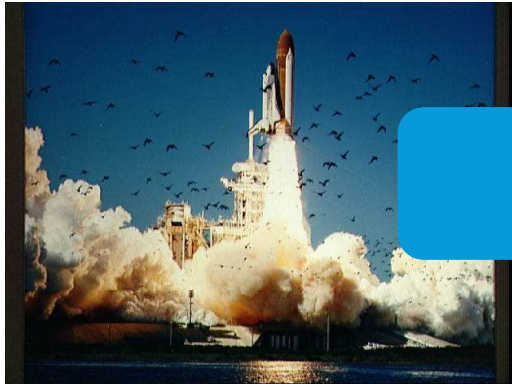


Dynamic Fault Tree analysis using Input/Output Interactive Markov Chains

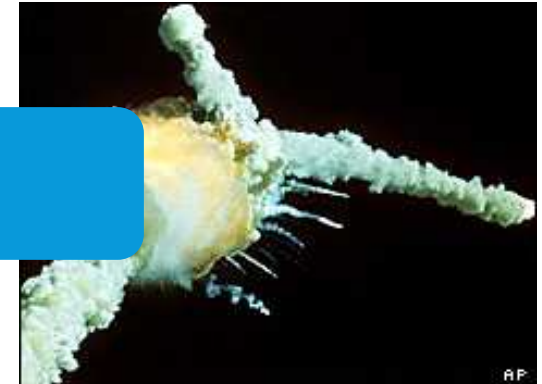
Hichem Boudali, Pepijn Crouzen, and Mariëlle Stoelinga.

**Formal Methods and Tools group
CS, University of Twente, NL.**

Motivation (and setting)



Systems do fail



-- Reliability Engineering --

Goal: Reduce system failure probability.

Methodology: Identify/analyze failure modes and their effects.

Example methodology:
Dynamic Fault Trees (DFT)

But:
DFTs have drawbacks ☹️

- Dynamic fault trees (DFT).
 - Definition, Example, Solution, Drawbacks.
- Input/Output interactive Markov chains (I/O-IMC).
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- Summary.

- **Dynamic fault trees (DFT).**
 - Definition, Example, Solution, Drawbacks.
- Input/Output interactive Markov chains (I/O-IMC).
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- Summary.

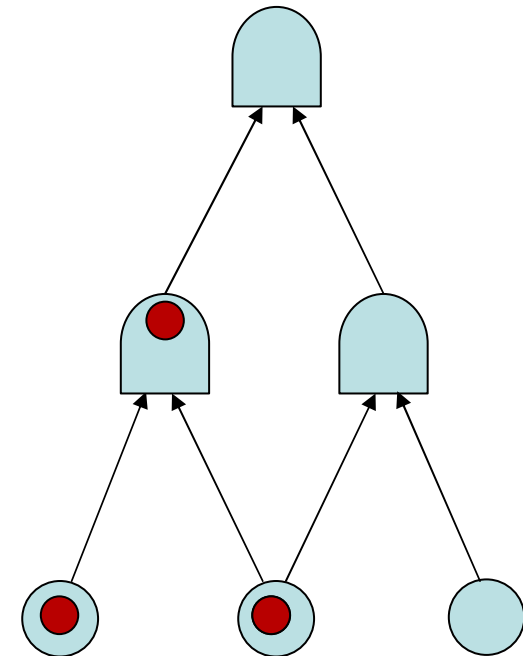
Dynamic Fault Trees (DFT)

- Extend standard fault trees with **dynamic gates**.
- Enable modelling complex **behaviours** and **interactions** between components.
- **combination** & **order** of failures matter.

Unreliability = Prob[System fails within T time units]

(dynamic) Fault trees

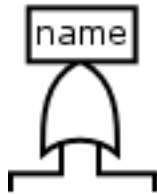
- Upside-down tree (graph)
- Leaves: **Basic events** (BE)
- Nodes: **Gates** (complex events)
- BEs + Gates: Elements
- Arrows: Causal relations
- One **top-node**: the “**root**” node
- The top-node models system failure
- Failure propagation: From leaves to root



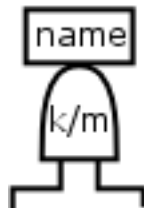
DFTs: Static gates (combination)



And-gate: fails if all its inputs have failed.

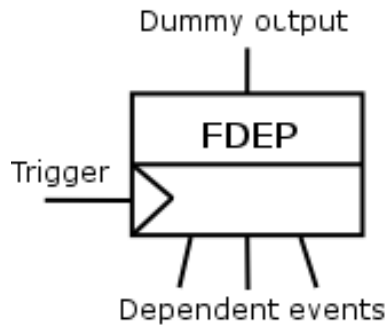


Or-gate: fails if one of its inputs has failed.

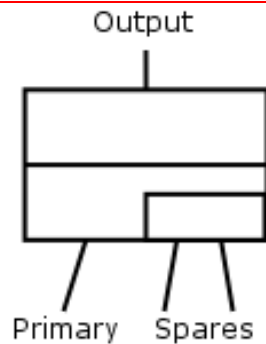


k/m-gate: fails if at least k of its m inputs have failed.

DFTs: Dynamic gates (order)



Functional dependency: When the trigger event fires, the dependent events also fire.

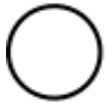


Spare gate: When the primary fires, the leftmost spare becomes active. The gate fires when there are no more spares. Spares are dormant before they are activated



Priority-AND: Same behavior as an AND-gate, except that the inputs must fire in left-to-right order, otherwise the PAND-gate will not fire.

DFTs: Basic events (BE)



Cold basic event: A basic event that, when active, fires after some delay. It cannot fire of its own accord when dormant.

**BE maps to a
Basic Physical
component**



Warm basic event: A basic event that fires after some delay. It is more likely to fire when active.

