8. Behavioural equivalences

David Pereira José Proença

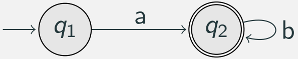
RAMDE 2021/2022

Requirements and Model-driven Engineering

CISTER – ISEP
Porto, Portugal

https://cister-labs.github.io/ramde2122

Recall

1. Non-deterministic Finite Automata: 

2. Process algebra: $P = a.Q \quad Q = b.Q \quad P|Q$
3. Interaction between processes
4. Meaning of PA using NFA

Still missing

- When is a process P equivalent to a process Q?
- When can a process P be safely replaced by a process Q?
- When can a sequence of interactions be safely implemented as interacting components?

Syllabus

- High-level overview or requirements and associated processes
- Mathematical Preliminaries
  - Basic mathematical notations
  - Set theory
  - PropositionalLogic
  - First Order Logic
- Behavioural modelling
  - Single component
  - Many components
  - Equivalences
    - Language Equivalence
    - (Bi)similarity
    - Realisability
  - Verification

Behavioural Equivalences – Intuition

Two automata (or LTS) should be equivalent if they cannot be distinguished by interacting with them.

Equality of functional behaviour

is not preserved by parallel composition: non compositional semantics, cf,

`x:=4; x:=x+1` and `x:=5`

Graph isomorphism

is too strong (why?)

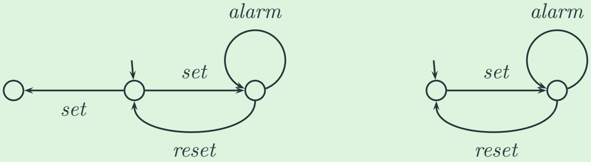
EQ1 – Language equivalence

Language equivalence

Definition

Two automata A, B are language equivalent iff $L_A = L_B$
(i.e. if they can perform the same finite sequences of transitions)

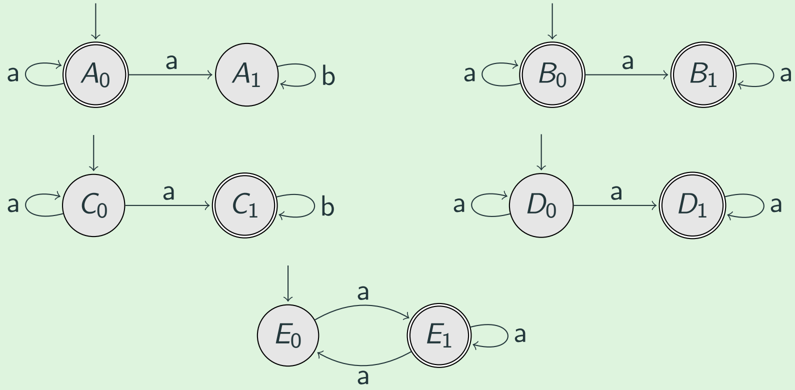
Example



Language equivalence applies when one can neither interact with a system, nor distinguish a slow system from one that has come to a stand still.

Exercise

Ex. 8.1: Find pairs of automata with the same language



EQ2 – Similarity

Simulation

the quest for a behavioural equality:

able to identify states that cannot be distinguished by any realistic form of observation

Simulation

A state q simulates another state p if
every transition from q is corresponded by a transition from p and
this capacity is kept along the whole life of the system to which state space q belongs to.

Definition

Given $\langle S_1, N, \longrightarrow_1 \rangle$ and $\langle S_2, N, \longrightarrow_2 \rangle$ over N (ignoring initial and final states) a relation $R \subseteq S_1 \times S_2$ is a simulation iff, for all $\langle p, q \rangle \in R$ and $a \in N$,

$$(1) \quad p \xrightarrow{a}_1 p' \;\Rightarrow\; \langle \exists\, q' \;:\; q' \in S_2 : \; q \xrightarrow{a}_2 q' \wedge \langle p', q' \rangle \in R \rangle$$

$$\begin{tikzcd} p \arrow[d, "a"] & R & q \\ p' & & \end{tikzcd} \qquad \Longrightarrow \qquad \begin{tikzcd} & & q \arrow[d, "a"] \\ p' & R & q' \end{tikzcd}$$

Ex. 8.2: Find simulations

$$
\begin{tikzcd}
 & q_1 \arrow[r, "d"] & q_2 \\
q_0 \arrow[ur, "a"] \arrow[dr, "a"'] & & \\
 & q_4 \arrow[r, "e"'] & q_3
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & & p_2 \\
p_0 \arrow[r, "a"] & p_1 \arrow[ur, "d"] \arrow[dr, "e"'] & \\
 & & p_3
\end{tikzcd}
$$

