Solutions Selected Exercises BPMS

Note that all exercises are important!
This selection does not cover all material relevant for the exam.

prof.dr.ir. Wil van der Aalst
Currently, many hospitals are investigating the introduction of workflow management systems to support patient diagnosis and treatment processes. Clearly, there are many complications compared to classical administrative processes. First of all, each patient is unique and given the many possible conditions and diseases there are many interacting processes. Second, there is a need for flexibility. Therefore, currently workflow management systems are only used to support relatively structured processes such as radiology, lab testing, etc. However, as workflow management systems become more powerful and flexible and patient data becomes electronic, workflow technology will become more applicable in hospitals.
In this assignment we consider the fictive *diagnosis process of patients having a severe cough*. Patient having a chronic cough (i.e., a cough that lasts for at least 3 weeks) consult a specialist of the daycare center in the hospital.

First an appointment is made via the specialist's secretary. This is followed by the patient actually visiting the specialist in the hospital (daycare center).

The specialist records the symptoms and decides that (1) there is no need to worry and the process ends here, (2) the patient should make a new appointment to visit the specialist in four weeks, or (3) the symptoms require an X-ray to be made. If the patient is invited to make a new appointment, then based on the subsequent visit the same three possible continuations are possible, etc. Moreover, the patient may not bother to make a new appointment. In the later case, the process ends if there is not a new appointment in 3 months.
• If the specialist decides that an X-ray needs to be made, the patient is referred to radiology where the X-ray photograph is taken by a radiologist. The X-ray is sent to the specialist. Based on the X-ray, the specialist may decide to do more tests: (1) another X-ray, (2) a blood test, and/or (3) a CT-scan, e.g., the specialist may order a blood test and a CT scan but not another X-ray.

• The X-ray is made by a radiologist of the X-ray department. The blood test consists of two steps: taking the blood by a nurse and doing the actual test by a laboratory assistant (both from the test department). The CT-scan is done by a radiologist of the CT department.
• Each test result is communicated to the specialist, but multiple tests can be done in parallel. Each test result is evaluated by the specialist separately, i.e., the specialist does not wait until all test results are available because this could put the patient in danger. If a test result indicates a serious problem that needs to be addressed immediately, an emergency treatment is started without waiting for the other test results.

• Note that multiple emergency treatments may be started (i.e., each time a serious problem surfaces). If none of the test results indicates such a problem, the normal treatment process is started. The diagnosis process ends if the necessary emergency treatment(s) or normal treatment have/has been started.
Model the above workflow diagnosis process using the notation of the book, i.e., a process definition (including triggers) and a resource classification. Make sure that the process definition is sound, i.e., it is always possible to end properly (token in final place) and there are no dead parts/dangling tokens. Test the process model using some typical scenarios to make sure that it is always possible to terminate properly!

- Model the above workflow diagnosis process using the notation of the book, i.e., depict a process definition (including triggers) which is sound. It can help to check the process model using some typical scenarios to make sure that it is always possible to terminate properly!

- Depict the resource classification using the notation of the book.
Roles:
Secr = secretary
Sp = specialist
Rad = Radiologist
Nurse = nurse
LA = laboratory assistant

Groups (organizational units):
H = hospital
XRD = X-ray department
CTD = CT-scan department
DDC = daycare center
TD = test department
(Note that we do not consider the patient to be a resource in this solution.)

Roles:
Secr = secretary
Sp = specialist
Rad = Radiologist
Nurse = nurse
LA = laboratory assistant

Groups (organizational units)
H = hospital
XRD = X-ray department
CTD = CT-scan department
DDC = daycare center
TD = test department
Roles:
Secr = secretary
Sp = specialist
Rad = Radiologist
Nurse = nurse
LA = laboratory assistant

Groups (organizational units):
H = hospital
XRD = X-ray department
CTD = CT-scan department
DDC = daycare center
TD = test department
Q
• What patterns does the process model contain?
• Modify the process model so that the pay bill task can only execute when the process is in state c5
• What pattern does this modification require?
• What patterns are difficult/impossible to capture with workflow nets?
• What patterns does the process model contain?
• Modify the process model so that the pay bill task can only execute when the process is in state c5

• What pattern does this modification require?

milestone

• What patterns are difficult/impossible to capture with workflow nets?
  • partial join variants
  • cancellation
  • multiple instance
  • general synchronising merge
  • Etc.
The illustration below shows the nominal form of the milestone pattern. The difficulty with this form is that if the thread of control has passed place milestone, then task c will never be enabled anymore. Modify the model so that branch 2 does not block.

