Technische Universität Braunschweig

ISF

Family Mining on Statecharts

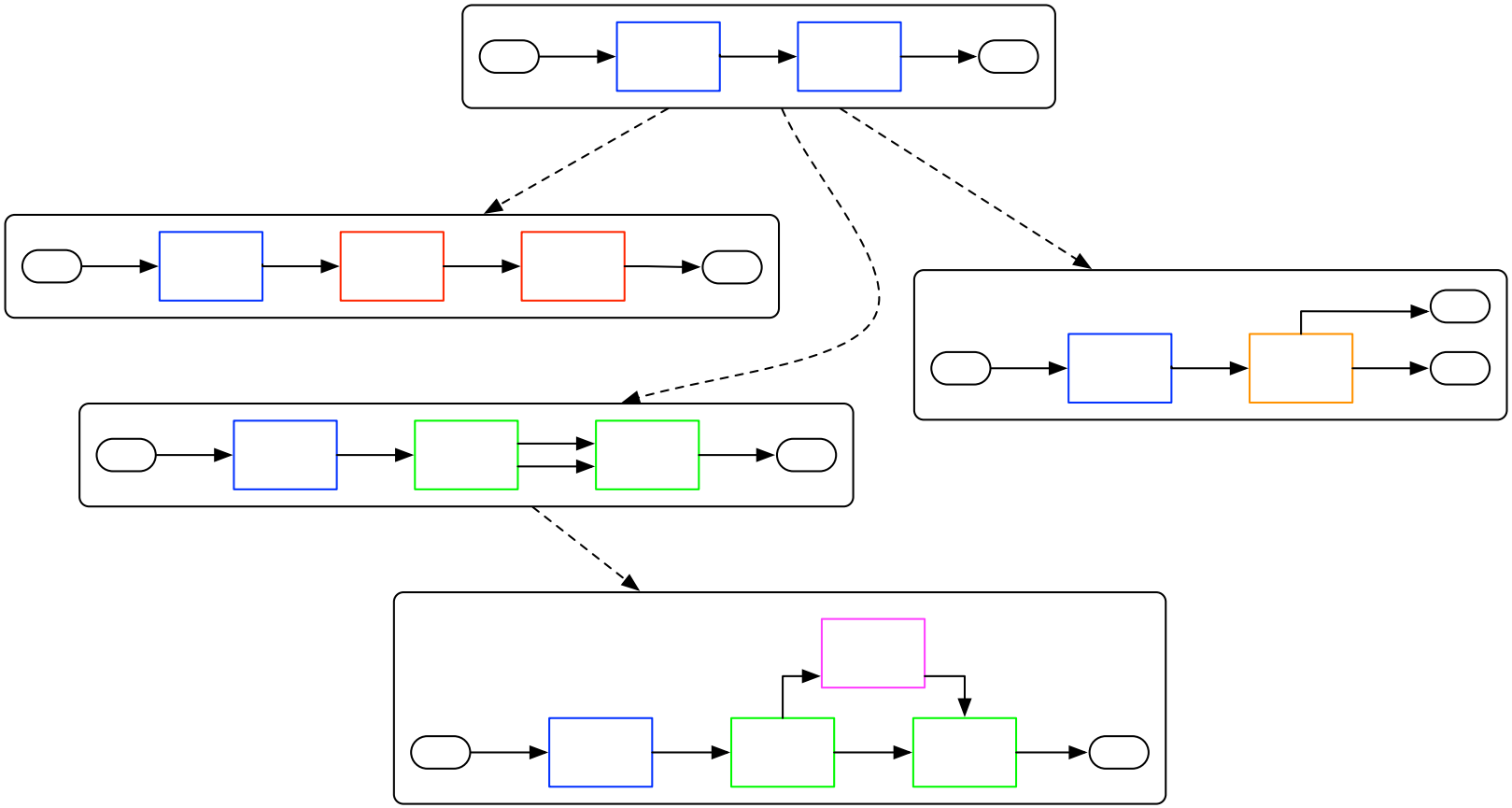
Master's thesis ideas

David Wille, May 5, 2014

Overview

- Motivation
- Background
- Current approach
- Ideas

Motivation

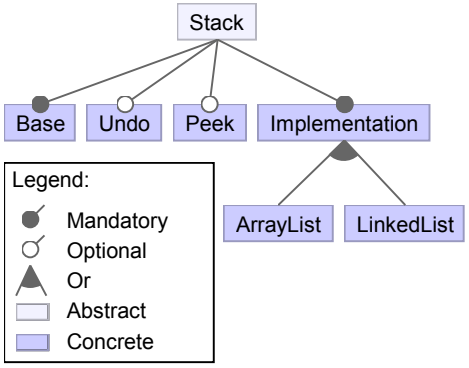


Identifying differences and commonalities is crucial!