Example

Ex. 8.2: Find simulations

$$\begin{tikzcd}
 & q_1 \arrow[r, "d"] & q_2 \\
q_0 \arrow[ur, "a"] \arrow[dr, "a"'] & & \\
 & q_4 \arrow[r, "e"'] & q_3
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & p_2 \\
p_0 \arrow[r, "a"] & p_1 \arrow[ur, "d"] \arrow[dr, "e"'] & \\
 & & p_3
\end{tikzcd}$$

$$q_0 \lesssim p_0 \qquad \text{cf.} \qquad \{\langle q_0, p_0\rangle, \langle q_1, p_1\rangle, \langle q_4, p_1\rangle, \ldots\}$$

Similarity

Definition

$$p \lesssim q \;\equiv\; \langle \exists\, R \;::\; R \text{ is a simulation and } \langle p, q \rangle \in R \rangle$$

We say p is simulated by q.

Lemma

The similarity relation is a preorder
(ie, reflexive and transitive)

EQ3 – Bisimilarity

Bisimulation

Definition

Given $\langle S_1, N, \longrightarrow_1 \rangle$ and $\langle S_2, N, \longrightarrow_2 \rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a bisimulation iff both R and its converse R° are simulations.

I.e., whenever $\langle p, q \rangle \in R$ and $a \in N$,

$$
\begin{aligned}
&(1) \quad p \xrightarrow{a}_1 p' \;\Rightarrow\; \langle \exists\, q' \;:\; q' \in S_2 : \; q \xrightarrow{a}_2 q' \,\wedge\, \langle p', q' \rangle \in R \rangle \\
&(2) \quad q \xrightarrow{a}_2 q' \;\Rightarrow\; \langle \exists\, p' \;:\; p' \in S_1 : \; p \xrightarrow{a}_1 p' \,\wedge\, \langle p', q' \rangle \in R \rangle
\end{aligned}
$$

$$
\begin{tikzcd}
p \arrow[dd, "a"] & R & q & & & & q \arrow[dd, "a"] \\
& & & \Rightarrow & & & \\
p' & & & & p' & R & q'
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
p \arrow[dd, "a"] & & & & p & R & q \arrow[dd, "a"] \\
& & & \Leftarrow & & & \\
p' & R & q' & & & & q'
\end{tikzcd}
$$

Examples

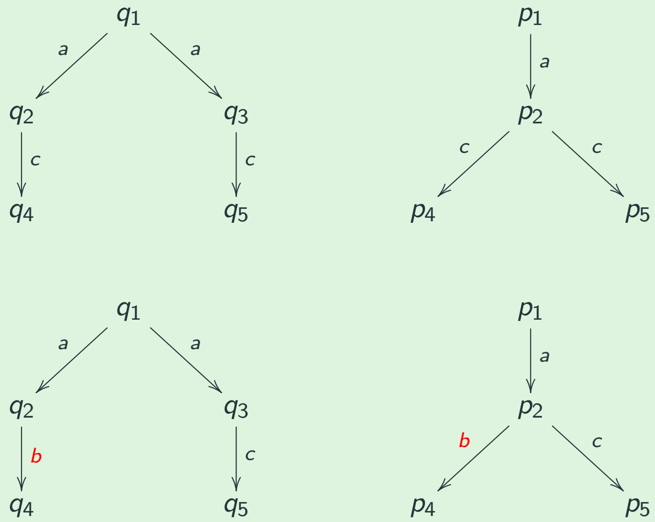
Ex. 8.3: Find bisimulations that include q_1

$$\begin{tikzcd}
 & q_1 \arrow[dl, "a"'] \arrow[dr, "a"] & \\
q_2 \arrow[rr, "c"] & & q_3 \arrow[loop right, "c"]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
m \arrow[d, "a"'] \\
n \arrow[loop right, "c"]
\end{tikzcd}$$

$$q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{a} \cdots
\qquad\qquad
\begin{tikzcd}
h \arrow[loop right, "a"]
\end{tikzcd}$$

Examples

Ex. 8.4: Find bisimulations that include q_1



Bisimilarity

Definition

$$p \sim q \;\equiv\; \langle \exists\, R \;::\; R \text{ is a bisimulation and } \langle p, q\rangle \in R \rangle$$

We say p is bisimilar to q.

