Factorization for Component-Interaction Automata

Nikola Beneš Ivana Černá Filip Štefaňák

ParaDiSe Laboratory
Faculty of Informatics
Masaryk University
Czech Republic

SOFSEM
January 24th, 2012

Motivation

Component-Based Software Development

- autonomous (third-party) components
- hierarchical composition
- correctness issues

Component-Interaction Automata

- automata-based formalism
- various kinds of component assembly
- captures interaction properties

Motivation

Factorization Problem

- dual to composition
- filling a "gap" in a system

Motivation

Factorization Problem

- dual to composition
- filling a "gap" in a system

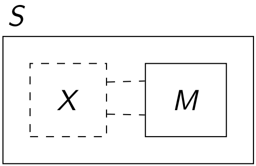
Formally:

- let $\parallel$ be a **composition** operation
- let $\simeq$ be a **conformity** relation (equivalence, refinement, ...)
- given S, M, find X such that $X \parallel M \simeq S$

Motivation

Factorization Problem

- dual to composition
- filling a "gap" in a system



Formally:

- let $\parallel$ be a **composition** operation
- let $\simeq$ be a **conformity** relation (equivalence, refinement, ...)
- given S, M, find X such that $X \parallel M \simeq S$

Preliminaries

Component-Interaction Automata

- labelled transition systems
- equipped with **ports** and **actions**

Preliminaries

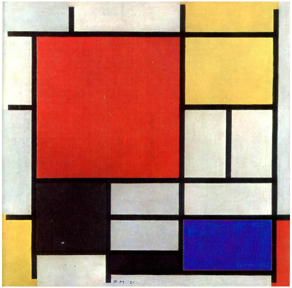
Component-Interaction Automata

- labelled transition systems
- equipped with **ports** and **actions**
- labels are triples $(\mathit{from}, \mathit{action}, \mathit{to})$
  - input: $(-, a, 1)$
  - output: $(1, a, -)$
  - internal: $(1, a, 1)$, $(1, a, 2)$

Preliminaries

Component-Interaction Automata

- labelled transition systems
- equipped with **ports** and **actions**
- labels are triples $(\mathit{from}, \mathit{action}, \mathit{to})$
  - input: $(-, a, 1)$
  - output: $(1, a, -)$
  - internal: $(1, a, 1)$, $(1, a, 2)$



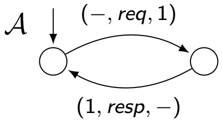
Composition $\otimes^{\mathcal{F}}$

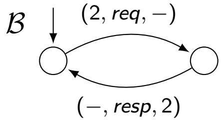
- two CI automata with disjoint ports
- complementary labels may be combined

$$(1, a, -) + (-, a, 2) \rightarrow (1, a, 2)$$

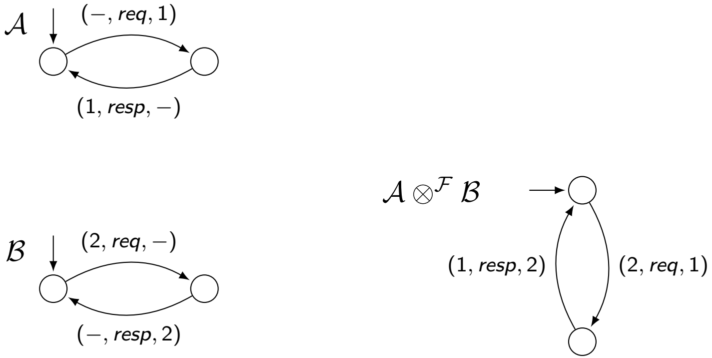
- composition parameter $\mathcal{F}$ – which labels ought to be combined and/or left open