Family Models vs. Feature Models

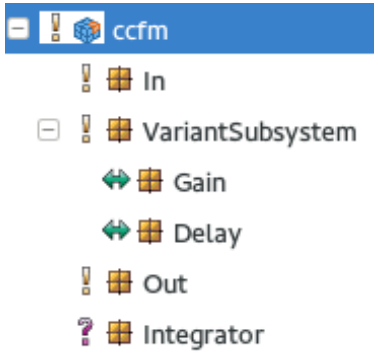
Feature Models

- *problem domain*
- only models variability
- without further details



Family Models

- *solution space*
- concrete design
- implementation details



Creating Compare Elements

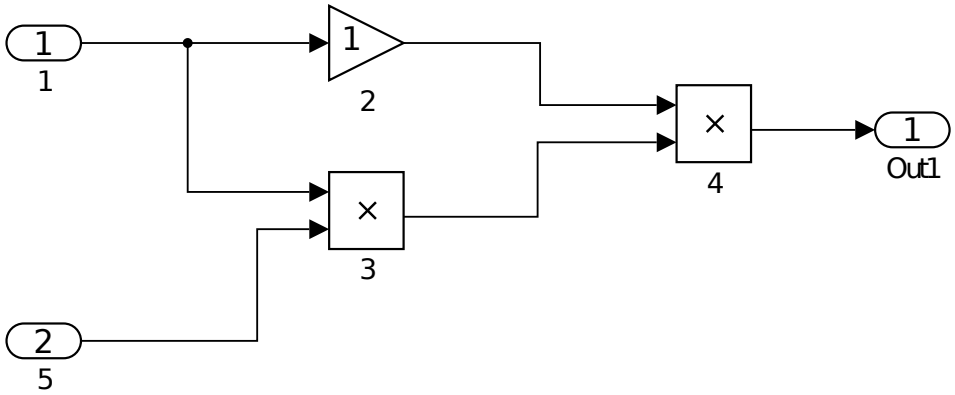


Figure 1: Variant 1

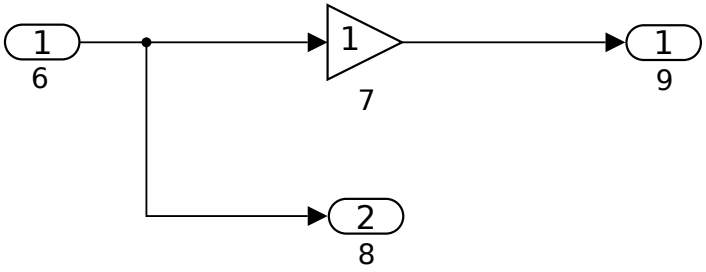


Figure 2: Variant 2

Creating Compare Elements

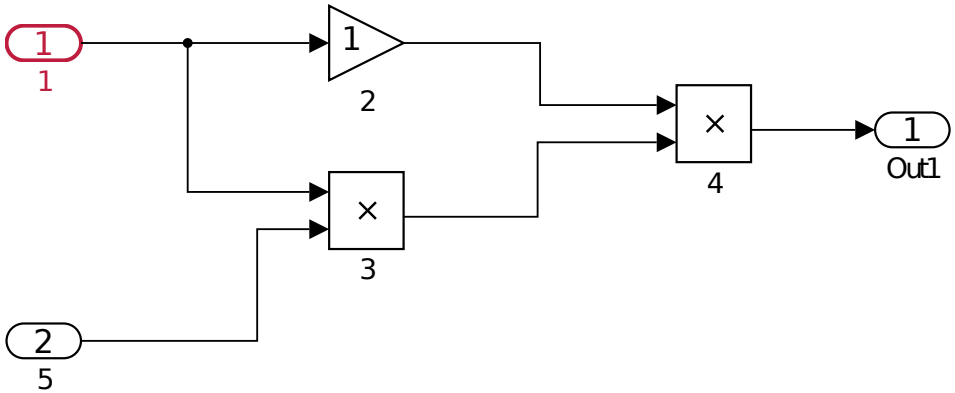


Figure 1: Variant 1

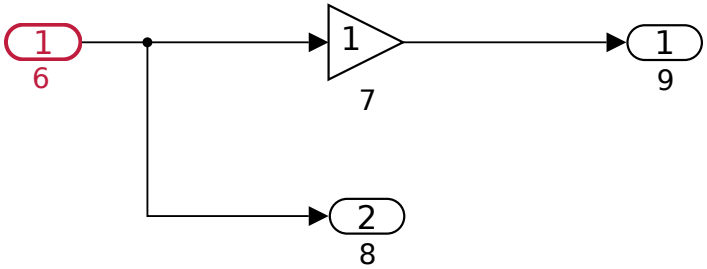
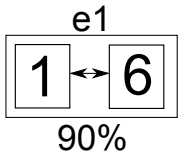


Figure 2: Variant 2



Creating Compare Elements

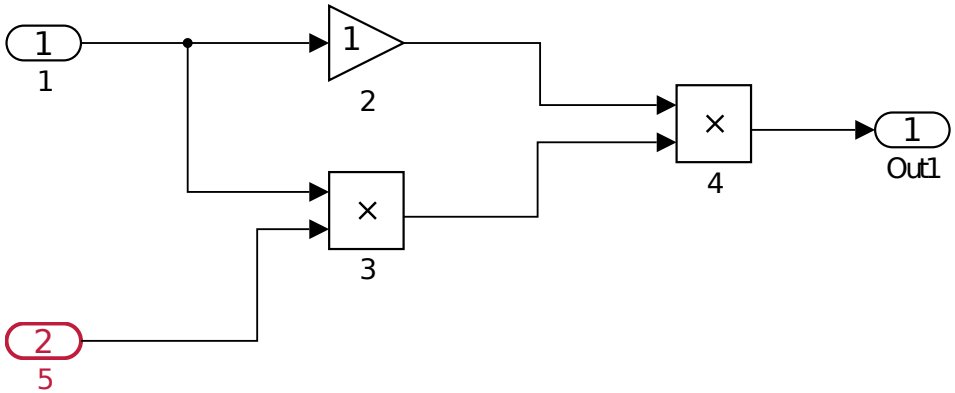


Figure 1: Variant 1

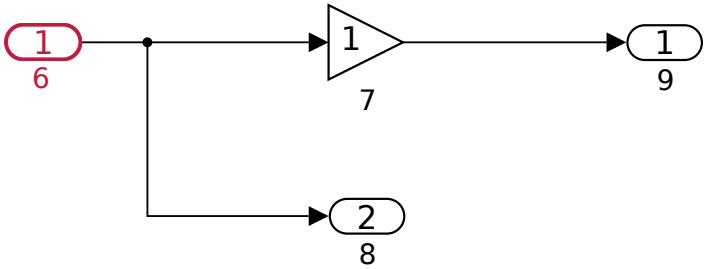
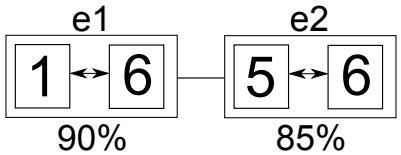


Figure 2: Variant 2



Creating Compare Elements

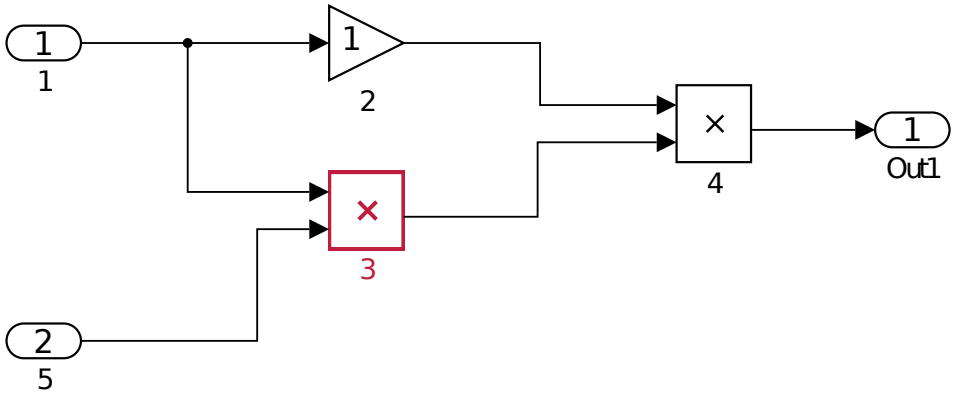


Figure 1: Variant 1

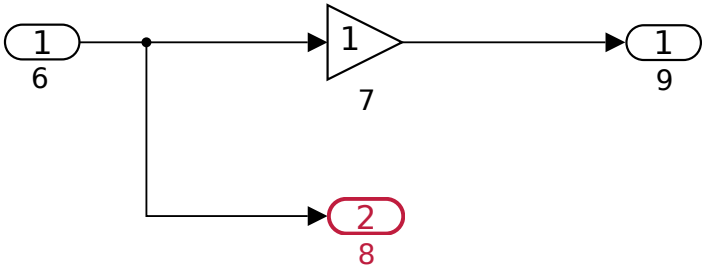
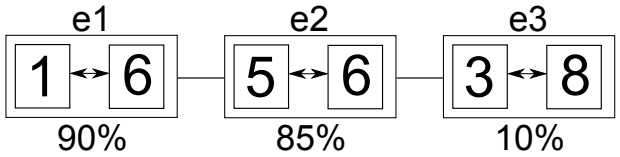


Figure 2: Variant 2



Creating Compare Elements

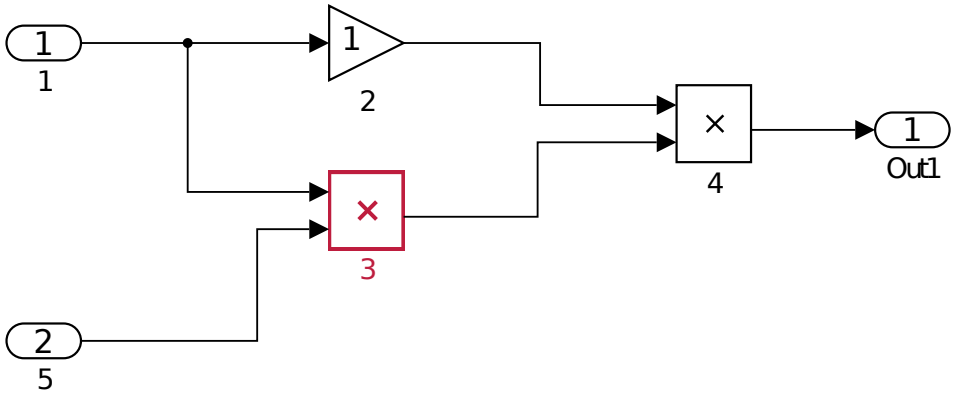


Figure 1: Variant 1

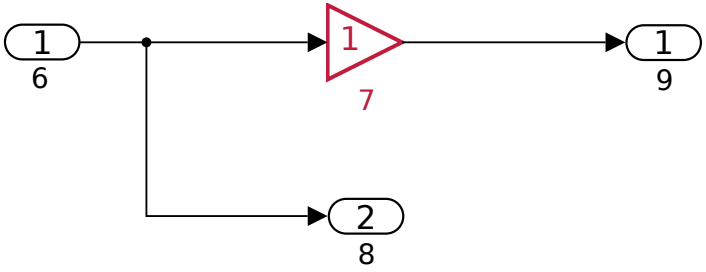
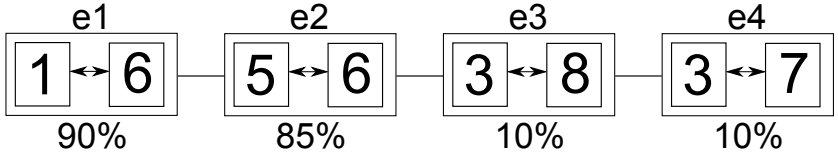