Lemma

Two processes P and Q are bisimilar if there is a bisimulation that includes $\langle P, Q\rangle$.

Properties

Warning

$$\boxed{\left[p \precsim q \text{ and } q \precsim p\right] \text{ does not imply } \left[p \sim q\right]}$$

Properties

Warning

$$\left[p \lesssim q \text{ and } q \lesssim p\right] \text{ does not imply } \left[p \sim q\right]$$

Example

$$q_0 \lesssim p_0, \;\; p_0 \lesssim q_0 \quad \text{but} \quad p_0 \not\sim q_0$$

$$\begin{tikzcd}
 & q_1 & \\
q_0 \arrow[ur, "a"] \arrow[dr, "a"'] & & \\
 & q_2 \arrow[r, "b"] & q_3
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
p_0 \arrow[r, "a"] & p_1 \arrow[r, "b"] & p_3
\end{tikzcd}$$

Notes

Similarity as the greatest simulation

$$\lesssim \triangleq \bigcup \{S \mid S \text{ is a simulation}\}$$

Bisimilarity as the greatest bisimulation

$$\sim \triangleq \bigcup \{S \mid S \text{ is a bisimulation}\}$$

Exercises

Ex. 8.5: P,Q Bisimilar?

$$\mathbf{P} = a.P_1$$
$$P_1 = b.P + c.P$$

$$\mathbf{Q} = a.Q_1$$
$$Q_1 = b.Q_2 + c.Q$$
$$Q_2 = a.Q_3$$
$$Q_3 = b.Q + c.Q_2$$

Ex. 8.6: P,Q Bisimilar?

$$\mathbf{P} = a.(b.\mathbf{0} + \mathbf{0})$$

$$\mathbf{Q} = a.b.\mathbf{0}$$

Ex. 8.7: P,Q Bisimilar?

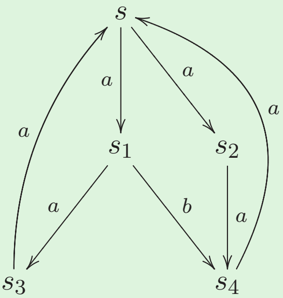
$$\mathbf{P} = a.(b.\mathbf{0} + c.\mathbf{0})$$

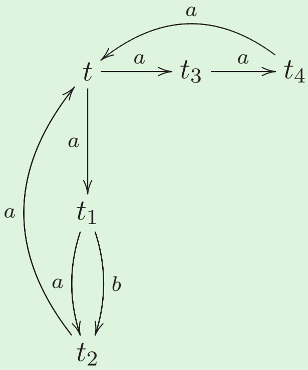
$$\mathbf{Q} = a.b.\mathbf{0} + a.c.\mathbf{0}$$

Draw their LTS. If bisimilar, find the bisimulation.

Exercises

Ex. 8.8: Find a bisimulation with $\langle s, t \rangle$





Exercise

Ex. 8.9: Find a simulation between $SmUni$ and $SmUni'$

$$CM = \text{coin}.\overline{\text{coffee}}.CM \qquad CM' = \text{coin}.(\overline{\text{coffee}}.CM' + \text{coin}.\overline{\text{latte}}.CM')$$

$$CS = \text{pub}.\overline{\text{coin}}.\text{coffee}.CS \qquad CS' = \text{pub}.\overline{\text{coin}}.(\text{coffee}.CS' + \overline{\text{coin}}.\text{latte}.CS')$$

$$SmUni = (CM|CS)\backslash\{\text{coin}, \text{coffee}\} \qquad SmUni' = (CM'|CS')\backslash\{\text{coin}, \text{coffee}, \text{latte}\}$$

Weak bisimilarity

Considering τ-transitions

Weak transition

$$p \stackrel{\alpha}{\Longrightarrow} q \quad \text{iff} \quad p\,(\stackrel{\tau}{\longrightarrow})^*\, q_1 \stackrel{a}{\longrightarrow} q_2\,(\stackrel{\tau}{\longrightarrow})^*\, q$$

$$p \stackrel{\tau}{\Longrightarrow} q \quad \text{iff} \quad p\,(\stackrel{\tau}{\longrightarrow})^*\, q$$

where $\alpha \neq \tau$ and $(\stackrel{\tau}{\longrightarrow})^*$ is the reflexive and transitive closure of $\stackrel{\tau}{\longrightarrow}$.

