there are CityCats queuing for stop B, then the CityCats originating from A (upstream) have priority over CityCats originating from C (downstream). So in the state shown, CityCat 8 has to wait for CityCats 5, 7, and 4. CityCats 5, 7, and 4 are handled in FCFS order. CityCat 2 has to wait for CityCat 1 because CityCat 2 is moving downstream and CityCat 1 is moving upstream.
Options

citycat.cpn

Step: 0
Time: 0
Options
History

Declarations

\text{Standard declarations}
\begin{itemize}
\item colset UNIT = unit;
\item colset INT
\item colset BOOL
\item colset STRING
\item colset CityCat = int with 1..10 timed;
\item colset CCList = list CityCat;
\item fun d(c) = if c<7 then 10 else 5;
\item val ds = 10;
\item var c: CityCat;
\item var l:CCList;
\end{itemize}

Monitors
main
priority
Assignment 2 (1.5 points)

- Consider the following Petri net:
a) Give the coverability graph. (0.50 points)

Different algorithm (see lecture and variants mentioned)

This lecture: no tree, not a unique solution, check with respect to all predecessors!!!
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
b)

- Describe the set of markings that are NOT REACHABLE but that are COVERABLE according to the coverability graph. Motivate your answer using the coverability graph constructed under (a). (0.25 points)

- There are no (non-trivial) markings that are not reachable but coverable according to the coverability graph. Note that any marking which puts a token in p1, p2, or p3 and any number of tokens in p4 is reachable. Of course one can say that [p4] is coverable and not reachable. However, the question aims at situations where the coverability graph over-approximates the set of reachable markings, i.e., suggests markings to be reachable that are not actually reachable because of dependencies between the number of tokens in (un)bounded places or by not allowing any number of tokens on a place (e.g., just an even number). This is not the case here.
• Are all paths in the coverability graph realizable?
  • If not, provide a firing sequence that is not possible in the marked Petri net but that is possible according to the coverability graph.
  • If so, is this a coincidence or not? In other words: Is it always the case that any path in the coverability graph corresponds to a potential firing sequence? Clearly motivate the answer. (0.25 points)

• NO, see for example the sequence t1, t2, t3, t4, t4, t4, t4 which is clearly not possible in the marked Petri net but that is possible according to coverability graph, i.e., the corresponding path exists in the coverability graph.
d)

- Indicate for each transition whether it is impartial, fair, or just (or satisfies no fairness property). (0.50 points)
- $t_1$, $t_2$, and $t_3$ are all impartial because it is not possible to construct an infinite firing sequence where not all of these transitions appear infinitely often. If one stops executing one of these transitions, the system will block after a while. $t_4$ has no fairness as it is possible to construct an infinite firing sequence where $t_4$ remains enabled but never fires.
Assignment 3 (1 point)

- Consider the following CPN taken from the lecture material:

```plaintext
color Product = string;
color Quantity = int;
color PQ = product Product * Quantity;
var p: Product;
var a,b,c,d,e,f: Quantity
val sA = ("productA", 0);
val sB = ("productB", 0);
val oA = ("productA", 0);
val oB = ("productB", 0);
val bA = ("productA", 0);
val bB = ("productB", 0);
val ouA = ("productA", 150);
val ouB = ("productB", 100);
val opA = ("productA", 50);
val opB = ("productB", 60);
```

```
In
Out
In
Out
```

```
\text{customer\_order} \quad \text{PQ} \quad \text{t1} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
\text{repl\_order} \quad \text{PQ} \quad \text{PQ} \quad \text{PQ} \quad \text{PQ} \\
\text{repl\_delivery} \quad \text{PQ} \quad \text{PQ} \quad \text{PQ} \quad \text{PQ} \\
```

```
[b=a] \quad \text{t2} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```

```
[b=a] \quad \text{t3} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```

```
[(c=0) and also (b>=a)] \quad \text{t4} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```

```
[(b+d-c <= f)] \quad \text{t5} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```

```
[b+a] \quad \text{t1} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```