Also provide another example of a fundamentally different milestone.

Show a “fake” milestone.
Instruction answer – 7a

b can only remove token from m after c occurred

avoid adding fake milestones
deferred choice

d often has a trigger (e.g., time trigger)

after b occurred d can be executed instead of c
fake milestone!
Instruction question – 13

In a workflow system such as YAWL, active work items can be in an offered, allocated or started state. The transition to each state can be initiated by the system or the resource. The triple (offered by, allocated by, started by) is termed the interaction sequence for a task.

Explain the events associated with the following interaction sequences:

- SSR
- SRS
- SRR
- SSS
- RSS
- RSR
- RRS
- RRR

How are each of them configured in YAWL?

Which patterns do they correspond to?
## Instruction answer – 13a

<table>
<thead>
<tr>
<th>Offer</th>
<th>Alloc</th>
<th>Start</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>R</td>
<td>R</td>
<td>The system offers the workitem to one or more participants. One participant self-allocates the workitem; it is withdrawn from the other offered participants. The participant manually starts the workitem.</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>R</td>
<td>The system offers then immediately allocates the workitem to a participant. The participant manually starts the workitem.</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>S</td>
<td>The system offers the workitem to one or more participants. One participant self-allocates the workitem; it is withdrawn from the other offered participants. The workitem immediately starts.</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>The system offers and allocates the workitem to one participant in a started state.</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>An administrator offers the workitem to one or more participants. One participant self-allocates the workitem and it is withdrawn from the other offered participants. The participant manually starts the workitem.</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>R</td>
<td>An administrator chooses one or more participants to offer the workitem to. One participant from those offered is immediately allocated the workitem. The participant manually starts the workitem.</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>S</td>
<td>An administrator offers the workitem to one or more participants. One participant self-allocates the workitem and it is withdrawn from the other offered participants. The workitem starts immediately.</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>S</td>
<td>An administrator chooses one or more participants to offer the workitem to. One participant from those offered is immediately allocated the workitem. The workitem immediately starts.</td>
</tr>
</tbody>
</table>
Step 1 : Choose Behaviour At Interaction Points

There are three key decision points for managing the resourcing of work items spawned from a task. At each of these interaction points, you may choose to have the system dynamically make a decision on resourcing at each point, or alternatively, allow a user to manually make each decision. Each interaction point is briefly described below:

- **Offer**: The point at which it is decided that a number of participants could undertake the work item.
- **Allocation**: The point at which one of the participants offered the work item is nominated to do that work item.
- **Start**: The point at which the participant allocated a work item begins working on it.

Offering a work item for this task to a number of participants is to be done by: ☐ User ☐ System

Allocating a work item for this task to one of the offered participants is to be done by: ☐ User ☐ System

Starting an allocated work item of this task is to be done by: ☐ User ☐ System
Instruction answer – 13c

- Created
  - Offered to a single resource
  - Offered to multiple resources
    - Suspended
      - Started
        - Completed
        - Failed
  - Allocated to a single resource
    - S:offer-s
    - R:allocate-s
    - S:allocate
    - S:create
    - R:allocate-m
    - R:start-s
    - R:start
    - R:start-m
    - R:resume
    - R:complete
    - R:suspend
    - R:fail
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Distribution Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRR</td>
<td>Distribution by Offer – Single/Multiple Resource</td>
<td>Resource-Initiated Allocation</td>
</tr>
<tr>
<td>SSR</td>
<td>Distribution by Allocation – Single Resource</td>
<td>Resource-Initiated Execution – Allocated Work Item</td>
</tr>
<tr>
<td>SRS</td>
<td>Distribution by Allocation – Single Resource</td>
<td>Resource-Initiated Allocation</td>
</tr>
<tr>
<td>SSS</td>
<td>Distribution by Allocation – Single Resource</td>
<td>Commencement on Creation</td>
</tr>
<tr>
<td>RRR</td>
<td>Distribution by Offer – Single/Multiple Resource</td>
<td>Resource-Initiated Allocation</td>
</tr>
<tr>
<td>RSR</td>
<td>Distribution by Allocation – Single Resource</td>
<td>Resource-Initiated Execution – Allocated Work Item</td>
</tr>
<tr>
<td>RRS</td>
<td>Distribution by Allocation – Single Resource</td>
<td>Resource-Initiated Allocation</td>
</tr>
<tr>
<td>RSS</td>
<td>Distribution by Allocation – Single Resource</td>
<td>Commencement on Creation</td>
</tr>
</tbody>
</table>
With the aid of a task lifecycle diagram, explain the difference between the *delegation* and *stateful reallocation* patterns.