Figure 2: Variant 2



Creating Compare Elements

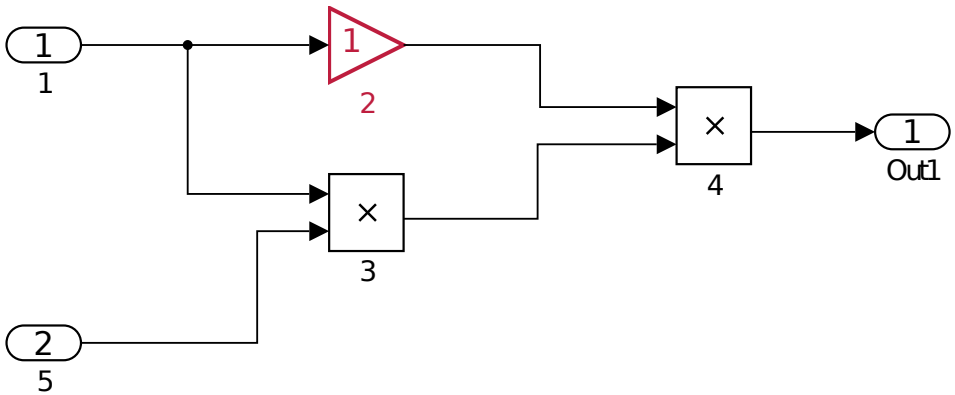


Figure 1: Variant 1

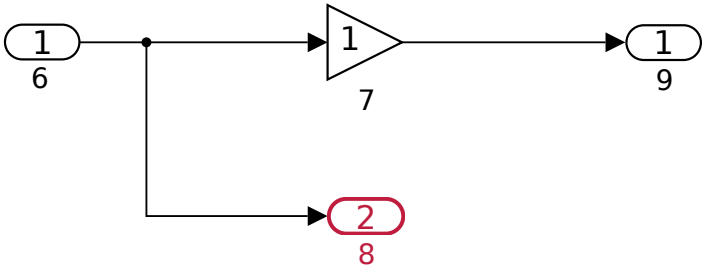
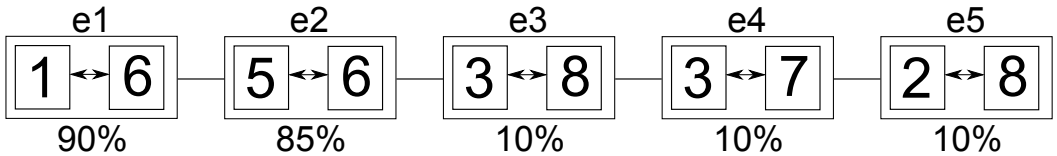


Figure 2: Variant 2



Creating Compare Elements

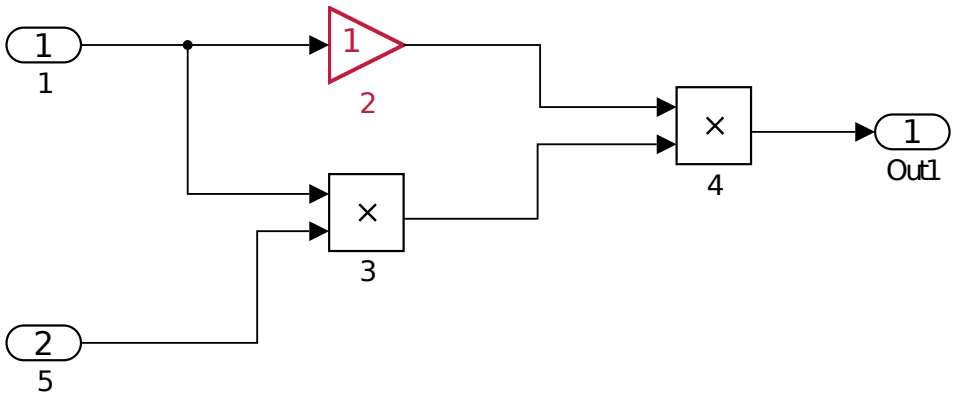


Figure 1: Variant 1

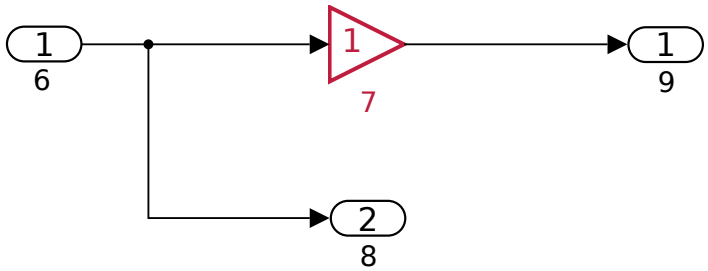
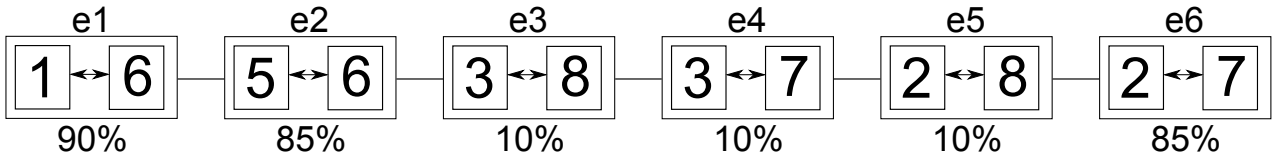


Figure 2: Variant 2



Creating Compare Elements

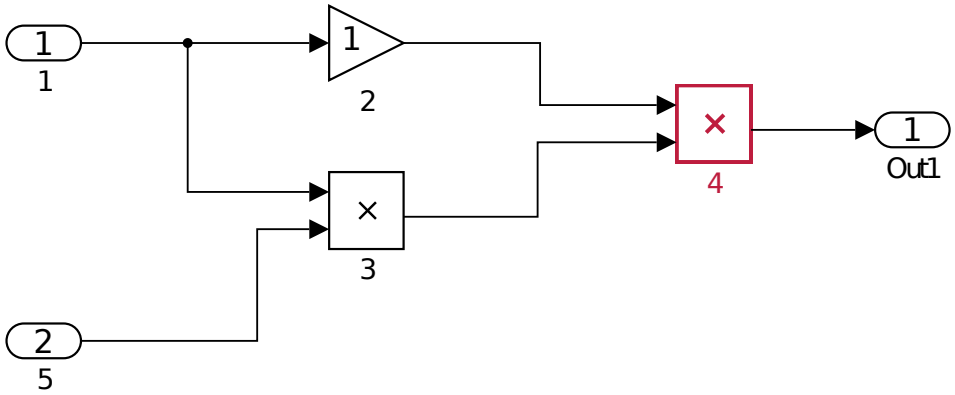


Figure 1: Variant 1

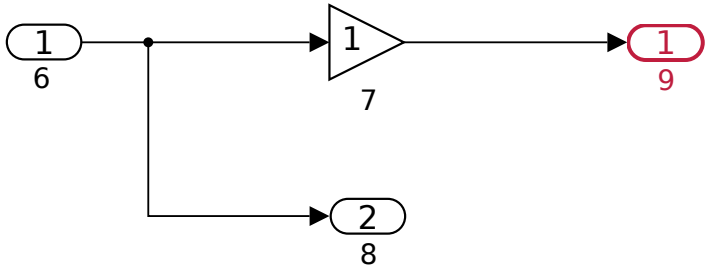
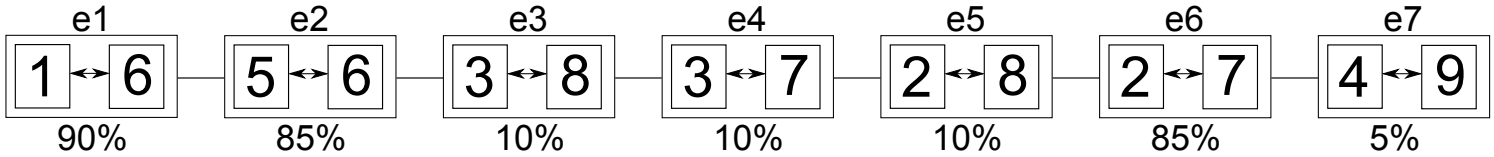