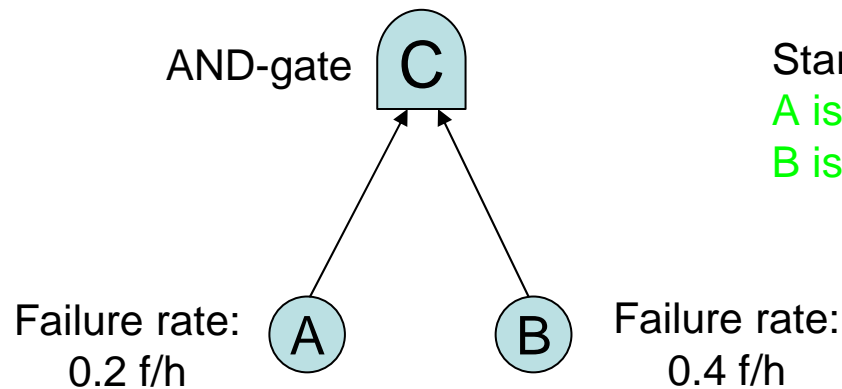
**Temperature
of a BE:
Relevant when
used as a spare**



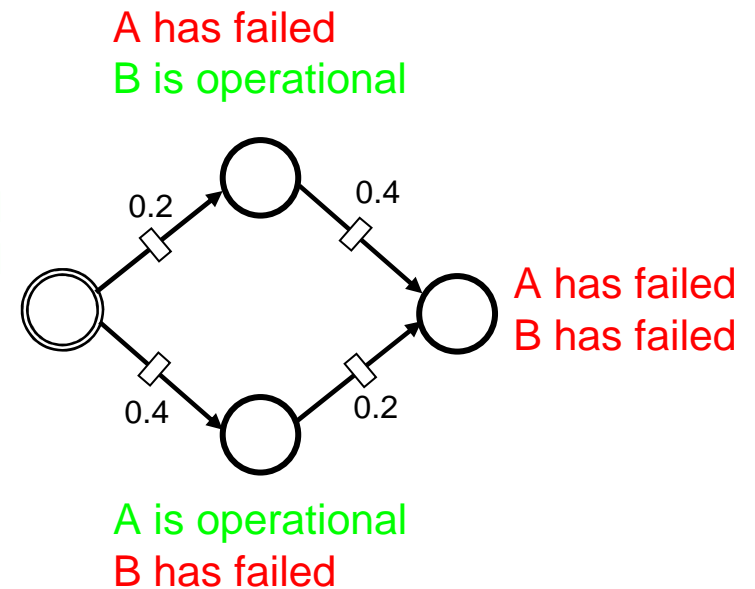
Hot basic event: A basic event that fires after some delay. Being dormant or active has no influence on the likelihood of the event firing.

DFT solution

- Convert the DFT into a Continuous-time Markov chain.
- Analyze CTMC using standard solution techniques.
- For (partially) static DFT, binary decision diagrams can be used!



Starting state:
A is operational
B is operational



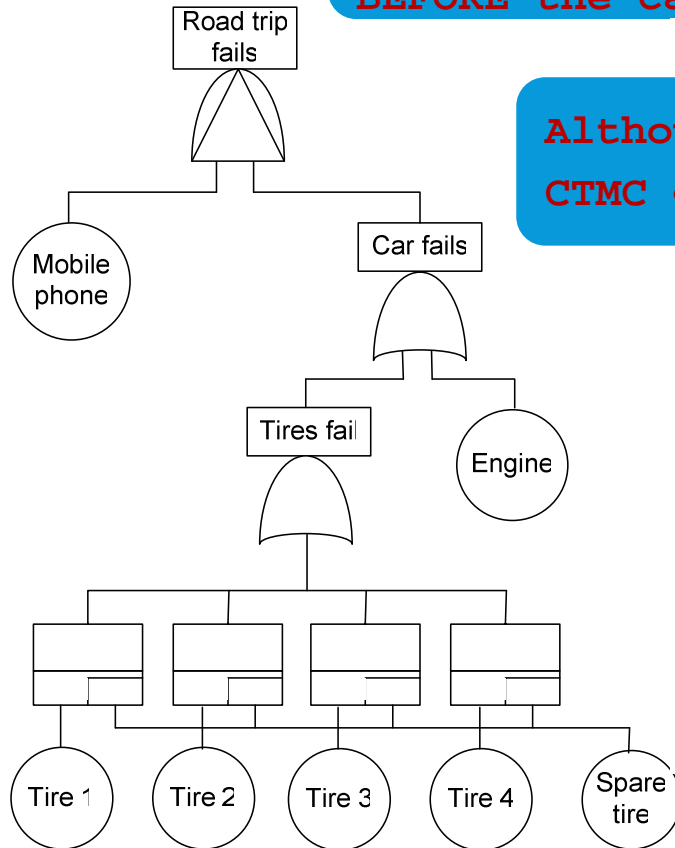
$$\Pr(A \text{ fails in } T \text{ hours}) = 1 - e^{-0.2 \cdot T}$$

$$A\text{'s Mean time to failure} = 1/0.2 = 5 \text{ hours}$$

Unreliability = Prob[Being in state ]

DFT example

Road trip fails if mobile phone fails BEFORE the car fails



Although distinct modules, CTMC generation in One shot

State-Space Explosion!

One of the drawbacks

Spare tire is cold: It cannot fail when not in use

DFT drawbacks

- State-space explosion. **Compositional Aggregation**
- No formal syntax and semantics. **DAG** **I/O-IMC**
- Lack of modularity:
 - Dynamic modules (e.g. ‘Tires’ subsystem in the example) can not be reused. **Compositionality**
 - Restrictions on certain inputs to gates (e.g. spare gate). **Lift restrictions**
- DFT-to-MC* conversion algorithm is hard to extend and/or modify. **Extension: At the element level**

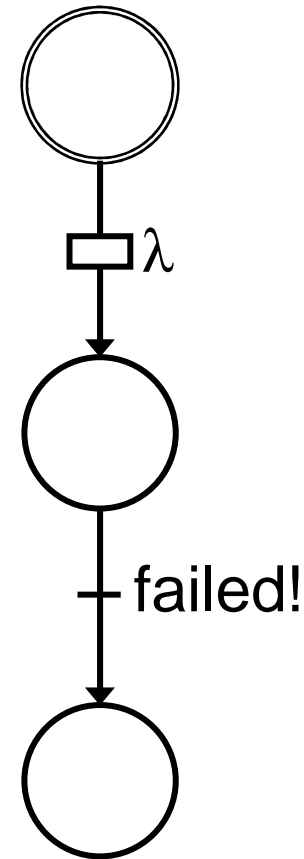
*: DIFTree algorithm

- Dynamic fault trees (DFT).
 - Definition, Example, Solution, Drawbacks.
- **Input/Output interactive Markov chains (I/O-IMC).**
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- Summary.