How are each of them configured in YAWL?

What user interactions are associated with each of them?
Instruction answer – 15b

Manage Resourcing Wizard for Atomic Task “Approve Purchase Order”

Step 5: Establish Default User Runtime Privileges for this Task

- Can a participant suspend a started work item of this task?  ○ No  ○ Yes
- Can a participant reallocate a work item of this task to another participant, resetting state?  ○ No  ○ Yes
- Can a participant reallocate a work item of this task to another participant, retaining state?  ○ No  ○ Yes
- Can a participant deallocate themselves from a work item of this task?  ○ No  ○ Yes
- Can a participant delegate a work item of this task to another participant?  ○ No  ○ Yes
- Can a participant skip a work item of this task?  ○ No  ○ Yes
Instruction answer – 15c

A work item is displayed with details:
- **Work Items**: 1.1:Approve_Purchase_Order_1901
- **Specification**: OrderFulfilment
- **Task**: Approve Purchase Order
- **Case**: 1.1
- **Status**: Enabled
- **Created**: Mar.27, 2009 18:04:27
- **Age**: 0:00:10:33

A red box highlights the `Delegate` button.
Instruction answer – 15d
Theorem 1 (Soundness, liveness and boundedness). Let $N$ be a WF-net without reset and/or inhibitor arcs. $N$ is sound if and only if $(\overline{N}, [i])$ is live and bounded.

Provide (if possible):

a) A WF-net $N$ without inhibitor arcs such that $N$ is sound and its short-circuited counterpart is unbounded.

b) A WF-net $N$ without reset arcs such that $N$ is sound and its short-circuited counterpart is unbounded.

c) A WF-net $N$ without inhibitor arcs such that $N$ is sound and its short-circuited counterpart is non-live.

d) A WF-net $N$ without reset arcs such that $N$ is sound and its short-circuited counterpart is non-live.
Solution 4 (a)

A WF-net without inhibitor arcs such that $N$ is sound and its short-circuited counterpart is unbounded
Solution 4 (b)

A WF-net without reset arcs such that $N$ is sound and its short-circuited counterpart is unbounded.
A WF-net $N$ without inhibitor arcs such that $N$ is sound and its short-circuited counterpart is non-live.

A WF-net $N$ without reset arcs such that $N$ is sound and its short-circuited counterpart is non-live.

Both are NOT possible as shown by the following Lemma:

**Lemma 1** (Soundness implies liveness). Let $N$ be a WF-net. If $N$ is sound, then $(\overline{N},[i])$ is live.
Provide (if possible):

a) A WF-net N without inhibitor arcs such that N is not sound and its short-circuited counterpart is live and bounded.

b) A WF-net N without reset arcs such that N is not sound and its short-circuited counterpart is live and bounded.
Solution 5 (a)

A WF-net $N$ without inhibitor arcs such that $N$ is not sound and its short-circuited counterpart is live and bounded.

or
A WF-net $N$ without reset arcs such that $N$ is not sound and its short-circuited counterpart is live and bounded.
Exercise 6

Proposition 1 (Proper completion is implied). Let $N$ be a WF-net. The “option to complete” implies “proper completion”, i.e., $(\forall M \in R(N,[i]) \ [o] \in R(N,M) \Rightarrow (\forall M \in R(N,[i]) \ (M \geq [o]) \Rightarrow (M = [o])))$.

- Is the reverse also true ("proper completion" implies "option to complete")?
- If yes, provide proof.
- If no, give counter-example.
• Not true, i.e., "proper completion" does NOT imply "option to complete".
• Counter example:

Proper completion because if p4 has a token the rest is empty. After firing t3 there is no option to complete.
Definition 2 (Extended Petri net). An extended Petri net is \( W, A, L, R, H \), where:

- \( (P, T, F) \) is a basic Petri net,
- \( W \in F \cup N \setminus \{0\} \) is an (arc) weight function,
- \( A \subseteq A \) is a set of (activity) labels,
- \( L \in T \to A \cup \{\tau\} \) is a labeling function,
- \( R \in T \to 2^P \) is a function defining reset arcs, and
- \( H \in T \to 2^P \) is a function defining inhibitor arcs.