```
[b+d-c <= f] \quad \text{t4} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```

```
[b+a] \quad \text{t5} \\
\text{PQ} \quad (p,a) \quad (p,a) \quad (p,a) \quad (p,a) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,a) \quad (p,c) \quad (p,b) \quad (p,b) \\
\text{PQ} \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \quad (p,b-a) \\
\text{stock} \quad 1\text{`sA}++1\text{`sB} \quad (p,b) \quad (p,b) \\
\text{order\_up\_to\_level} \quad 1\text{`ouA}++1\text{`ouB} \quad \text{order\_point} \quad 1\text{`opA}++1\text{`opB} \\
```
```
In the model it is possible that many orders are backordered because the order point and/or the order up to level are too low. Moreover, the order point and/or the order up to level can also be too high, i.e., there is always an abundance of stock and never any backordering. Therefore, we transform the model into a "learning system". If a customer order can be delivered immediately, then both the order point and the order up to level of the corresponding product are decreased by 1. However, both levels should always be above the minimum value of 5. If a customer order cannot be delivered immediately, then both the order point and the order up to level of the corresponding product are increased by 1. Show the improved CPN model. Clearly describe the changes.
color Product = string;
color Quantity = int;
color PQ = product Product * Quantity;
var p: Product;
var a, b, c, d, e, f: Quantity

Function dcr is defined as follows: 
\[ \text{fun dcr}(x) = \begin{cases} x-1 & \text{if } x > 5 \\
 x & \text{else} \end{cases} \]

Transition t2 has similar connections as t1 however now the inscription on the backward arrows are: (p,e+1)
Assignment 4 (1 point)

• From some transactional system the following event log consisting of seven traces is extracted:

1. a d e f h
2. a e d f h
3. g h
4. a b c d f h
5. a c b d f h
6. a b d c f h
7. a c d b f h
a)

- Derive the $\rightarrow$ relation. (0.25 points)

\[ \begin{align*}
> & = \\
& = \{(a,b),(a,c),(a,d),(a,e),(b,c),(b,d),(b,f),(c,b),(c,d),(c,f),(d,b),(d,c),(d,e),(d,f),(e,d),(e,f),(f,h),(g,h)\} \\
\end{align*} \]

- $\rightarrow$ =

\[ \begin{align*}
& = \{(a,b),(a,c),(a,d),(a,e),(b,f),(c,f),(d,f),(e,f),(f,h),(g,h)\} \\
\end{align*} \]
b) Use the 8 steps of the alpha-algorithm (included on last page) to construct the corresponding Petri-net and draw the Petri-net (delivering all of the intermediate results is not necessary, only the resulting Petri-net is required). (0.5 points)
c) Give a trace possible according to the discovered model but not (yet) observed in the log. (0.25 points)

There are nine possible traces according to the discovered model. Seven are already given as input. The two missing ones are:

1. a d b c f h
2. a d c b f h
Assignment 1 (2 points)

Let us consider a simple production system where raw parts are preprocessed by a machine M1, stored in a temporary buffer, and finally assembled by a second machine M2. There is a single robot R that moves the parts between the input line, M1, the buffer, M2 and the output line. The buffer can hold at most 7 preprocessed items.
Model the production system as a classical Petri net. Include in this classical Petri net model a simple environment that is producing raw parts and consuming finished parts. There is no need to distinguish particular buffer places and parts. The actions by the two machines are not atomic, i.e., the start and the completion of these actions should be distinguishable. The actions of the robot may be considered to be atomic. (1.5 points)
1(a) sol.
1(b)

Also answer the following questions (0.5 points):

i. Can the system deadlock? (system = marked Petri net given under (a))

ii. Is the system bounded?

iii. Is the system reversible?

iv. Is the system live?
1(b) sol.

- Can the system deadlock? (system = marked Petri net given under (a))
  No, in any reachable marking a least one transition is enabled.
- Is the system bounded?
  No, transition *produce* can fire repeatedly putting an unlimited number of tokens in place *input line*.
- Is the system reversible?
  Yes, it is always possible to return to the initial state/marking.
- Is the system live?
  Yes, from any reachable marking and any transition it is possible to reach a state where this transition is enabled.
Consider the following top-level CPN model of a take-away restaurant

colset Customer = string timed;
colset Product = with coffee | tea | beer | fish | chips timed;
colset Products = list Product timed;
colset Order = product Customer * Products timed;
var c:Customer;
var p:Product;
var l:Products;
Chips and Beer

• The environment can place orders by sending a token via place order in to the restaurant subpage. An example of an order is ("John", [fish, chips, beer, beer]) (i.e., one serving of fish, one serving of chips, and two servings of beer for customer John).

• Each incoming order gets an order number to uniquely identify a request. It takes 1 minute to accept an order and to attach a number to it. The acceptance of an order triggers the production of food and drinks. Things are produced in parallel whenever possible and items are not linked to a particular customer order. Nevertheless, only requested items are produced. It takes 2 minutes to prepare a drink (i.e., coffee, tea, or beer) and 3 minutes to prepare food (i.e., one serving of fish or chips). When all items are produced, the customer is called and the items are delivered (i.e., handed over). The delivery takes 1 minute.

• There are 5 people working in the take-away restaurant. There is one person accepting orders and delivering items to the customer. There are two persons preparing drinks and there are two persons preparing food. These people can only do one thing at a time.

• Model the subpage restaurant based on the description above. You will need to introduce additional declarations (e.g., for the order number). Customers orders do not need to be handled in a fixed order (i.e., one customer order can overtake another one) but there should not be unnecessary waiting (i.e., resources are eager to help customers and work in parallel when possible). Use the CPN notation used in CPN Tools or the notation used in the lecture material.
2 sol.