Input/Output Interactive Markov Chains (I/O-IMC)

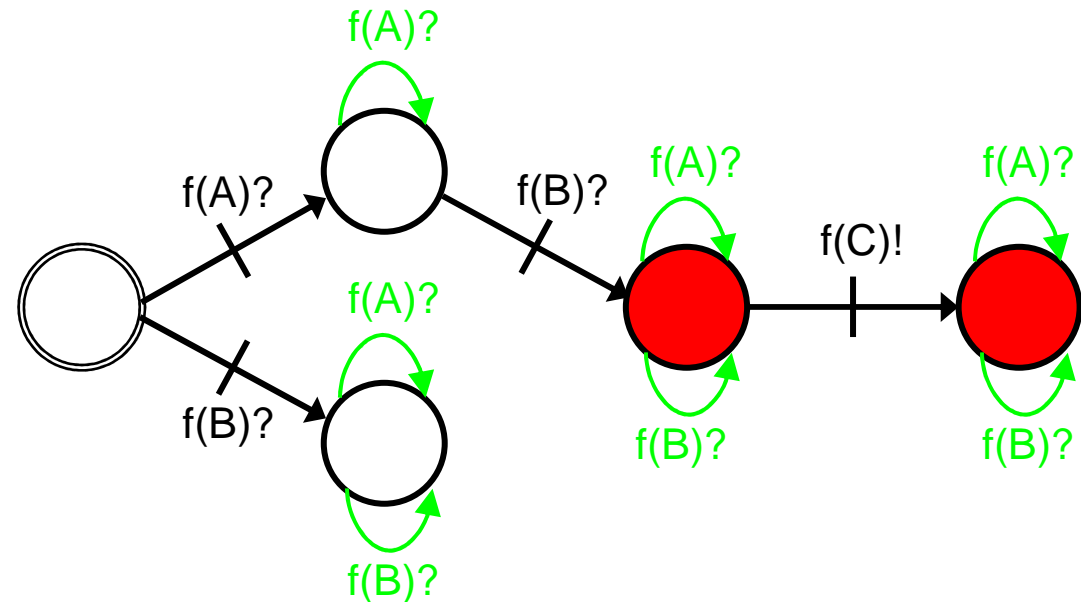
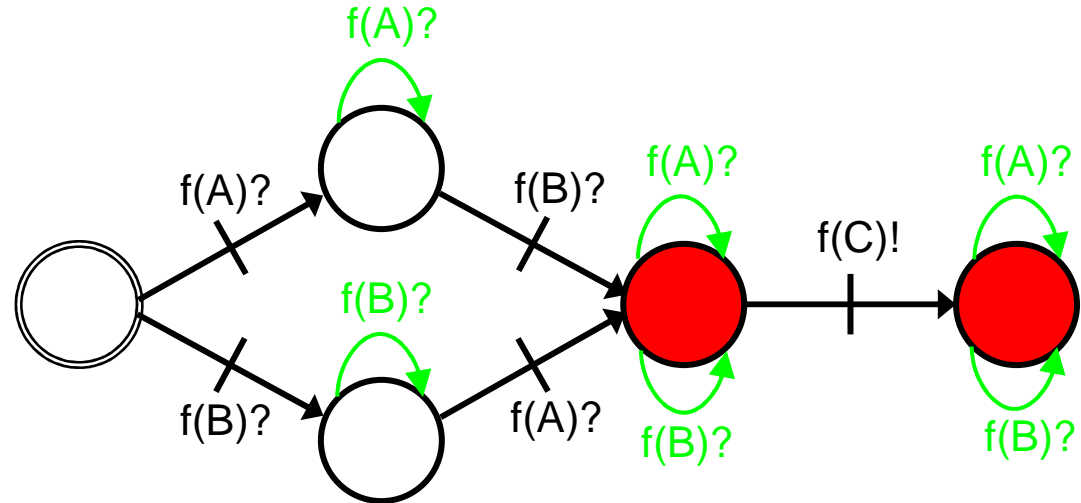
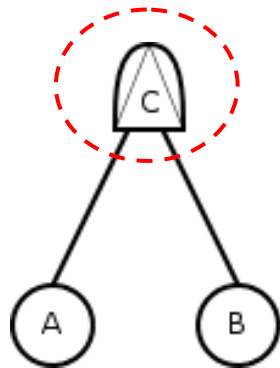
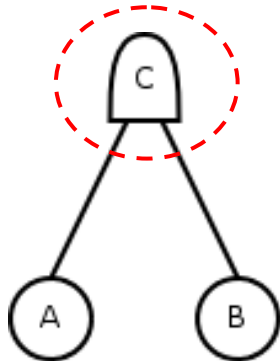
- A stochastic process algebra
- Combination of I/O automata and CTMC
- Discrete state space
- Markovian transitions
- Interactive transitions
- Action signature
 - ? - Input actions
 - ! - Output actions
 - ; - Internal actions
- Input-enabled