Figure 2: Variant 2



Creating Compare Elements

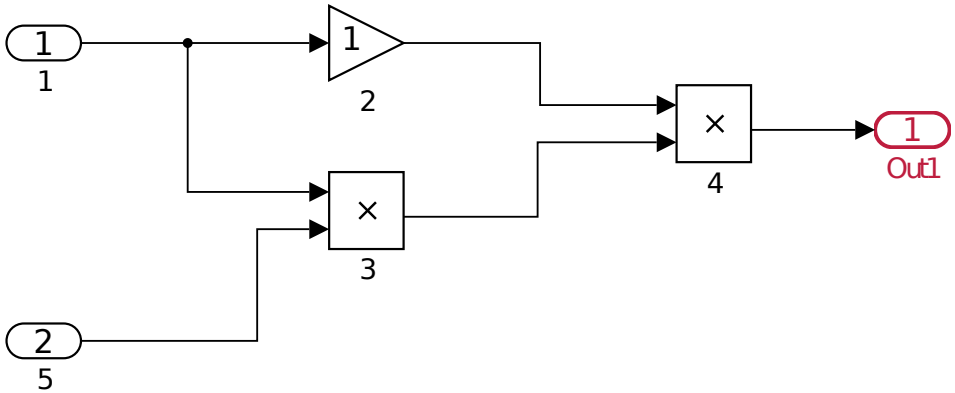


Figure 1: Variant 1

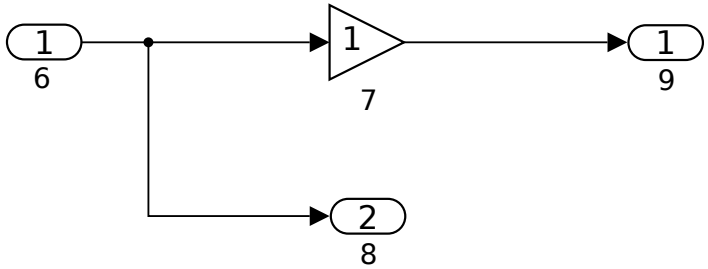
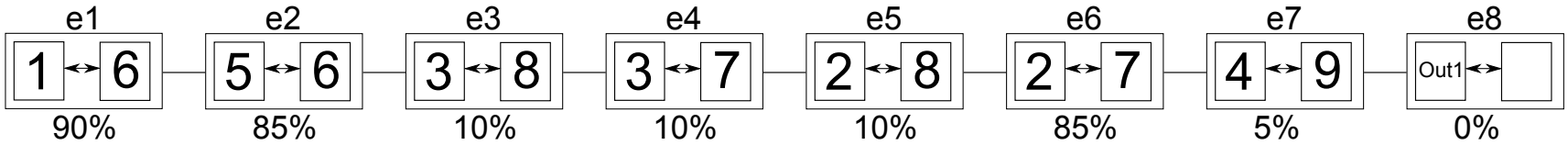
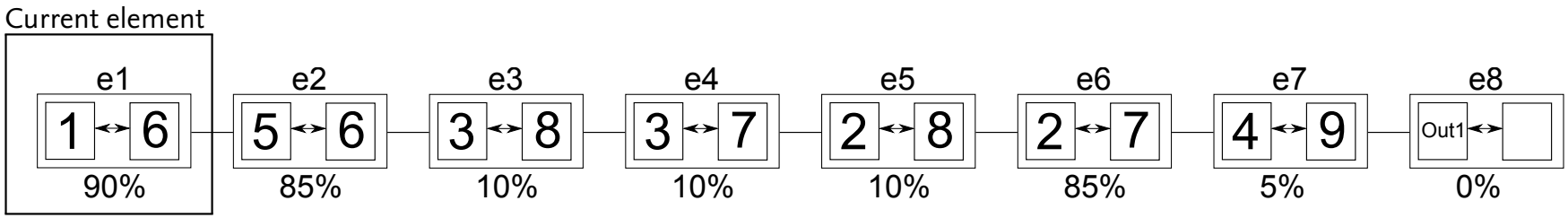


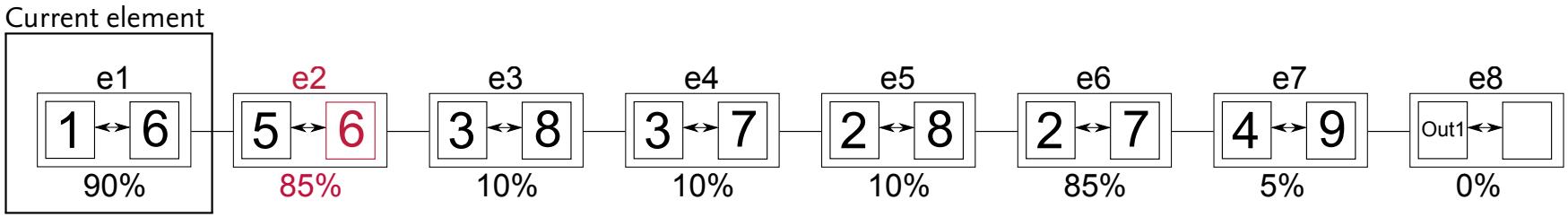
Figure 2: Variant 2



Matching

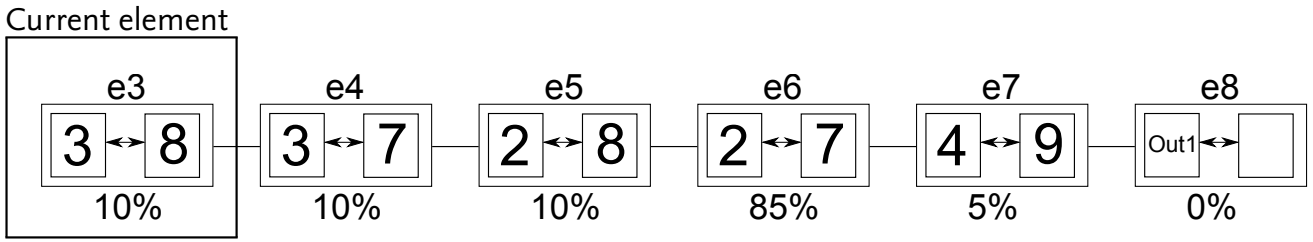

Current element
e1
1
6
90%
e2
5
6
85%
e3
3
8
10%
e4
3
7
10%
e5
2
8
10%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%

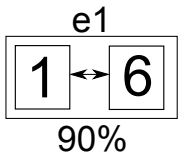
Matching



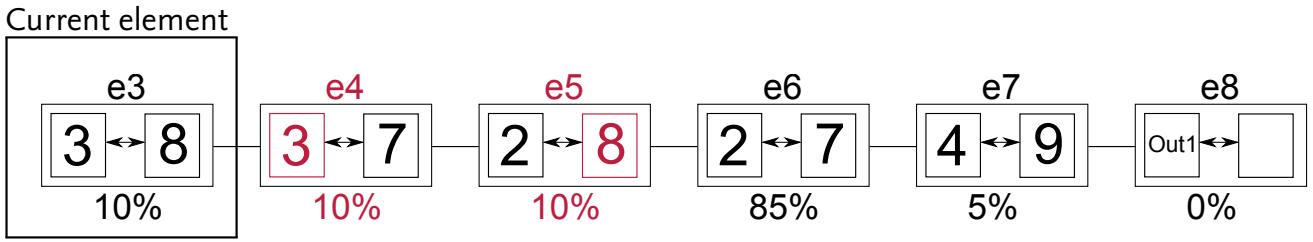
Found element e2 with same block, but less similarity. So e1 is optimal.