- Provide a general (i.e., also incorporate reset and inhibitor arcs) translation to map any WF-net with arc weights onto an equivalent WF-net without arc weights (i.e., \( W(x,y)=1 \)).
Solution 8

Basic idea for removing up-to-k weighted arcs:

(a) net with arc weights

(b) net without arc weights
Exercise 9 (a)

• Consider the two WF-nets.
• They can be composed into one WF-net by connecting the interface places and by adding a "start" and "end" place and transition.
• Is the composed WF-net sound?
Solution 9 (a)

- See the composed WF-net on the right.
- This composed WF-net is sound.
Exercise 9 (b)

- Consider the two WF-nets.
- They can be composed into one WF-net by connecting the interface places and by adding a "start" and "end" place and transition.
- Is the composed WF-net sound?
Solution 9 (b)

- The composed WF-net can be constructed as before.
- This composed WF-net is not sound due to a deadlock.
Exercise 9 (c)

- Consider the two WF-nets.
- They can be composed into one WF-net by connecting the interface places and by adding a "start" and "end" place and transition.
- Is the composed WF-net sound?
Solution 9 (c)

- The composed WF-net can be constructed as before.
- This composed WF-net is not sound due to a non-communicated internal choice.
Exercise 9 (d)

- Consider the two WF-nets.
- They can be composed into one WF-net by connecting the interface places and by adding a "start" and "end" place and transition.
- Is the composed WF-net sound?
Solution 9(d)

- The composed WF-net can be constructed as before.
- This composed WF-net is sound.
Exercise 1

- Remove reset arcs, i.e., construct an equivalent WF-net without reset arcs (possibly with inhibitor arcs)
Use construction:
Result of brute force:
Exercise 5

- Provide coverability graph.
- Is the WF-net:
  - a state machine?
  - a marked graph?
  - a free-choice net?
  - well-structured?
  - sound?
Solution 5

coverability graph
• Is the WF-net:
  – a state machine? NO
  – a marked graph? NO
  – a free-choice net? YES
  – well-structured? NO

– sound? NO
Consider the incomplete WF-net shown above. The workflow always starts with A and ends with E. In the model tasks B, C and D can be executed multiple times and in any order. However, there is one requirement: tasks B and D exclude one another. If B occurs, D is not allowed to occur and if D occurs, B is not allowed to occur. Hence ABCBABCBE and ADDDCCDCE are possible executions but ABCDE is not. Because B, C, and D are potentially concurrent, make sure that all choices are deferred choices. The model should be sound and use the book notation.

Is the WF-net:
- a state machine?
- a marked graph?
- a free-choice net?
- well-structured?
Is the WF-net:
- a state machine? NO
- a marked graph? NO
- a free-choice net? NO
- well-structured? NO
Exercise 7

Is the WF-net:
- a state machine?
- a marked graph?
- a free-choice net?
- well-structured?
- sound?
Solution 7

Is the WF-net:
- a state machine? NO
- a marked graph? NO
- a free-choice net? YES
- well-structured? NO
- sound? YES
Exercise 3

\[ L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}] \]

- Construct two transition systems for event log \( L_9 \)
  - One based on past with no abstractions (prefix automaton)
  - One based on the last event only.
- Compute all minimal non-trivial regions for both transition systems.
- Show the corresponding Petri nets.
- Check results using ProM.
Prefix automaton

\[ L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}] \]
Regions
Transition system based on last event only

\[ L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}] \]
Regions and Petri net
Past with no abstractions
(note optimization wrt to merging final states)
Based on the last event only
Exercise 4

\[ L_6 = \left[ \langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4 \right] \]

- Construct a transition system based on past with no abstractions (prefix automaton) for event log \( L_6 \).
- Compute all minimal non-trivial regions.
- Show the corresponding Petri net.
- Check result using ProM (also explore other abstractions).
\[ L_6 = [\langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4] \]
Some regions
Additional non-trivial minimal regions
Petri net

Note redundant places!
Even more minimal regions?

- These would be added when using the basic algorithm.
- However, they can be discarded because they have no exiting arcs and are therefore not limiting the behavior.
- ProM does not show these places.
Q
Exercise 1

• Consider an event log containing the following traces (frequencies are not given):
  a b c d b c d b c e
  a b c d b c d c b e
  a b c d c b d b c e
  a b c d c b d c b e
  a c b d b c d b c e
  a c b d b c d c b e
  a c b d c b d b c e
  a c b d c b d c b e

• Construct the transition system using a multi set abstraction based on past events.