Immediate

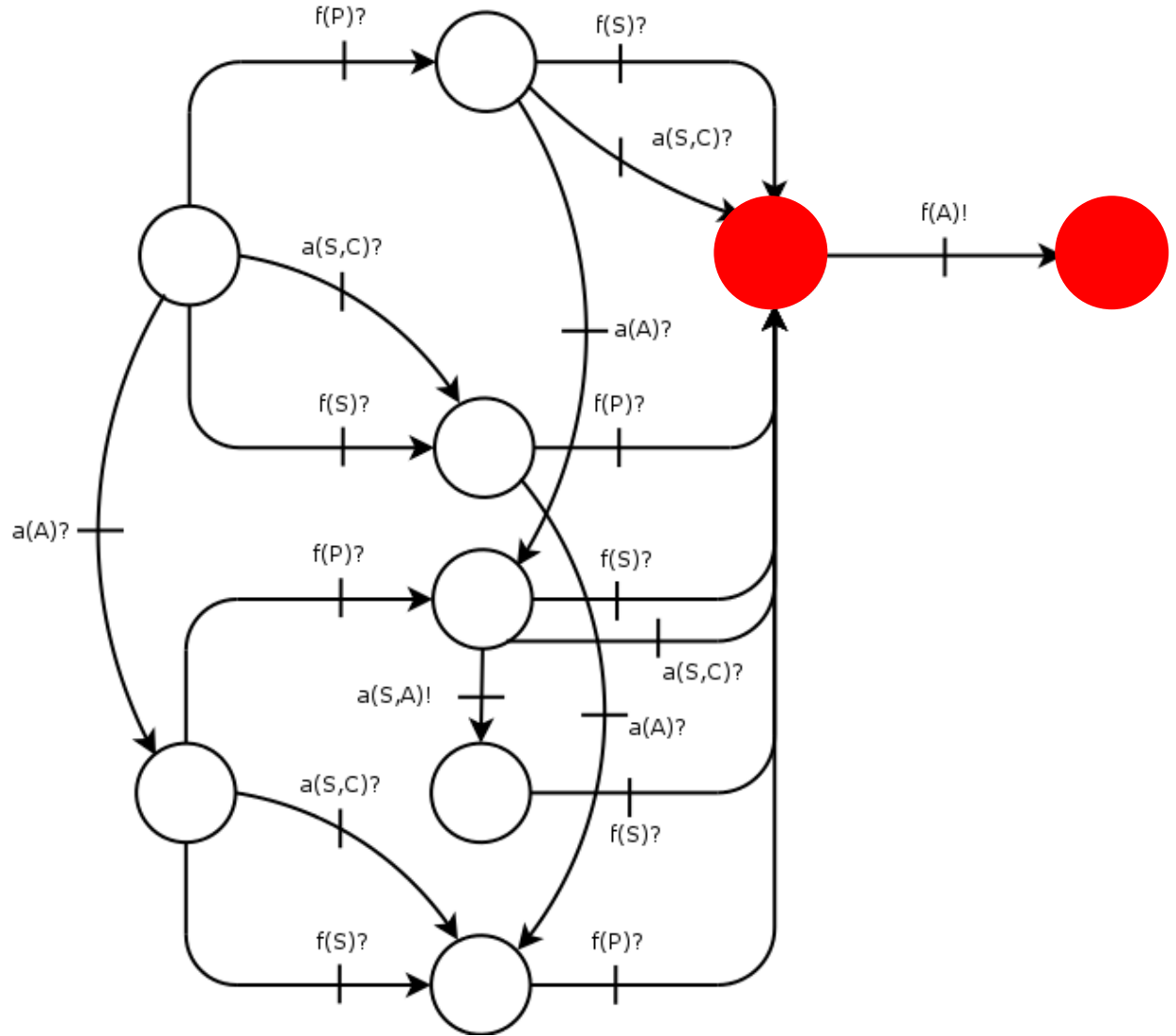
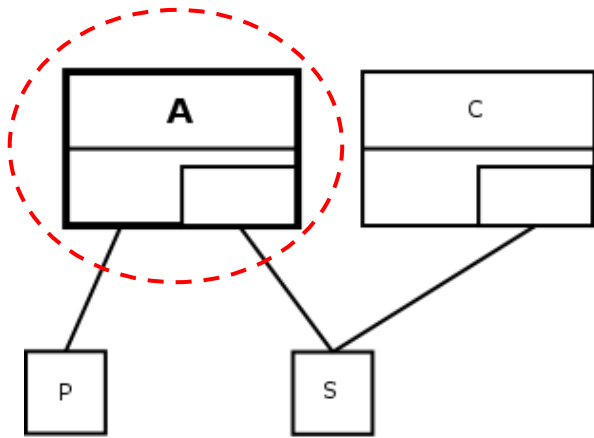


- Dynamic fault trees (DFT).
 - Definition, Example, Solution, Drawbacks.
- Input/Output interactive Markov chains (I/O-IMC).
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- Summary.

DFT semantics (DFT element to I/O-IMC)