Matching

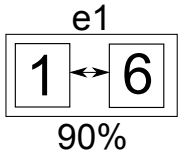

Current element
e3
3 ↔ 8
10%
e4
3 ↔ 7
10%
e5
2 ↔ 8
10%
e6
2 ↔ 7
85%
e7
4 ↔ 9
5%
e8
Out1 ↔
0%


e1
1 ↔ 6
90%

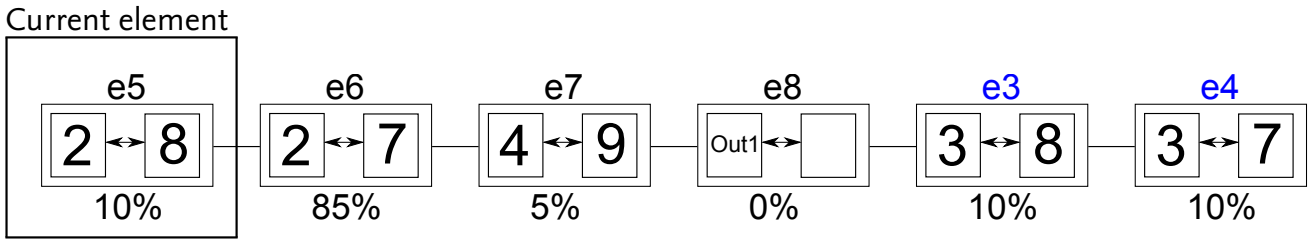
Matching



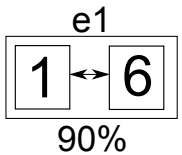
Found element e4 and e5. All have similarity of 10% so e3 and e4 are ambiguous.



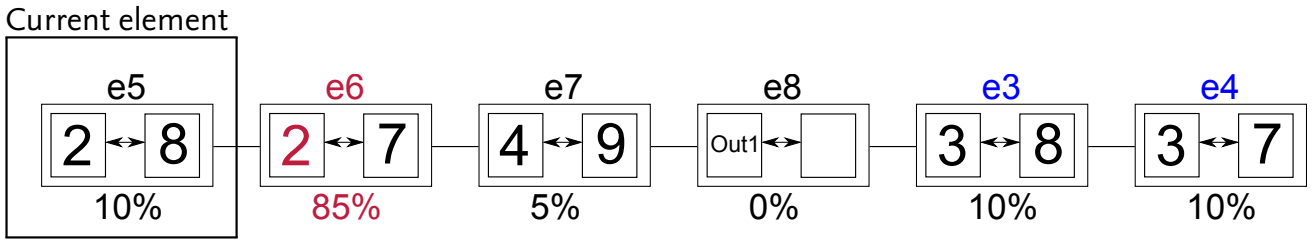
Matching



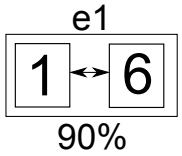
Sorted e3 and e4 to the end of list.



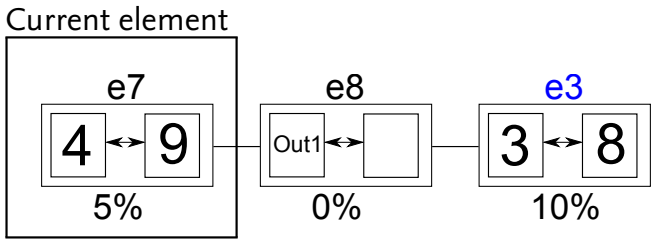
Matching

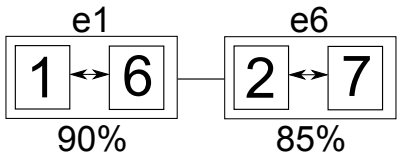


Found a better match. e6 > e5.

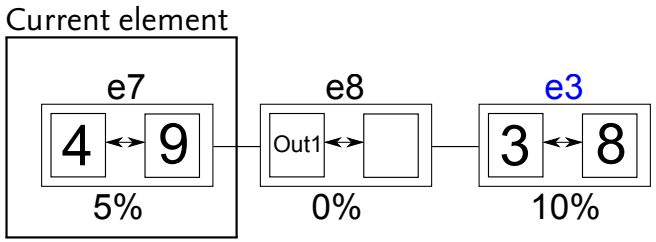


Matching

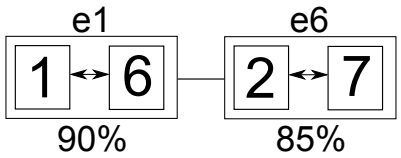




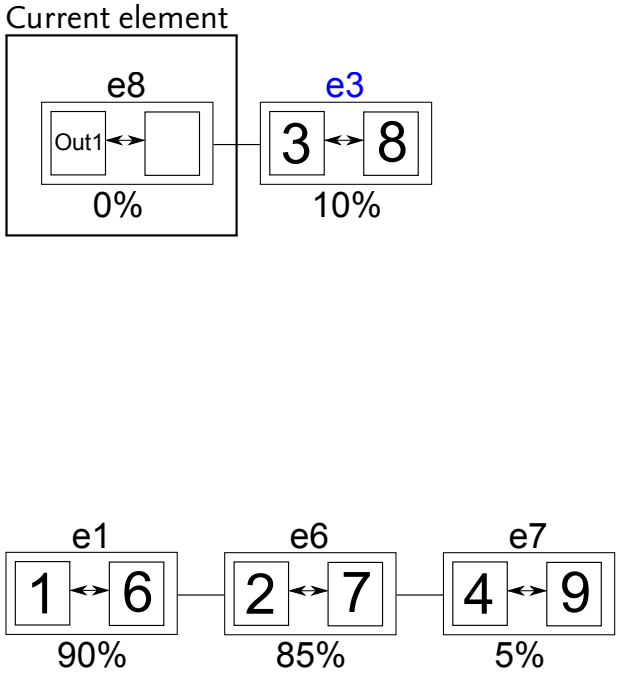
Matching



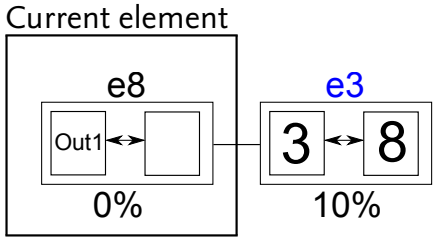
No better match found. e7 is optimal.