Weak bisimulation (vs. strong)

Given $\langle S_1, N, \longrightarrow_1\rangle$ and $\langle S_2, N, \longrightarrow_2\rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a bisimulation iff for all $\langle p, q\rangle \in R$ and $a \in N \cup \{\tau\}$,

$$(1)\quad p \stackrel{a}{\longrightarrow}_1 p' \;\Rightarrow\; \langle \exists\, q' \;:\; q' \in S_2 : \; q \stackrel{a}{\Longrightarrow}_2 q' \wedge \langle p', q'\rangle \in R\rangle$$

$$(2)\quad q \stackrel{a}{\longrightarrow}_2 q' \;\Rightarrow\; \langle \exists\, p' \;:\; p' \in S_1 : \; p \stackrel{a}{\Longrightarrow}_1 p' \wedge \langle p', q'\rangle \in R\rangle$$

Considering τ-transitions

Branching bisimulation

Given $\langle S_1, N, \longrightarrow_1\rangle$ and $\langle S_2, N, \longrightarrow_2\rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a bisimulation iff for all $\langle p, q\rangle \in R$ and $a \in N \cup \{\tau\}$,

(1) if $p \xrightarrow{a}_1 p'$ then either

(1.1) $a = \tau$ and $\langle p', q\rangle \in R$ or

(1.2) $\langle \exists\ q', q'' \in S_2\ ::\ q\,(\xrightarrow{\tau}_2)^*\, q' \xrightarrow{a}_2 q'' \wedge \langle p, q'\rangle \in R \wedge \langle p', q''\rangle \in R\rangle$

(2) if $q \xrightarrow{a}_2 q'$ then either

(2.1) $a = \tau$ and $\langle p', q'\rangle \in R$ or

(2.2) $\langle \exists\ p', p'' \in S_1\ ::\ p\,(\xrightarrow{\tau}_1)^*\, p' \xrightarrow{a}_1 p'' \wedge \langle p', q\rangle \in R \wedge \langle p'', q'\rangle \in R\rangle$

Exercise

Ex. 8.10: Search for a bisimulation, a weak bisimulation, and a branching bisimulation between *SmUni* and *SmUni'*

$$\begin{aligned} CM &= \mathrm{coin}.\overline{\mathrm{coffee}}.CM \\ CS &= \mathrm{pub}.\overline{\mathrm{coin}}.\mathrm{coffee}.CS \\ SmUni &= (CM|CS)\backslash\{\mathrm{coin}, \mathrm{coffee}\} \end{aligned}$$

$$\begin{aligned} CM'' &= \mathrm{coin}.(\mathrm{sel}.\overline{\mathrm{coffee}}.CM'' + \\ &\qquad \mathrm{coin}.\mathrm{sel}.\overline{\mathrm{latte}}.CM'') \\ CS'' &= \mathrm{pub}.\overline{\mathrm{coin}}.\overline{\mathrm{sel}}.(\mathrm{coffee}.CS'' \\ SmUni'' &= (CM''|CS'')\backslash\{\mathrm{coin}, \mathrm{coffee}, \mathrm{latte}, \mathrm{sel}\} \end{aligned}$$

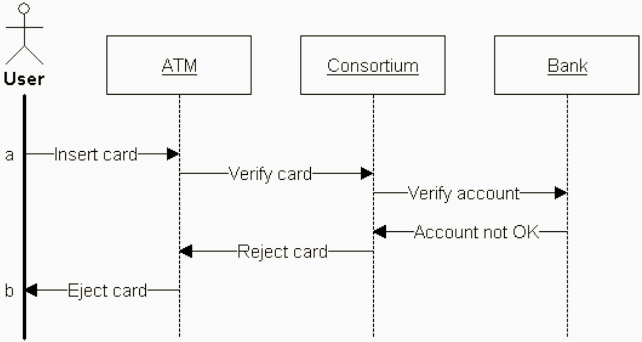
mCRL2 Tools

Slides 10:

`https://cister-labs.github.io/ramde2122/slides/10-mcrl2.pdf`

Realisability of Sequence Diagrams

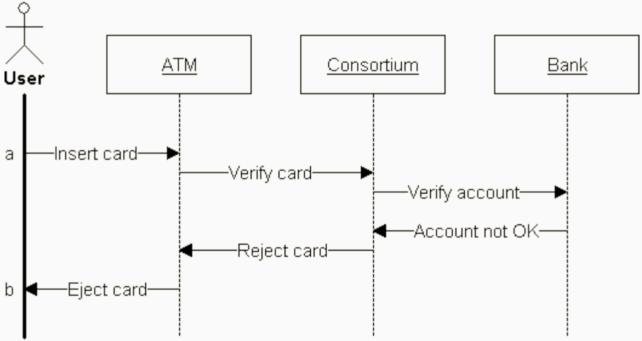
Recall: Sequence Diagrams as Interactive Processes