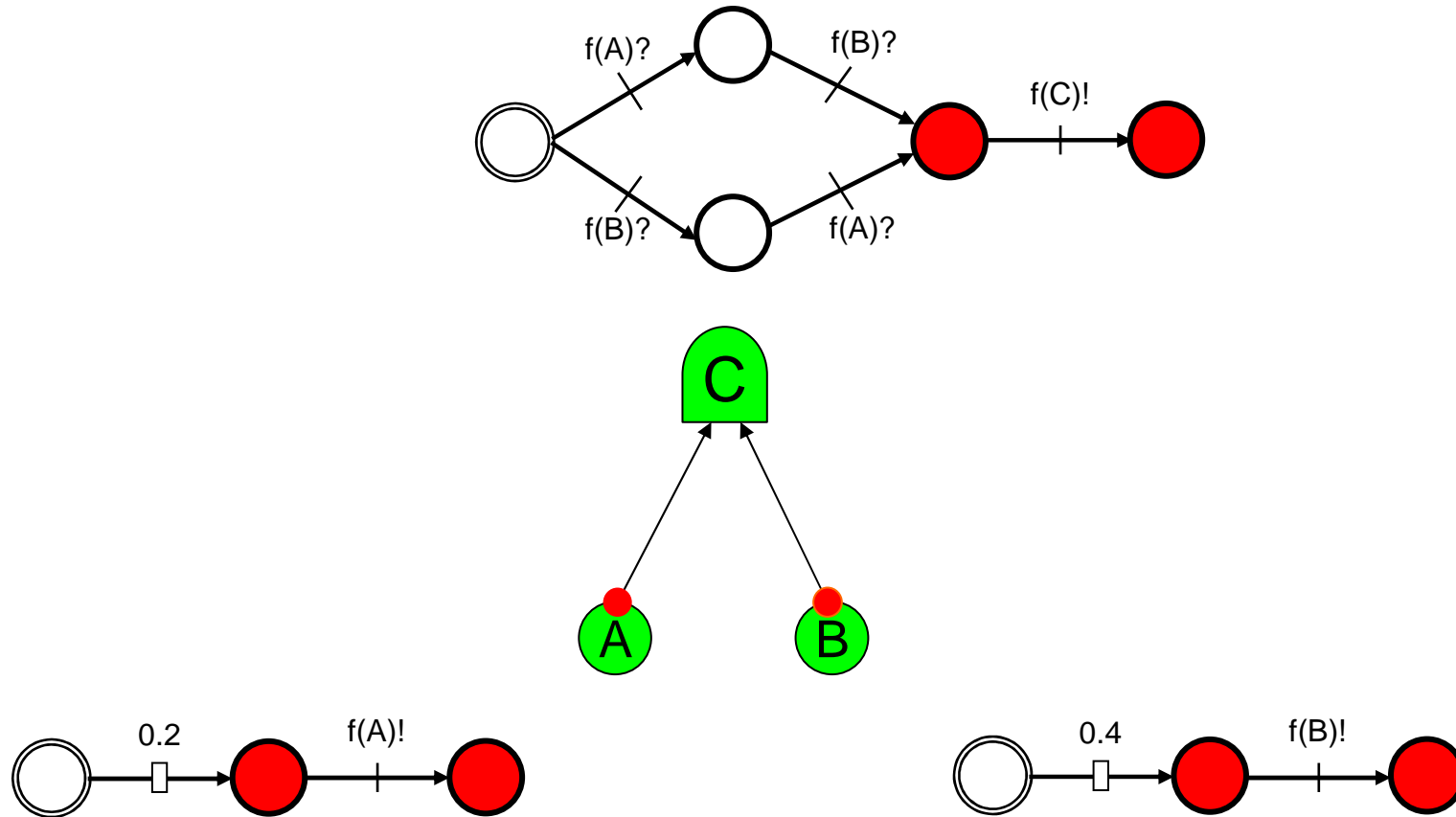


DFT semantics (DFT element to I/O-IMC)



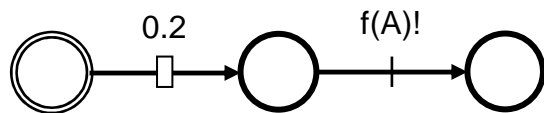
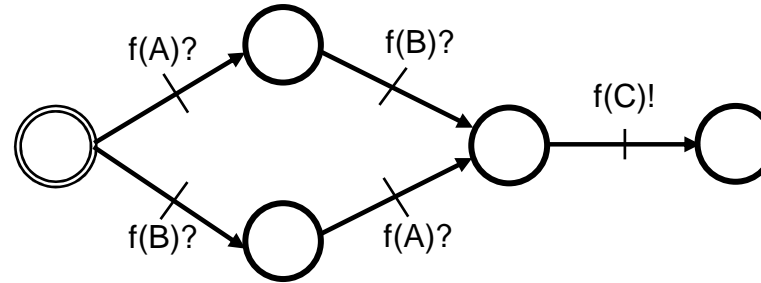
- Dynamic fault trees (DFT).
 - Definition, Example, Solution, Drawbacks.
- Input/Output interactive Markov chains (I/O-IMC).
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- Summary.