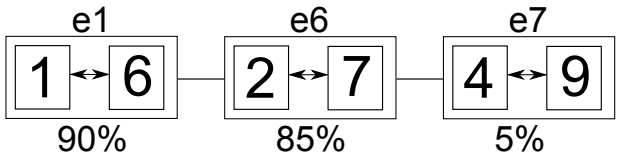
Matching



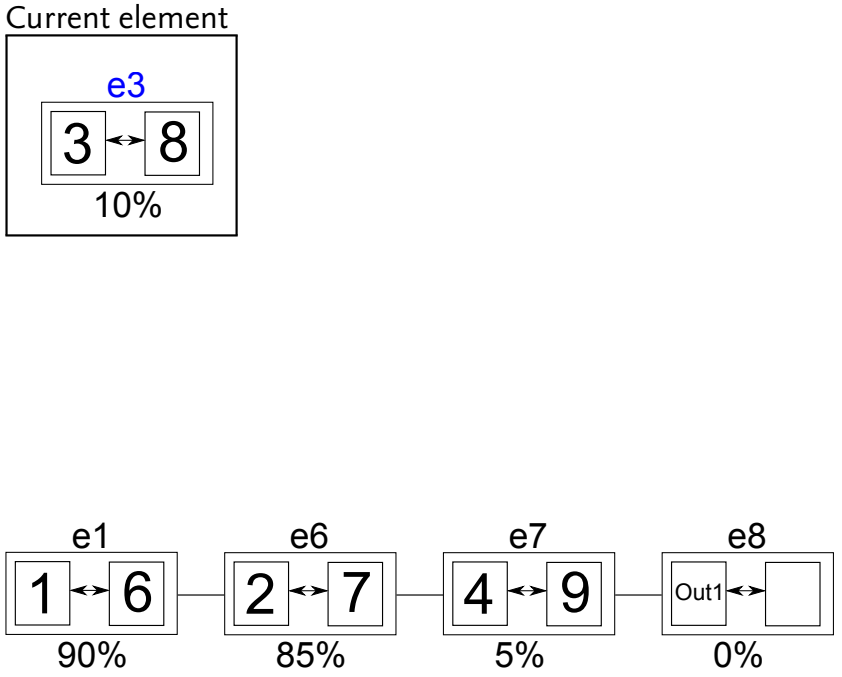
Matching



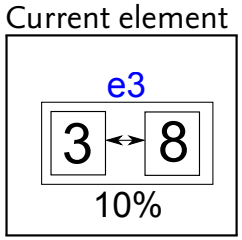
No better match found. e8 is optimal.



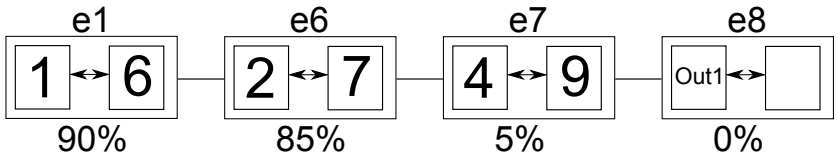
Matching



Matching

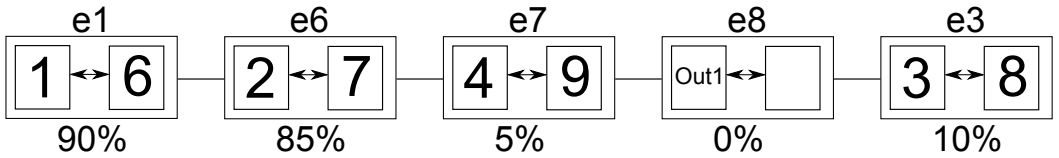


No better match found. e3 is optimal.



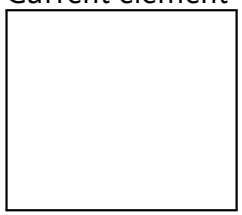
Matching

Current element

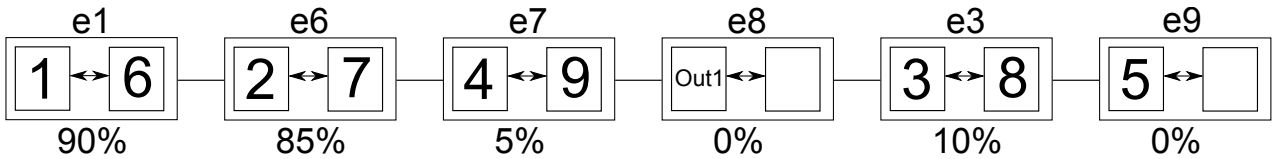


Matching

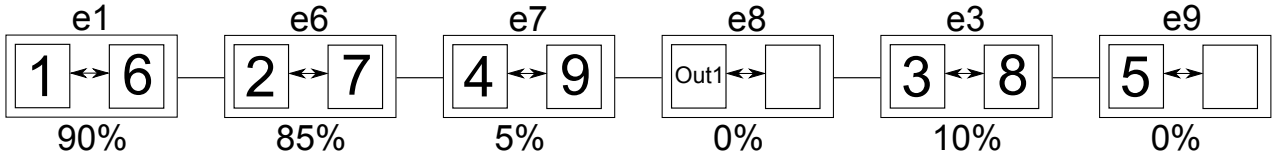
Current element

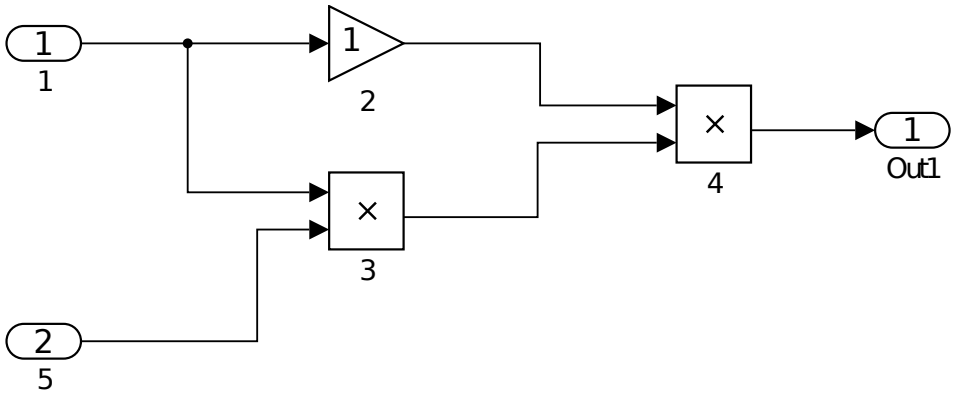


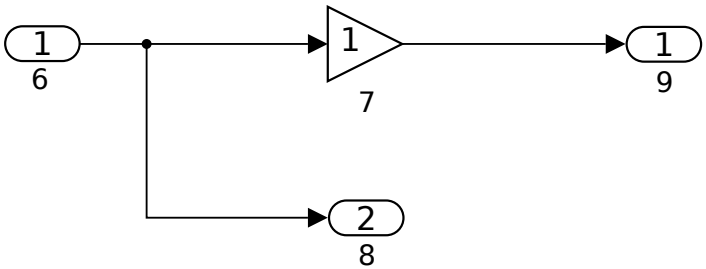
Block 5 was not yet considered, so it is an optional block.