- Objects as Processes (e.g.,processes U, A, C, B)
- Send actions (e.g., $\mathit{insertCard}$)
- Reveive actions (e.g., $\overline{\mathit{insertCard}}$)

- Unique action for each object pair
- Do not write $(\ldots + \mathbf{0})$

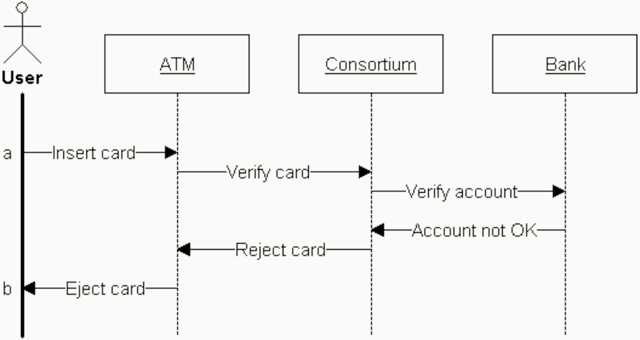
Recall: Language of Sequence Diagrams, Informally



This example has only 1 word and its prefixes

$$L_{sd} = \mathit{insertCard} \cdot \mathit{verifyCard} \cdot \mathit{verifyAccount} \cdot \mathit{accountNotOK} \cdot \mathit{rejectedCard} \cdot \mathit{ejectCard}$$

Recall: Sequence Diagrams as Interactive Processes



We can specify a SD as interactive processes

$$Sys = (U|A|C|E)\backslash \ldots$$
$$U = insertCard.\overline{ejectCart}.\mathbf{0}$$
$$A = \ldots$$
$$C = \ldots$$
$$E = \ldots$$

Sequence Diagrams covered by Interactive Processes

- Sequence diagrams depict scenarios
  (possible sequence of actions)
- Processes abstract implementations
  (simplified view of concrete implementations)

Processes can do more

E.g., an ATM that also *accepts* cards can (and should) still support the *rejection* scenario.

We want to observe interactions in such processes

Modified CCS semantics

$$\text{(com1)}\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \text{(com2)}\ \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \qquad \text{(com3)}\ \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau_a} P'|Q'}$$

$\alpha \in N \cup \overline{N} \cup \{\tau_a \mid a \in N\}$ is an action

Language inclusion

Recall *Sys* from Slide 27 and its diagram *sd*.

$$L_{sd} = \{iC \cdot vC \cdot cA \cdot aN \cdot rC \cdot eC\}$$
$$L_{Sys} = \{\tau_{iC} \cdot \tau_{vC} \cdot \tau_{cA} \cdot \tau_{aN} \cdot \tau_{rC} \cdot \tau_{eC}\}$$

Language inclusion

P includes sd

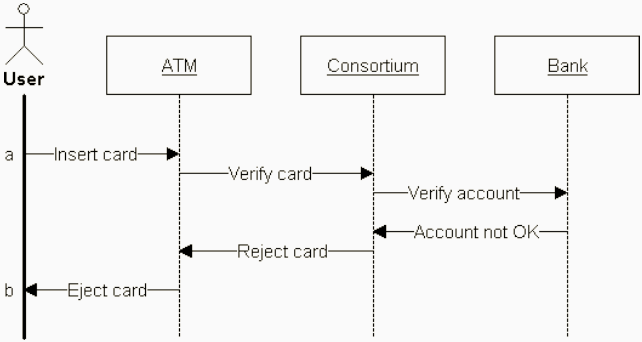
iff

$L_{sd} \subseteq L_{P^\dagger}$

$P^\dagger$ modifies P's LTS by:

filtering actions of *sd* and replacing τ_a by a

Are words enough?



Ex. 8.11: Let *sd* be the diagram above and recall Slide 27

Does *Sys* still includes *sd* if *U* is instead defined as below?

1. $U = \mathit{insertCard}.\overline{\mathit{ejectCard}}.\mathbf{0} + \mathit{insertCard}.\mathbf{0}$
2. $U = (\mathit{insertCard}.\overline{\mathit{ejectCard}}.\mathbf{0}) + \mathit{goAway}.\mathbf{0})$

Is language coverage enough?