• Determine the non-trivial minimal regions.

• Construct a Petri net using these regions.

• Show whether the given transition system is elementary or not.

• Diagnose differences between the transition system and Petri net.
Transition System and Minimal Regions

Answer
Exercise 1
Petri net based on non-trivial minimal regions

Exercise 1
The transition system is not elementary!

- \(\text{GER}(a) = \{s1\}\)
- \(\text{GER}(b) = \{s2,s4,s6,s8,s10,s12\}\)
- \(\text{GER}(c) = \{s2,s3,s6,s7,s10,s11\}\)
- \(\text{GER}(d) = \{s5,s9\}\)
- \(\text{GER}(e) = \{s13\}\)
- \(\text{pre}(a) = \{\{s1\}\}\)
- \(\text{pre}(b) = \{\{s2,s4,s6,s8,s10,s12\}\}\)
- \(\text{pre}(c) = \{\{s2,s3,s6,s7,s10,s11\}\}\)
- \(\text{pre}(d) = \{\{s4,s5,s8,s9,s12,s13\},\{s3,s5,s7,s9,s11,s13\}\}\)
- \(\text{pre}(e) = \{\{s4,s5,s8,s9,s12,s13\},\{s3,s5,s7,s9,s11,s13\}\}\)

- \(\text{GER}(a) = \{s1\} = \cap \text{pre}(a)\)
- \(\text{GER}(b) = \{s2,s4,s6,s8,s10,s12\} = \cap \text{pre}(b)\)
- \(\text{GER}(c) = \{s2,s3,s6,s7,s10,s11\} = \cap \text{pre}(c)\)
- \(\text{GER}(d) = \{s5,s9\} \neq \{s5,s9,s13\} = \cap \text{pre}(d)\)
- \(\text{GER}(e) = \{s13\} \neq \{s5,s9,s13\} = \cap \text{pre}(e)\)

The transition system is not elementary because forward closure does not hold.
Understanding the differences

• $\text{GER}(d) = \{s5,s9\} \neq \{s5,s9,s13\} = \cap \text{pre}(d)$
• $\text{GER}(e) = \{s13\} \neq \{s5,s9,s13\} = \cap \text{pre}(e)$

Traces such as $abce$, $acbdbce$, and $abcdbcdbcdbcdbcdbce$ are possible in the Petri net but not in the transition system!
Exercise 2

- Assume the transition system is derived from some event log.
- Determine the non-trivial minimal regions.
- Construct a Petri net using these regions.
- Show whether the given transition system is elementary or not.
- Diagnose differences between the transition system and Petri net.
Minimal non-trivial regions

Answer
Exercise 2
Petri net constructed using the minimal regions

Answer
Exercise 2
Elementary Transition System?

All simple requirements hold:
• all states are reachable from initial state,
• no self loops,
• transitions signal choices.
Remains to check:
• state separation (states cannot be identical wrt regions), and
• forward closure (regions should not allow for more behavior than the transition system).
• GER(a) = {s1}
• GER(b) = {s2,s4,s5,s8}
• GER(c) = {s2,s3,s5,s7}
• GER(d) = {s2,s3,s4,s6}
• GER(e) = {s5,s7,s8,s9}
• GER(f) = {s9}

• pre(a) = {{s1}}
• pre(b) = {{s2,s4,s5,s8}}
• pre(c) = {{s2,s3,s5,s7}}
• pre(d) = {{s2,s3,s4,s6}}
• pre(e) = {{s5,s7,s8,s9}}
• pre(f) = {{s3,s6,s7,s9},{s4,s6,s8,s9},{s5,s7,s8,s9}}

• GER(a) = {s1} = ∩ pre(a)
• GER(b) = {s2,s4,s5,s8} = ∩ pre(b)
• GER(c) = {s2,s3,s5,s7} = ∩ pre(c)
• GER(d) = {s2,s3,s4,s6} = ∩ pre(d)
• GER(e) = {s5,s7,s8,s9} = ∩ pre(e)
• GER(f) = {s9} = ∩ pre(f)
State separation

For every pair of different states we can find a region where one is in and the other is out.

<table>
<thead>
<tr>
<th></th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>s4</th>
<th>s5</th>
<th>s6</th>
<th>s7</th>
<th>s8</th>
<th>s9</th>
<th>s10</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Answer

Exercise 2
The transition system is elementary because all conditions hold.