Compositional Analysis Translation



Compositional Analysis

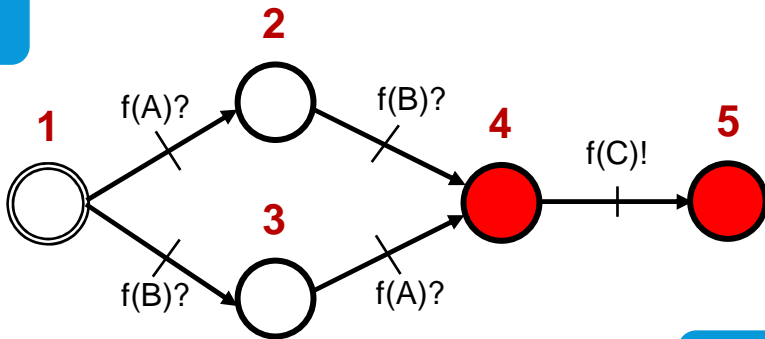
Parallel Composition



Compositional Analysis

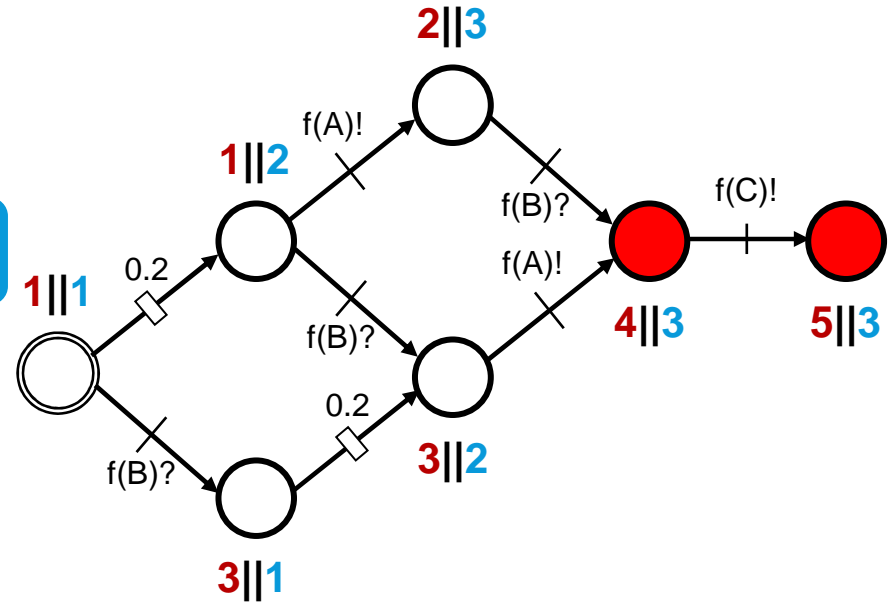
Parallel Composition

C



Inputs: $f(A)?$ and $f(B)?$
 Outputs: $f(C)!$

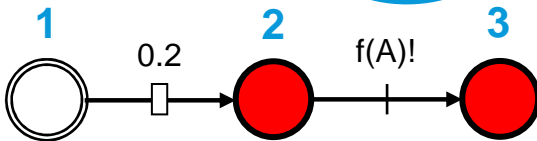
C||A



Synchronize on $f(A)$

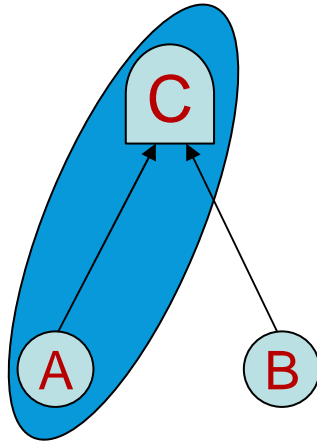
A

Inputs: none
 Outputs: $f(A)!$

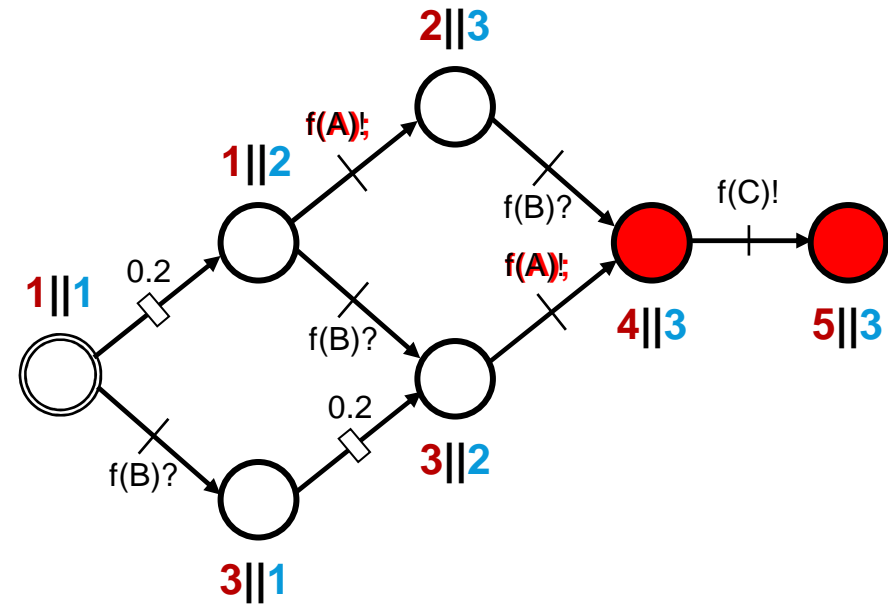


Compositional Analysis

Abstraction (hiding)



**Abstraction (hiding):
Makes signal internal**



Compositional Analysis

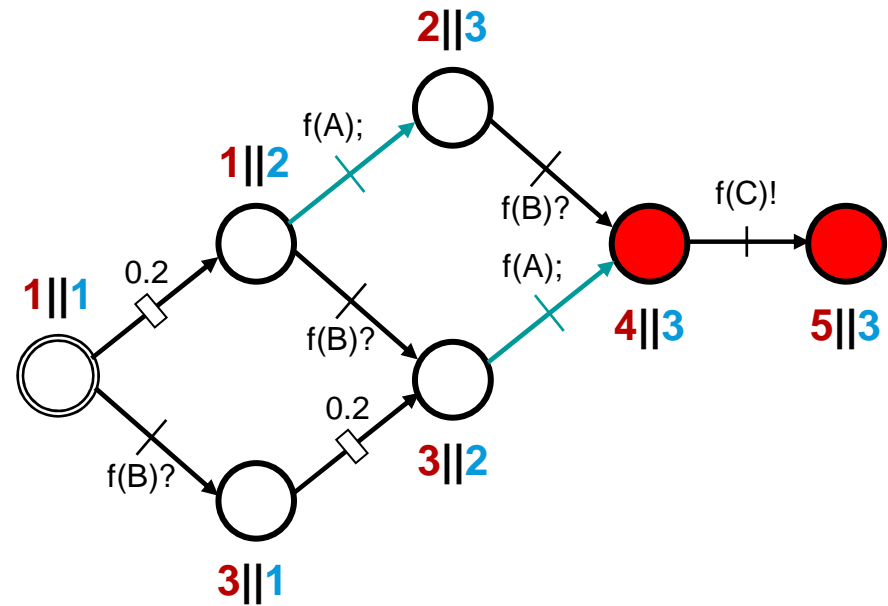
Aggregation (weak bisimulation)

Aggregation:

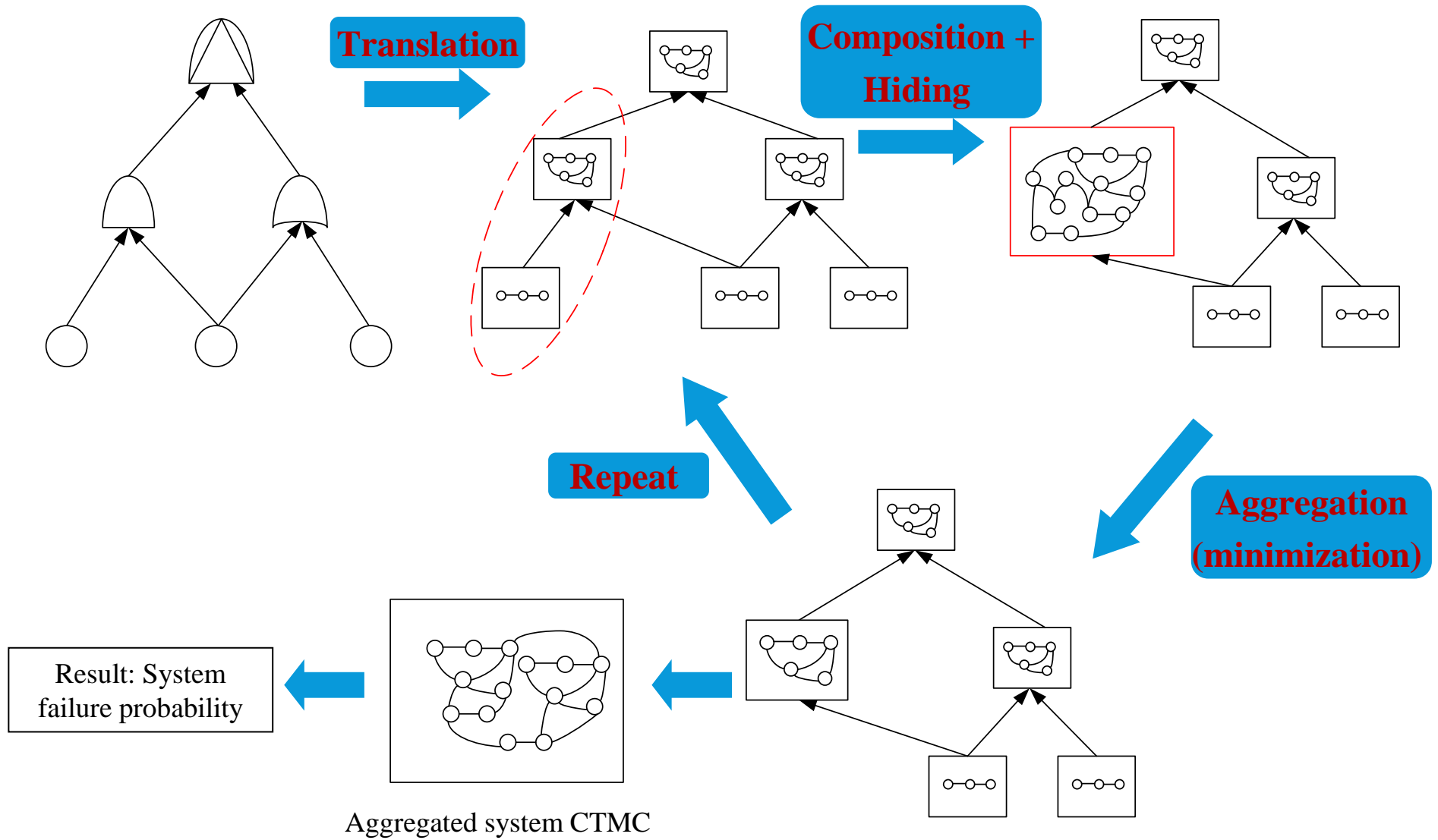
Finding a smaller model equivalent (behaviorally) to the original