Example



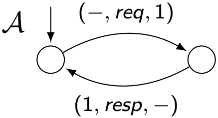


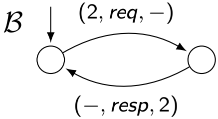
Example

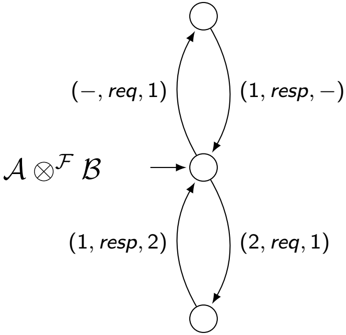


$$\mathcal{F} = \{(2, req, 1), (1, resp, 2)\}$$

Example



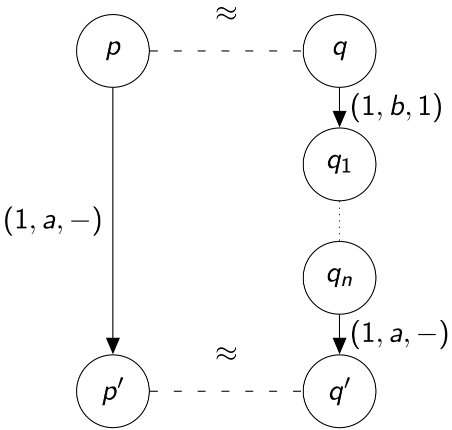




$$\mathcal{F} = \{(2, req, 1), (1, resp, 2), (-, req, 1), (1, resp, -)\}$$

Weak Bisimulation $\approx$

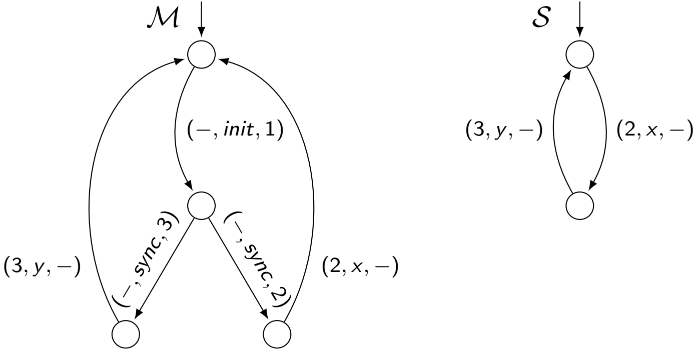
- observational equivalence
- ignore (in certain sense) internal transitions



Our Problem

Factorization Problem for Component-Interaction Automata

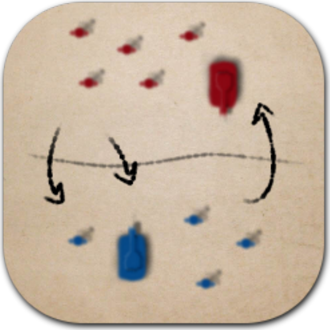
Given two CI automata $\mathcal{S}$, $\mathcal{M}$, find a parameter $\mathcal{F}$ and a CI automaton $\mathcal{X}$ such that $\mathcal{X} \otimes^{\mathcal{F}} \mathcal{M} \approx \mathcal{S}$.



Our Solution

The "Battle Plan": Reduction to Similar Problem

- Qin, H., Lewis, P.: Factorization of finite state machines under observational equivalence. In: CONCUR 90. LNCS, vol. 458.
- finite automata with
  - τ actions
  - handshake composition a with $\bar{a}$
  - weak bisimulation
- constraints on S
  - deterministic
  - without τ
- the method
  - transform $\mathcal{S}$, $\mathcal{M}$ into FSMs
  - solve the FSM factorization
  - transform the result back into a CI automaton



Our Solution

Transformations f, $g_{\mathcal{F}}$: CI $\rightarrow$ FSM

- the goal: $\mathcal{A} \otimes^{\mathcal{F}} \mathcal{B} \approx \mathcal{C}$ if and only if $g_{\mathcal{F}}(\mathcal{A}) \parallel g_{\mathcal{F}}(\mathcal{B}) \approx f(\mathcal{C})$
- $g_{\mathcal{F}}$ should be invertible

Transformation f

- retain the structure
- actions of the FSM are labels of the original CI automaton

Transformation $g_{\mathcal{F}}$

- parametrized with a set of labels $\mathcal{F}$
- external labels allowed by $\mathcal{F}$ retained
- external labels that can be combined are changed and possibly multiplied