```plaintext
Tool box
Help
Options
cpn2.cpn

Step: 0
Time: 0
Options
History

Declarations

- colset Customer = string timed;
- colset Product = with coffee | tea | beer | fish | chips timed;
- colset Products = list Product timed;
- colset Order = product Customer * Products timed;
- colset OrderNo = int;
- colset Request = product OrderNo * Customer * Products timed;
- colset Resource = with server | drinks | food;

var c:Customer;
var p:Product;
var l:Products;
var ord:Order;
var n:OrderNo;

fun alt(p.[]) = false | alt(p,p2::l) = (p=p2) or else alt(p,l);

Monitors
main

environment
restaurant
```
\begin{itemize}
\item \texttt{colset OrderNo = int;}
\item \texttt{colset Request = product OrderNo * Customer * Products timed;}
\item \texttt{colset Resource = with server | drinks | food;}
\end{itemize}
Assignment 3 (2 points) (only BIS/2II05)
a) Give an initial marking such that the Petri net is live and bounded. Show the
coverability graph. (0.25 points)
b) Give a non-empty initial marking such that the Petri net is bounded but not
live. Show the coverability graph. (0.25 points)
c) Give an initial marking such that the Petri net is unbounded. Show a part of
the coverability tree illustrating that the net is unbounded. (0.25 points).
d) Give a place invariant that shows which places are bounded independent of
the initial marking. (0.25 points)
e) Give a transition invariant that assigns a positive weight to all transitions.
(0.25 points)
f) Give three non-trivial siphons and three non-trivial traps (if possible). (The
empty set and the set of all places are considered trivial.) (0.25 points)
g) One of the Theorems in the paper by Desel and Reisig states that: "If every
proper siphon of a system includes an initially marked trap, then the system is
deadlock free.". Hence a dead marking has a siphon without an initially
marked trap. Provide a dead marking and a siphon for the net above such that
there is no initially marked trap. (0.25 points)
h) Is the net (extended) free choice and/or asymmetric choice? (0.25 points)
3(a)

Give an initial marking such that the Petri net is live and bounded. Show the coverability graph. (0.25 points)

If \( a+d \) is the initial state, then the net is live and bounded.

Note that we do not use the vector notation, i.e., \( a+d \) is a shorthand for \((1,0,0,1,0,0)\).
3(b)

Give a non-empty initial marking such that the Petri net is bounded but not live. Show the coverability graph. (0.25 points)

If a is the initial state, then the net is bounded but not live.

If a+b+d is the initial state, then the net is bounded but not live.
Give an initial marking such that the Petri net is unbounded. Show a part of the coverability tree illustrating that the net is unbounded. (0.25 points).

If \(a+d+f\) is the initial state, then the net is live but not bounded.

Note that we do not use the vector notation, i.e., \(a+\omega b+d\) is a shorthand for \((1,\omega,0,1,0,0)\).

Note again differences in notation and algorithm.
Give a place invariant that shows which places are bounded independent of the initial marking. (0.25 points) a+f and c+d are two invariants showing that the places a,c,d,f are bounded. Of course also a+f+c+d is an invariant showing the same. There is no semi-positive invariant covering b or e.
3(e)

Give a transition invariant that assigns a positive weight to all transitions. (0.25 points)
t1 + t2 + t3 + t4 is a transition invariant
Give three non-trivial siphons and three non-trivial traps (if possible). (The empty set and the set of all places are considered trivial.) (0.25 points)

Siphons: \{a,f\}, \{c,d\}, \{a,c,d,f\}, \{c,d,b\}, \{c,d,e\}, \{a,b,f\}, \{a,e,f\}, ...

Traps: \{a,f\}, \{c,d\}, \{a,c,d,f\}, ...
One of the Theorems in the paper by Desel and Reisig states that: "If every proper siphon of a system includes an initially marked trap, then the system is deadlock free.". Hence a dead marking has a siphon without an initially marked trap. Provide a dead marking and a siphon for the net above such that there is no initially marked trap. (0.25 points) Marking c+d is dead and the siphon \{a,f\} contains as traps the empty set and \{a,f\} which are both unmarked.
Is the net (extended) free choice and/or asymmetric choice? (0.25 points)

The net is not (extended) free choice because a and b share output transition $t_2$ without having identical output sets. However, the net is asymmetric choice.