Weak bisimulation:

Disregard internal steps

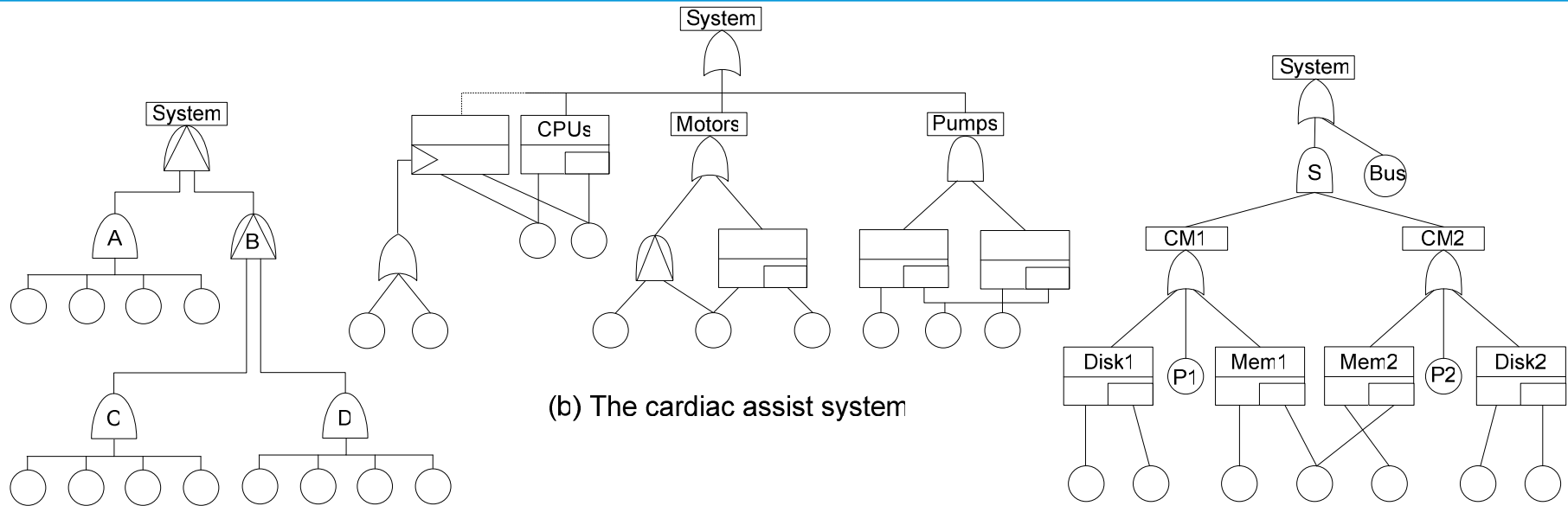


Compositional-Aggregation Overview



- Dynamic fault trees (DFT).
 - Definition, Example, Solution, Drawbacks.
- Input/Output interactive Markov chains (I/O-IMC).
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- Summary.

Case studies



(a) The cascaded PAND system

(b) The cardiac assist system

(c) A multi-processor distributed computing system

Case study	Analysis method	Max number of states	Max number of transitions	Unreliability (T=1)
(a)	DIFTree	4113	24608	0.00135668
(a)	Comp-Agg	132	426	0.00135668
(b)	DIFTree	8	10	0.657900
(b)	Comp-Agg	36	119	0.657900
(c)	DIFTree	253	1383	$2.00025 \cdot 10^{-9}$
(c)	Comp-Agg	157	756	$2.00025 \cdot 10^{-9}$

- Dynamic fault trees (DFT).
 - Definition, Example, Solution, Drawbacks.
- Input/Output interactive Markov chains (I/O-IMC).
- DFT semantics in terms of I/O-IMCs.
- DFT compositional analysis.
 - Translation, || Composition, Abstraction, Aggregation.
- Case studies.
- **Summary.**

Summary

Compositional semantics for DFTs

- Alleviate state-space explosion problem.
- Formal syntax & semantics.
- Enhanced DFT modularity:
 - Dynamic module reuse.
 - Lifting restrictions on allowed inputs.
- Readily extensible framework (extensions at the element level); e.g. repair.
- Works well for highly-modular dynamic FTs.

Gain at the modeling & analysis levels

References

- H. Boudali, P. Crouzen, M. Stoelinga. “Dynamic Fault Tree analysis using Input/Output Interactive Markov Chains”, to appear, DSN 2007 proceedings.
- H. Boudali, P. Crouzen, M. Stoelinga. “A compositional semantics for Dynamic Fault Trees in terms of Interactive Markov Chains”, Technical report, to appear.
- More info: hboudali@cs.utwente.nl

The END!