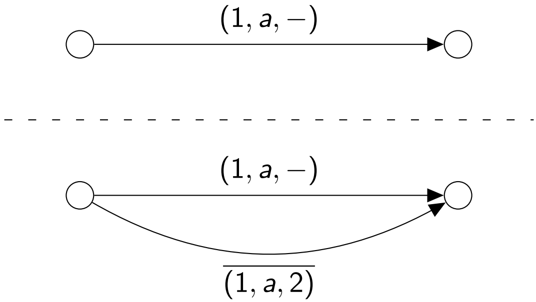
Our Solution

Transformation $g_{\mathcal{F}}$ – Example

- let $\mathcal{F} = \{(1, a, -), (1, a, 2), (3, b, 1), (4, b, 1)\}$

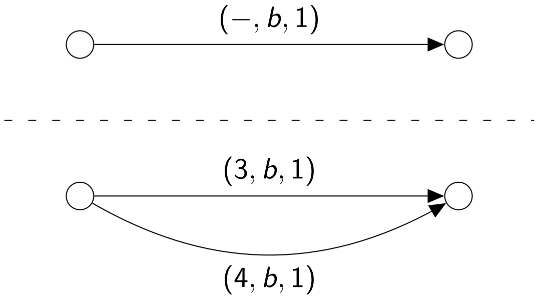
Transformation $g_{\mathcal{F}}$ – Example

- let $\mathcal{F} = \{(1, a, -), (1, a, 2), (3, b, 1), (4, b, 1)\}$



Transformation $g_{\mathcal{F}}$ – Example

- let $\mathcal{F} = \{(1, a, -), (1, a, 2), (3, b, 1), (4, b, 1)\}$



Our Solution

Ensuring Invertibility of $g_{\mathcal{F}}$

- $\mathcal{F}$ has to be label injective
- $(r, a, x), (r, a, y) \in \mathcal{F} \implies x = y$ where r is a port of $\mathcal{X}$

Ensuring Completeness of The Reduction

- $\mathcal{F}$ has to be complete wrt $\mathcal{M}$
- for every $(m, a, -)$ of $\mathcal{M}$ there has to some $(m, a, z) \in \mathcal{F}$

Finding the Right $\mathcal{F}$

- mirror the ports of $\mathcal{M}$: $r \to r'$
- set $\mathcal{F}$ contains all labels of S plus (r, a, r') for every $(r, a, -)$ of $\mathcal{M}$

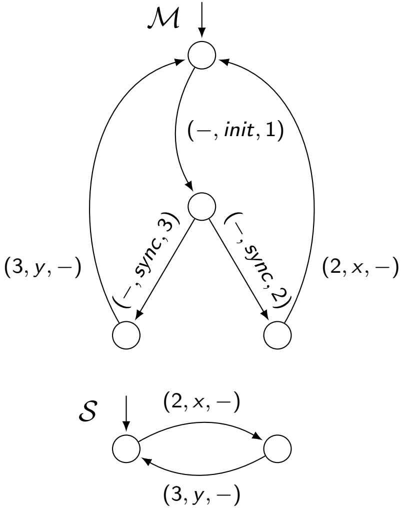
Our Solution

The Algorithm

input: CI automata $\mathcal{M}$ and $\mathcal{S}$ such that $\mathcal{S}$ is deterministic and $\mathit{Reach}(\mathcal{S})$ contains no internal labels.

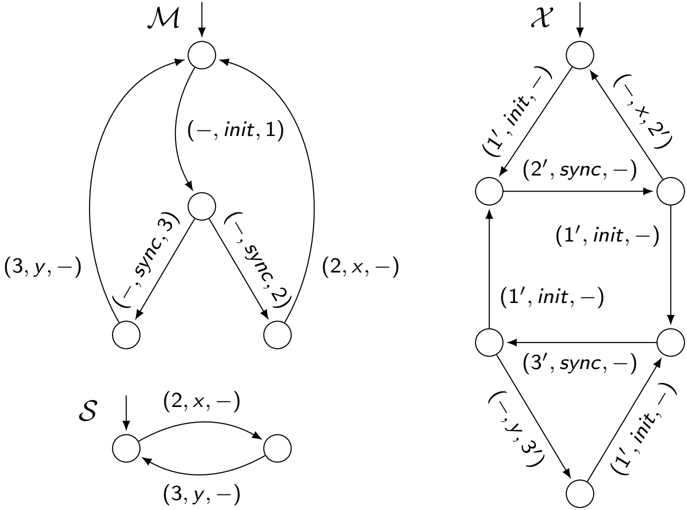
1. Let $\mathcal{F} = \mathit{Reach}(\mathcal{S}) \cup \{(r, a, r') \mid (r, a, -) \in \mathit{Reach}(\mathcal{M})\} \cup \{(s', a, s) \mid (-, a, s) \in \mathit{Reach}(\mathcal{M})\}$
2. Transform $\mathcal{M}$ into $g_{\mathcal{F}}(\mathcal{M})$ and $\mathcal{S}$ into $f(\mathcal{S})$.
3. Solve the factorization problem for FSM $g_{\mathcal{F}}(\mathcal{M})$ and $f(\mathcal{S})$ using the algorithm of Qin and Lewis.
   (If no solution exists, claim that no solution for the original problem exists.)
4. Produce a CI automaton $\mathcal{X}$ such that $g_{\mathcal{F}}(\mathcal{X})$ is the solution obtained in previous step.
5. Output $\mathcal{F}$, $\mathcal{X}$ as the solution of the original problem.