4) An extended free-choice net (EFC) is an ordinary Petri net such that

$$p_1 \cdot \cap p_2 \cdot \neq \emptyset \implies p_1 \cdot = p_2 \cdot \text{ for all } p_1, p_2 \in P.$$ 

5) An asymmetric choice net (AC) (also known as a simple net) is an ordinary Petri net such that

$$p_1 \cdot \cap p_2 \cdot \neq \emptyset \implies p_1 \cdot \subseteq p_2 \cdot \text{ or } p_1 \cdot \supseteq p_2 \cdot$$

for all $p_1, p_2 \in P.$
Assignment 4 (1 point) (only BIS/2II05)

From an event log of some transactional system the following five traces are extracted:

f,b,g,a,d

f,c,a,h,i,e

f,c,h,a,i,e

f,b,a,g,d

f,c,h,i,a,e
4(a)

Derive the relation

\{(f,b),(f,c),(b,g),(b,a),(c,a),
 (c,h),(g,d),(a,d),(h,i),(a,e),(i,e)\}

- f, b, g, a, d
- f, c, a, h, i, e
- f, c, h, a, i, e
- f, b, a, g, d
- f, c, h, i, a, e
Let $W$ be a workflow log over $T$. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists \sigma \in W \ t = \sigma \}$,
2. $T_I = \{ t \in T \mid \exists \sigma \in W \ t = \text{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists \sigma \in W \ t = \text{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \land A \neq \emptyset \land B \subseteq T_W \land B \neq \emptyset \land \forall a \in A \forall b \in B \ a \rightarrow_W b \land \forall a_1,a_2 \in A \ a_1 \#_W a_2 \land \forall b_1,b_2 \in B \ b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall (A',B') \in X \ A \subseteq A' \land B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{ i_W, o_W \}$,
7. $F_W = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_W \land a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_W \land b \in B \} \cup \{ (i_W,t) \mid t \in T_I \} \cup \{ (t,o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W,T_W,F_W)$.
4(b) sol.

- f → b
- f → c
- b → g
- b → a
- c → a
- c → h
- g → d
- a → d
- a → e
- h → i
- i → e

Diagram:
- b # c
- c # b
- d # e
- e # d
- ...

File Mining Analysis Conversion Reports Window Help
Results: Alpha algorithm plugin on Rave log1...
Suppose that task \( g \) would be invisible, i.e., \( g \) not recorded and the resulting traces are \((f,b,a,d), (f,c,a,h,i,e), (f,c,h,a,i,e), (f,b,a,d), \) and \((f,c,h,i,a,e)\). Is the alpha-algorithm able to discover the resulting process correctly? If so, give the Petri net constructed by the alpha-algorithm. If not, explain why the algorithm fails and what the error in this particular example will be. (0.25 points)

No, see below the resulting net. The problem is that there is a dependency between \( b \) and \( d \) (i.e., the choice for \( d \) is influenced whether \( b \) or \( c \) was selected in the beginning) This is not captured by the > relation.
4(c)

\[ \text{f, b, g, a, d} \]
\[ \text{f, c, a, h, i, e} \]
\[ \text{f, c, h, a, i, e} \]
\[ \text{f, b, a, g, d} \]
\[ \text{f, c, h, i, a, e} \]
Smarter algorithms can tackle this problem easily. See for example the result produced by the alpha++ miner.
color Product = string;
color Quantity = int;
color PQ = product Product * Quantity;
var p: Product;
var a,b,c,d,e,f: Quantity
val sA = ("productA",0);
val sB = ("productB",0);
val oA = ("productA",0);
val oB = ("productB",0);
val bA = ("productA",0);
val bB = ("productB",0);
val ouA = ("productA",150);
val ouB = ("productB",100);
val opA = ("productA",50);
val opB = ("productB",60);

1`sA++1`sB
1`ouA++1`ouB

1`opA++1`opB

1`bA++1`bB

1`a++1`a

In
Out
In
Out

In
Out
In
Out

In
Out
In
Out

In
Out
In
Out

In
Out
In
Out

In
Out
In
Out

In
Out
In
Out

In
Out
In
Out

1`sA++1`sB

[(c=0) andalso (b>=a)]
[b<a]
[b>=a]
In the model it is possible that many orders are backordered because the order point and/or the order up to level are too low. Moreover, the order point and/or the order up to level can also be too high, i.e., there is always an abundance of stock and never any backordering. Therefore, we transform the model into a "learning system". If a customer order can be delivered immediately, then both the order point and the order up to level of the corresponding product are decreased by 1. However, both levels should always be above the minimum value of 5. If a customer order cannot be delivered immediately, then both the order point and the order up to level of the corresponding product are increased by 1. Show the improved CPN model. Clearly describe the changes. (1 point)
Function dcr is defined as follows:

```
fun dcr(x) = if x>5 then x-1 else x;
```

Transition t2 has similar connections as t1 however now the inscription on the backward arrows are: (p,e+1)
Consider again the initial model depicted above. The model allows for backorders and, in principle, backorders may queue for quite some time. Adapt the model such that backorders expire after one month, i.e., if a backorder has not been delivered after one month, it is automatically cancelled. Note that this requires changes to the color sets (e.g., making them timed, etc.). Therefore, clearly list the changes in both the set of declarations and the CPN model. (1 point)
color Product = string;
color Quantity = int;
color OID = int timed;
color PQ = product Product * Quantity;
color IPQ = product OID * PQ;
var p: Product;
var a,b,c,d,e,f: Quantity;
var i: OID;

...
a) Give the reachability graph for this Petri net (0.5 point).

b) Give two non-trivial place invariants (0.25 points)

c) Give a non-trivial transition invariant (0.25 points)

Not representative: too simple!!!
a) Give the reachability graph for this Petri net (0.5 point).

b) Give two non-trivial place invariants (0.25 points)

\[ p_1 + p_2 + p_3 + p_4 \quad \text{and} \quad p_1 + p_4 + p_5 + p_6 \]

c) Give a non-trivial transition invariant (0.25 points)

\[ t_3 + t_4 \quad \text{or} \quad t_1 + t_2 + t_3 + t_4 + t_5 + t_6 \]
Questions
### Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecturer/Instructor</th>
<th>Topic</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>14-6-2013</td>
<td>Inst.</td>
<td>Read Chapters 1-2 of book.</td>
<td>Make all exercises in Section 1 and Section 2.</td>
</tr>
<tr>
<td>25</td>
<td>Lect.</td>
<td>Process mining: the Alpha-algorithm (13)</td>
<td>Read Chapter 8 and supplementary material.</td>
</tr>
<tr>
<td>21-6-2013</td>
<td>Lect.</td>
<td>Invariants and process mining</td>
<td>Make all exercises in Section 2.</td>
</tr>
<tr>
<td>23-6-2013</td>
<td>Ass.</td>
<td>Deadline Part II of “CPN assignment”</td>
<td>Hand-in assignment must be completed in time (see detailed instructions).</td>
</tr>
<tr>
<td>27</td>
<td>Exam</td>
<td>Final exam (6 points)</td>
<td>Good Luck!</td>
</tr>
</tbody>
</table>

- **Subclasses (12)**
- Reachability, coverability, and net properties.

**Notes:**
- Study old BIS exams.
- Make all exercises in Section 5 and Section 6.
- Start making exercises in Section 3.
- Read Chapters 1-2 of book.
- Read C Section 4.
- Make all exercises in Section 6 of book.
- Read C Section 3 of book.
- Make all exercises in Section 3 of book.
- Read C Section 2 of book.
- Make all exercises in Section 2 of book.
- Read C Section 1 of book.