Extra slides

Future work

- Weaker bisimulation relation (i.e. more aggressive state reduction)
- Extension to non-exponential distributions (e.g. use of phase-type distributions)
- Further extensions to DFT modeling capabilities (i.e. definition of new gates and corresponding I/O-IMC)
- Fully automated tool (at this point, the tool is only partially automated)

1. \mathcal{P} and \mathcal{Q} are composable if $A_{\mathcal{P}}^O \cap A_{\mathcal{Q}}^O = A_{\mathcal{P}}^{int} \cap A_{\mathcal{Q}} = A_{\mathcal{P}} \cap A_{\mathcal{Q}}^{int} = \emptyset$.
2. If \mathcal{P} and \mathcal{Q} are composable I/O-IMCs, their composition $\mathcal{P} \parallel \mathcal{Q}$ is the I/O-IMC $(S_{\mathcal{P}} \times S_{\mathcal{Q}}, (s_{\mathcal{P}}^0, s_{\mathcal{Q}}^0), ((A_{\mathcal{P}}^I \cup A_{\mathcal{Q}}^I) \setminus (A_{\mathcal{P}}^O \cup A_{\mathcal{Q}}^O), (A_{\mathcal{P}}^O \cup A_{\mathcal{Q}}^O), (A_{\mathcal{P}}^{int} \cup A_{\mathcal{Q}}^{int})))$, $\rightarrow_{\mathcal{P} \parallel \mathcal{Q}}, \rightarrow_{\mathcal{P} \parallel \mathcal{Q}}^M$, where

$$\begin{aligned} \rightarrow_{\mathcal{P} \parallel \mathcal{Q}} &= \{(s, t) \xrightarrow{a}_{\mathcal{P} \parallel \mathcal{Q}} (s', t) \mid s \xrightarrow{a}_{\mathcal{P}} s' \wedge a \in A_{\mathcal{P}} \setminus A_{\mathcal{Q}}\} \\ &\quad \cup \{(s, t) \xrightarrow{a}_{\mathcal{P} \parallel \mathcal{Q}} (s, t') \mid t \xrightarrow{a}_{\mathcal{Q}} t' \wedge a \in A_{\mathcal{Q}} \setminus A_{\mathcal{P}}\} \\ &\quad \cup \{(s, t) \xrightarrow{a}_{\mathcal{P} \parallel \mathcal{Q}} (s', t') \mid s \xrightarrow{a}_{\mathcal{P}} s' \wedge t \xrightarrow{a}_{\mathcal{Q}} t' \wedge a \in A_{\mathcal{P}} \cap A_{\mathcal{Q}}\} \\ \rightarrow_{\mathcal{P} \parallel \mathcal{Q}}^M &= \{(s, t) \xrightarrow{\lambda}_{\mathcal{P} \parallel \mathcal{Q}}^M (s', t) \mid s \xrightarrow{\lambda}_{\mathcal{P}}^M s'\} \\ &\quad \cup \{(s, t) \xrightarrow{\lambda}_{\mathcal{P} \parallel \mathcal{Q}}^M (s, t') \mid t \xrightarrow{\lambda}_{\mathcal{Q}}^M t'\} \end{aligned}$$

3. Let $B \subseteq A^V$ be a set of visible actions. We define the I/O-IMC hide B in \mathcal{P} by $(S_{\mathcal{P}}, s_{\mathcal{P}}^0, (A_{\mathcal{P}}^I \setminus B, A_{\mathcal{P}}^O \setminus B, A_{\mathcal{P}}^{int} \cup B), \rightarrow_{\mathcal{P}}, \rightarrow_{\mathcal{P}}^M)$.

Definition 3 (Weak bisimulation). Let $P = \langle S, s^0, A, \rightarrow, \rightarrow^M \rangle$ be an I/O-IMC. Let R be an equivalence relation on S . Then R is a weak bisimulation iff for all $(s, t) \in R$, $a \in \text{Act}(P)$

1. $s \xrightarrow{a} s'$ implies that there is a weak transition $t \xRightarrow{a} t'$ with $(s', t') \in R$.
2. $s \xRightarrow{a} s'$ and s' stable imply that there is a t' such that $t \xRightarrow{a} t'$ and t' stable and $\gamma_M(s', C^{int}) = \gamma_M(t', C^{int})$, for all equivalence classes $C \in (S/R) \setminus \{[s']_R\}$

The states s and t in P are weakly bisimilar, notation $s \approx_P t$, if and only if there exists a weak bisimulation R with $(s, t) \in R$. Weak bisimilarity for an I/O-IMC P is defined as the union of all weak bisimulations on P : $\approx_P = \bigcup \{R \mid R \text{ is a weak bisimulation on } P\}$. We often omit the name of the I/O-IMC if it is clear from context.

Here, we use the following notation. $\gamma_M(s, C) = \sum \{|\lambda \mid s \xrightarrow{\lambda} s' \wedge s' \in C|\}$, where $\{|\dots|\}$ denotes a multiset of transition rates; $C^{int} = \{s' \mid \exists s \in C \cdot s' \xRightarrow{} s\}$, $\xRightarrow{} is the reflexive, transitive closure of \xrightarrow{int} where $\xrightarrow{int} = \{(s, t) \mid (s, a, t) \in \rightarrow \wedge a \in A^{int}\}$; and $s \xRightarrow{a} s' \leftrightarrow (a \in A^V \wedge \exists t, t' \cdot s \xRightarrow{a} t \xrightarrow{a} t' \xRightarrow{} s') \vee (a \in A^{int} \wedge s \xRightarrow{} s')$.$

Preservation Theorem (WB is a congruence)

Theorem 1. *Weak bisimilarity for an I/O-IMC is the largest weak bisimulation for that I/O-IMC. Weak bisimilarity is also a congruence with respect to parallel composition and hiding. Let \mathcal{P} and \mathcal{Q} be two I/O-IMCs with identical action signatures, let \mathcal{R} be an I/O-IMC composable with \mathcal{P} and \mathcal{Q} and let $B \subseteq A_{\mathcal{P}}^V$:*

1. $\approx_{\mathcal{P}}$ is a weak bisimulation on \mathcal{P} and it is the largest weak bisimulation on \mathcal{P} ,
2. $\mathcal{P} \approx \mathcal{Q}$ implies $\mathcal{P} \parallel \mathcal{R} \approx \mathcal{Q} \parallel \mathcal{R}$,
3. $\mathcal{P} \approx \mathcal{Q}$ implies $\mathcal{R} \parallel \mathcal{P} \approx \mathcal{R} \parallel \mathcal{Q}$,
4. $\mathcal{P} \approx \mathcal{Q}$ implies $\text{hide } B \text{ in } \mathcal{P} \approx \text{hide } B \text{ in } \mathcal{Q}$.