Example Solution



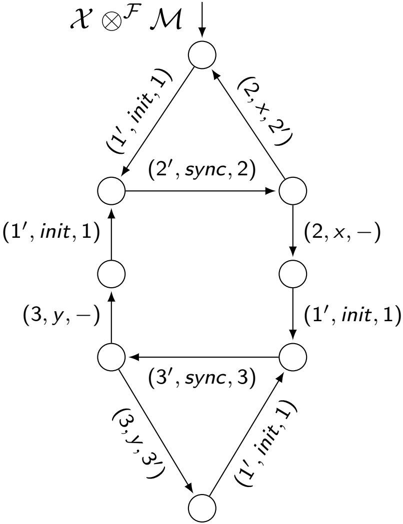
$$\{(2, x, -), (3, y, -), (2, x, 2'), (3, y, 3'), (1', init, 1), (2', sync, 2), (3', sync, 3)\}$$

Example Solution

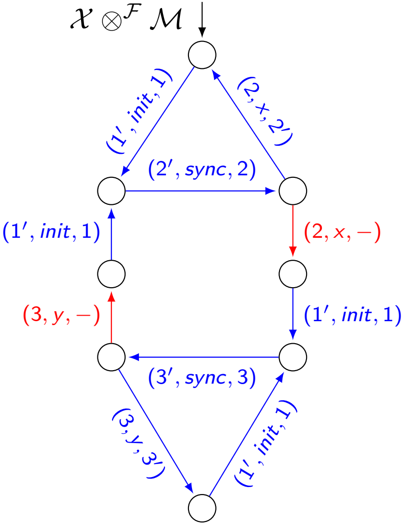


$$\{(2, x, -), (3, y, -), (2, x, 2'), (3, y, 3'), (1', init, 1), (2', sync, 2), (3', sync, 3)\}$$

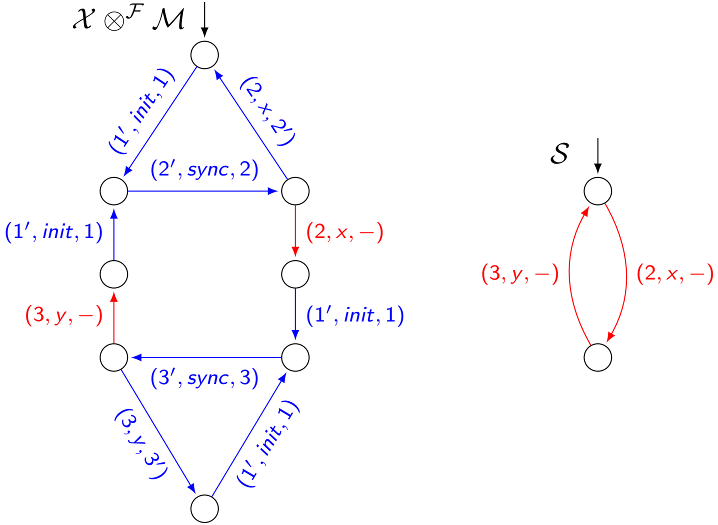
Example Solution



Example Solution



Example Solution



Conclusion

Component-Interaction Automata

- modelling component-based systems
- flexible (parametrized) composition

Factorization Problem

- dual to composition
- given $\mathcal{S}$ and $\mathcal{M}$,
  find $\mathcal{F}$ and $\mathcal{X}$ such that $\mathcal{X} \otimes^{\mathcal{F}} \mathcal{M} \approx \mathcal{S}$

Our Solution

- reduction to FSMs, solved by Qin and Lewis (CONCUR 1990)
- requires deterministic $\mathcal{S}$ with no internal transitions

Future Work

- port minimization
- general description of all solutions (modal transition system?)