Implementations can have:

- extra undesirable behaviour
- less behaviour

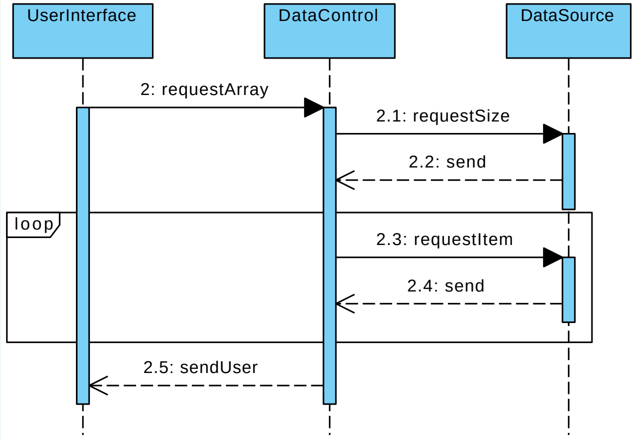
Alternative: change the inclusion/equivalence

Let $SD = \{sd_1, sd_2, \ldots\}$ be a set of sequence diagrams.

Language inclusion:	$L_{SD} \subseteq L_{P^\dagger}$
Language equivalence:	$L_{SD} = L_{P^\dagger}$
Similarity:	$NFA(SD) \lesssim P^\dagger$
Bisimilarity:	$NFA(SD) \sim P^\dagger$

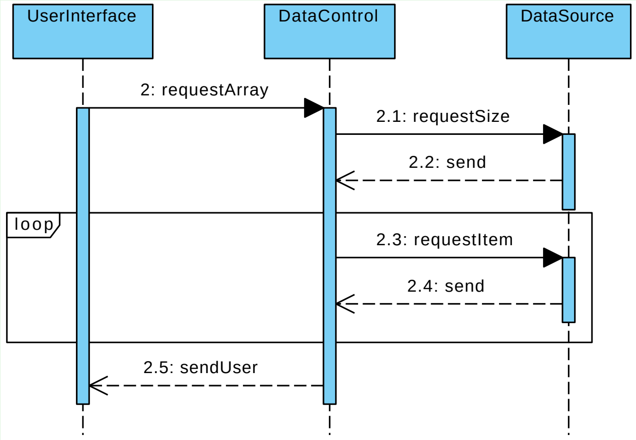
Exercise

Ex. 8.12: Draw an NFA that captures the following diagram



Exercise

Ex. 8.13: Write a process for each object of the diagram



Realisability

Question: after encoding SD into processes:

Can we recover the behaviour of the original sequence diagram

by composing

the encoded processes?

Realisability

A set SD of sequence diagrams is realisable

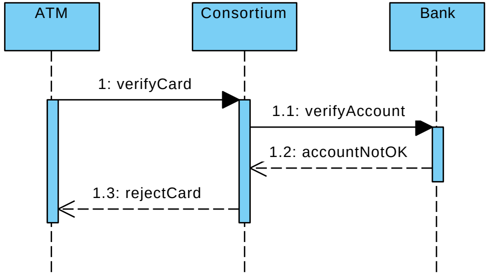
iff

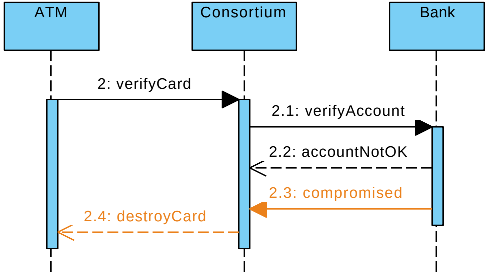
$$NFA(SD) \sim Comp(Proc(SD))^{\dagger}$$

$Proc(SD)$ returns the set of encoded processes for each $sd \in SD$
$Comp(P_1, P_2, \ldots) = (P_1|P_2|\ldots)\backslash\{actions\ of\ SD\}$

Exercise

Ex. 8.14: Are the diagrams below realisable?





1. draw $NFA(SD)$
2. calculate $Proc(SD)$
   Hint: $B = \overline{vA}.(aN.\mathbf{0} + aN.c.\mathbf{0})$
3. draw $Comp(\cdot)$
4. search for a bisimulation

Ex. 8.15: Verify if the diagram in Slide 34 is realisable.

Exercise

Ex. 8.16: Verify if the diagram is realisable.