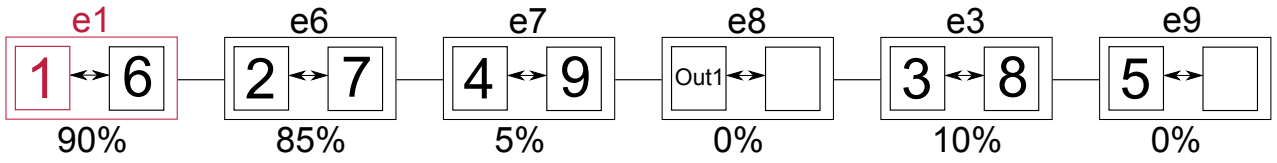
Merging

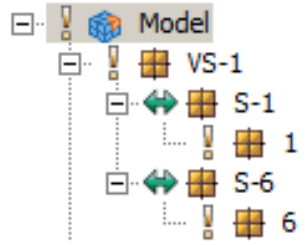

e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%

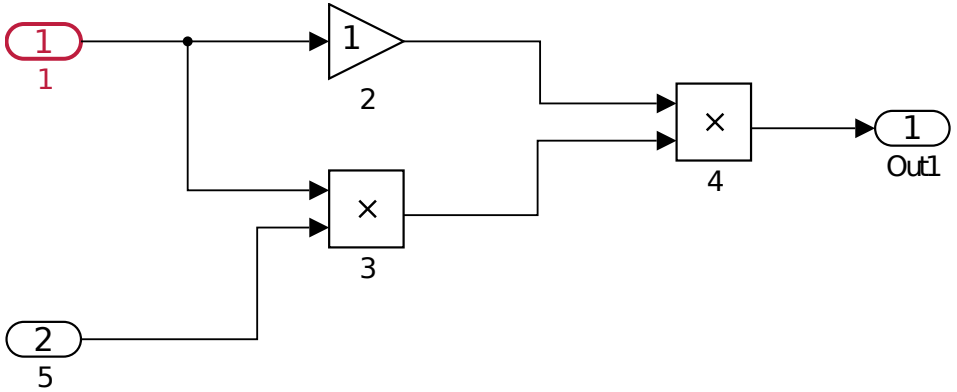

1
1
1
2
×
4
1
Out1
×
3
2
5

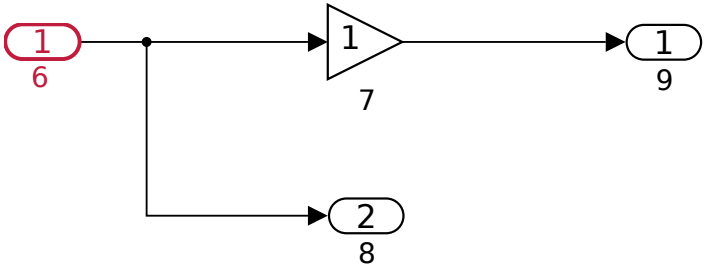

1
6
1
7
1
9
2
8

Merging

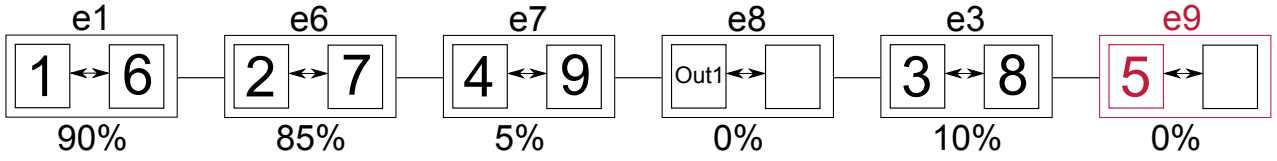

e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%

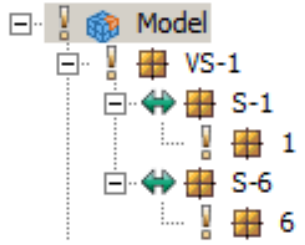

Model
VS-1
S-1
1
S-6
6

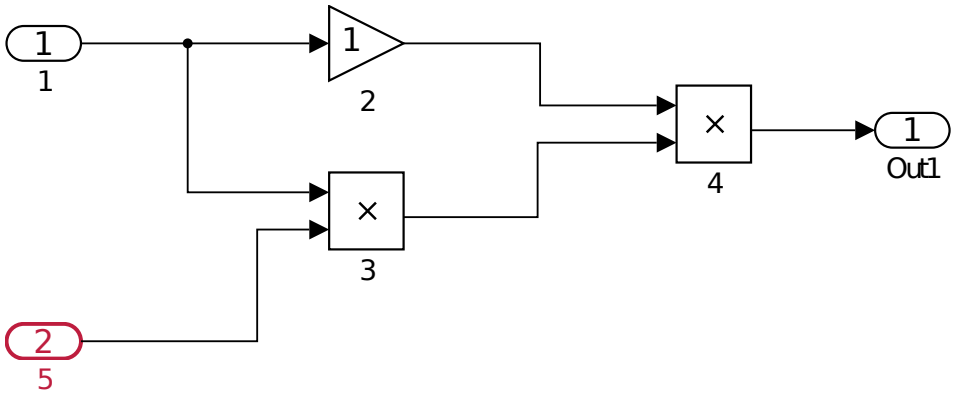

1
1
1
2
×
4
1
Out1
×
3
2
5

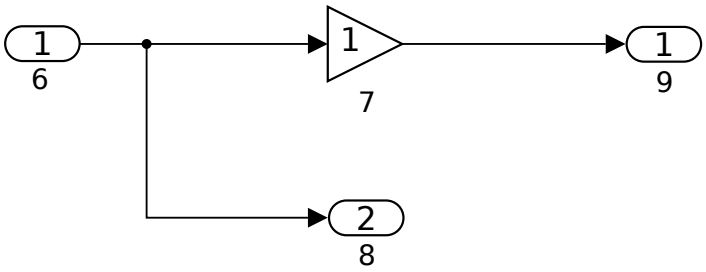

1
6
1
7
1
9
2
8

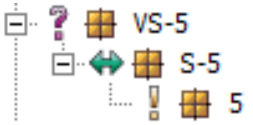
Merging


e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%

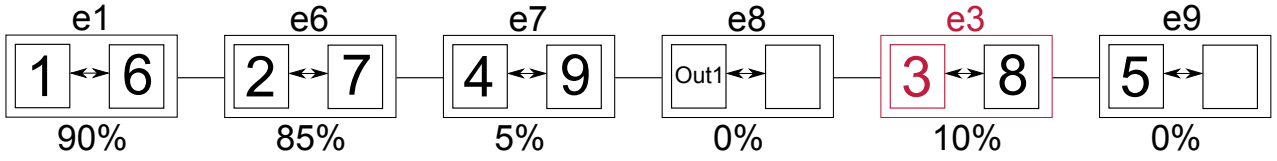

Model
VS-1
S-1
1
S-6
6

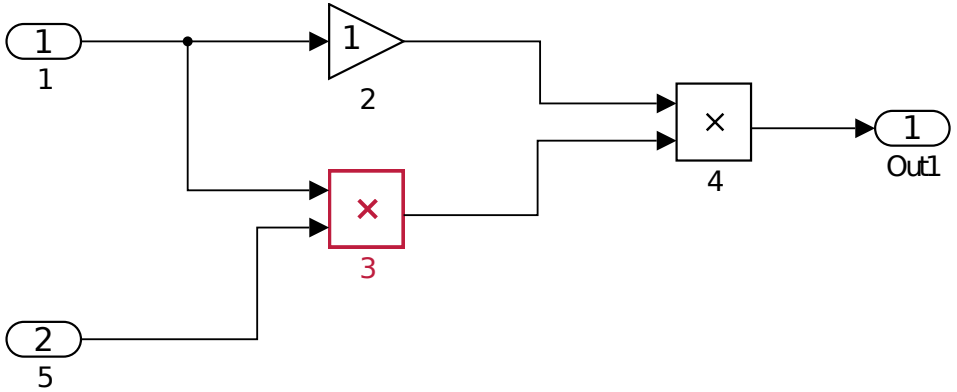

1
1
1
2
×
4
1
Out1
×
3
2
5

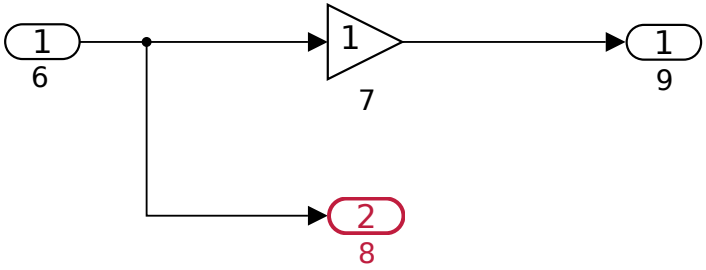

1
6
1
7
1
9
2
8

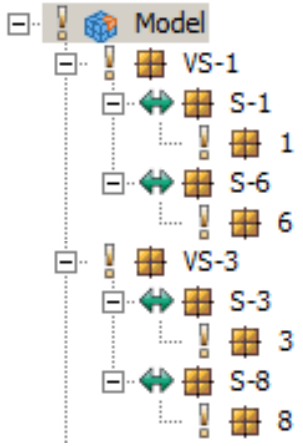