No differences in behavior: bisimilar.
Exercise 5:
Compute Fitness Using Missing and Remaining Tokens

Consider the event log containing 33 cases.
What is the fitness?
Fitness = 0.895705521

<table>
<thead>
<tr>
<th>trace</th>
<th>frequency</th>
<th>produced tokens (p)</th>
<th>remaining tokens (r)</th>
<th>consumed tokens (c)</th>
<th>missing tokens (m)</th>
<th>produced tokens (all)</th>
<th>remaining tokens (all)</th>
<th>consumed tokens (all)</th>
<th>missing tokens (all)</th>
</tr>
</thead>
<tbody>
<tr>
<td>acd</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>bce</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>ace</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>25</td>
<td>5</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>bcd</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>25</td>
<td>5</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>dca</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>abd</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sum p</th>
<th>sum r</th>
<th>sum c</th>
<th>sum m</th>
<th>fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>17</td>
<td>163</td>
<td>17</td>
<td>0.895705521</td>
</tr>
</tbody>
</table>

Answer
Exercise 5
ProM 5.2 output

fitness of 0.8957

- total of 17=1+3+6+6+1 remaining tokens
- total of 17=1+1+7+5+3 missing tokens

Answer Exercise 5
Exercise 6: Compute Fitness Using Missing and Remaining Tokens

Consider the event log containing 20 cases.

- What is the fitness?

<table>
<thead>
<tr>
<th>trace</th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>abefcda</td>
<td>10</td>
</tr>
<tr>
<td>abbefcccd</td>
<td>10</td>
</tr>
</tbody>
</table>
Fitness = 0.8

<table>
<thead>
<tr>
<th>trace</th>
<th>frequency</th>
<th>produced tokens (p)</th>
<th>remaining tokens (r)</th>
<th>consumed tokens (c)</th>
<th>missing tokens (m)</th>
<th>produced tokens (all)</th>
<th>remaining tokens (all)</th>
<th>consumed tokens (all)</th>
<th>missing tokens (all)</th>
</tr>
</thead>
<tbody>
<tr>
<td>abefcd</td>
<td>10</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>90</td>
<td>20</td>
<td>90</td>
<td>20</td>
</tr>
<tr>
<td>abbefcdd</td>
<td>10</td>
<td>11</td>
<td>2</td>
<td>11</td>
<td>2</td>
<td>110</td>
<td>20</td>
<td>110</td>
<td>20</td>
</tr>
</tbody>
</table>

200
200
40
40

sum p
sum r
sum c
sum m

fitness 0.8

Answer Exercise 6
Answer

Exercise 6

ProM 5.2 output

- total of 40 = 10 + 10 + 20 remaining tokens
- fitness of 0.8
- total of 40 = 20 + 10 + 10 missing tokens
Question 5: How many valid configurations?
Solution 5: 6 valid configurations

- f cannot be blocked: hiding of f is independent of rest
- e or (c and d) need to be activated
- c is activated if and only if d is activated
- hence, $2 \times 3 = 6$ valid configurations
Example of a valid configuration
Example of an invalid configuration
Minimal requirement: Create a configurable WF-net such that each variant corresponds to a valid configurations.
Solution 6
Question 7: Three event logs L1-L3

L1 = \[<a,b,c,d,e>^{20}, <a,b,d,c,e>^{30}, <a,c,b,d,e>^{10}, <a,c,d,b,e>^{20}, <a,d,b,c,e>^{30}, <a,d,c,b,e>^{10}] \]
L2 = \[<a,c,d,e>^{20}, <a,d,c,e>^{30}] \]
L3 = \[<f,g>^{20}] \]

- Construct three process models for these event logs using the *multiset abstraction* (based on past) and state-based regions.
- Create a configurable WF-net such that each of the three variants corresponds to a valid configuration.
Solution: Regions and Petri net for L1

L1 = \[<a,b,c,d,e>^{20},<a,b,d,c,e>^{30},<a,c,b,d,e>^{10},<a,c,d,b,e>^{20},<a,d,b,c,e>^{30},<a,d,c,b,e>^{10}>\]
Regions and Petri net for L2

\[ L_2 = [<a,c,d,e>^{20}, <a,d,c,e>^{30}>] \]
Regions and Petri net for L3

L3 = [<f,g>^{20}]
Merge models into a configurable model
Practise!