VS-5
S-5
5

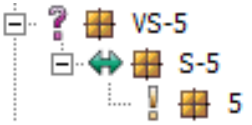
Merging


e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%

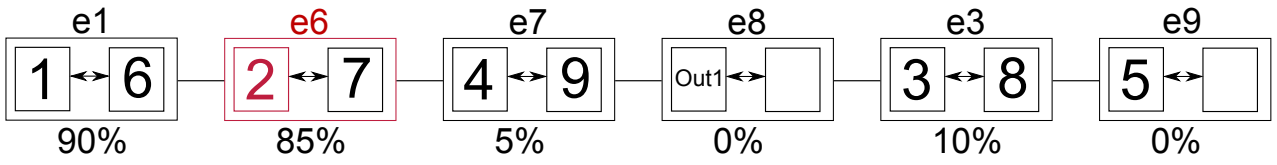

1
1
1
2
×
4
1
Out1
×
3
2
5

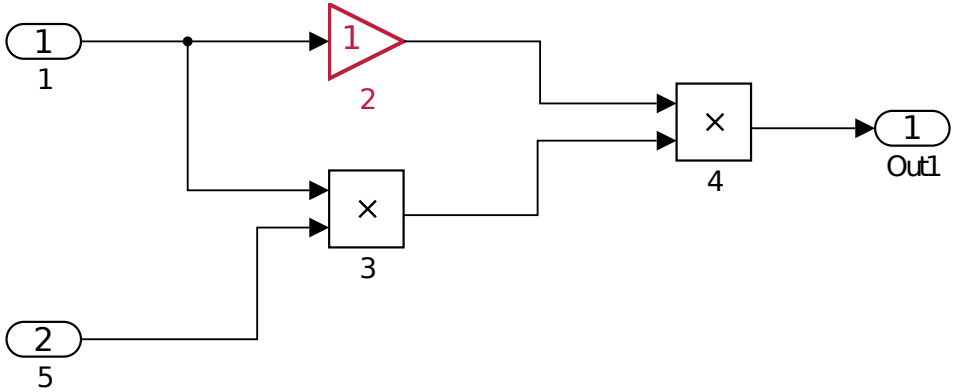

1
6
1
7
1
9
2
8

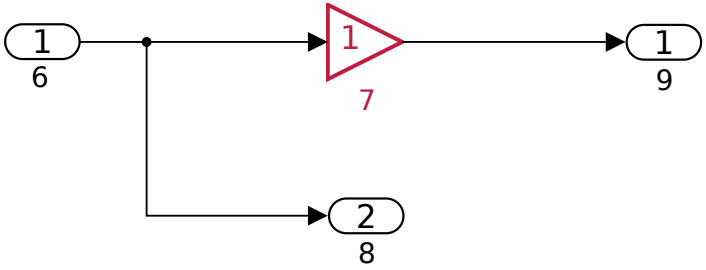

Model
VS-1
S-1
1
S-6
6
VS-3
S-3
3
S-8
8

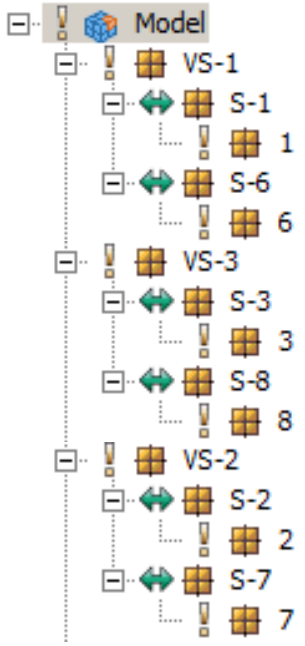

VS-5
S-5
5

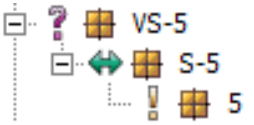
Merging


e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%

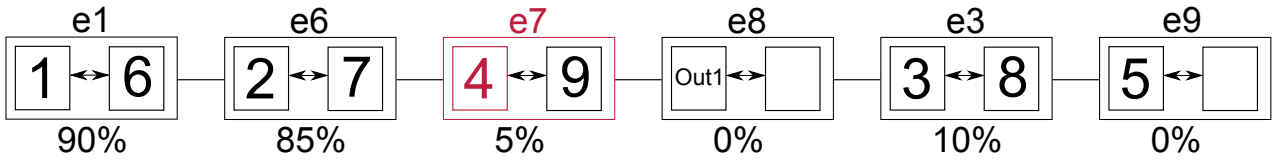

1
1
1
2
×
3
×
4
1
Out1
2
5

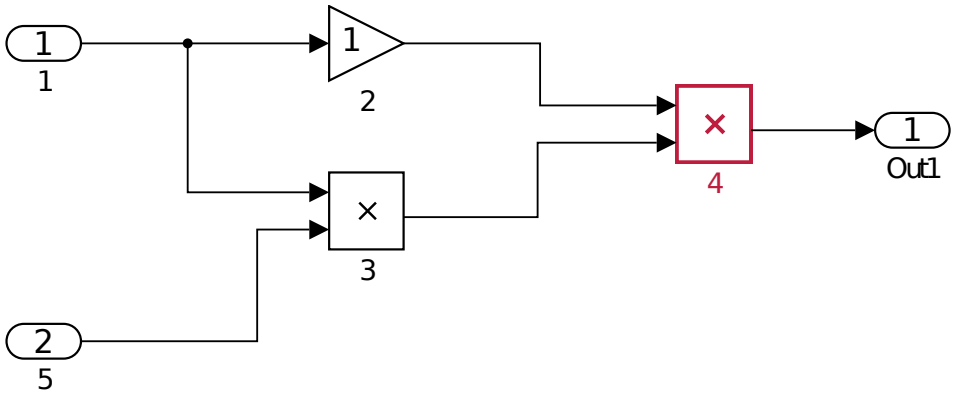

1
6
1
7
1
9
2
8

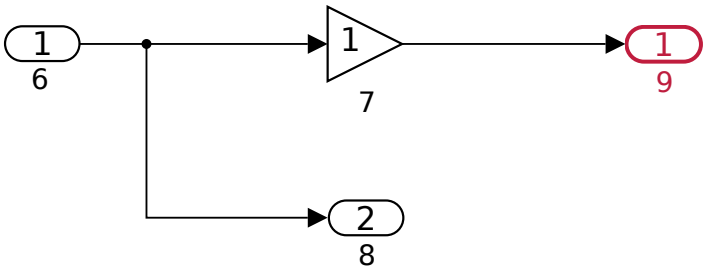

Model
VS-1
S-1
1
S-6
6
VS-3
S-3
3
S-8
8
VS-2
S-2
2
S-7
7

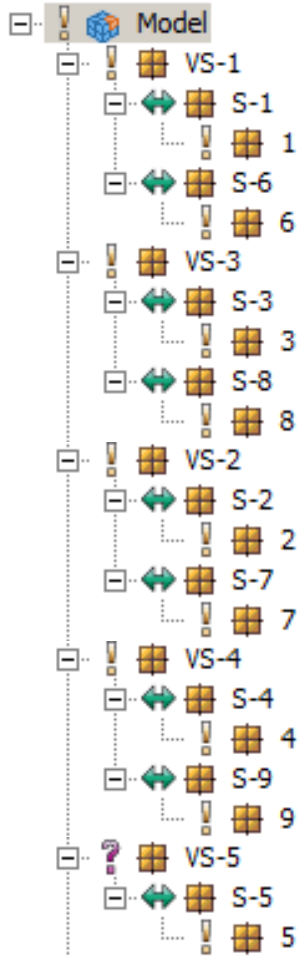

VS-5
S-5
5

Merging

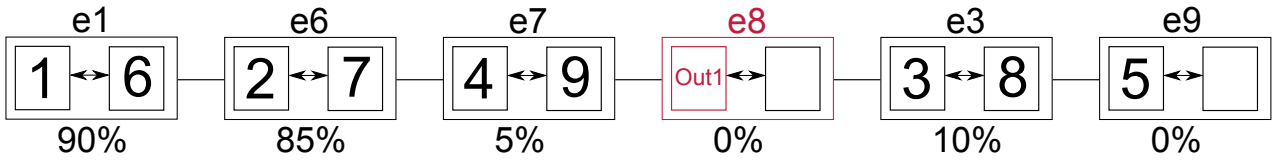

e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%

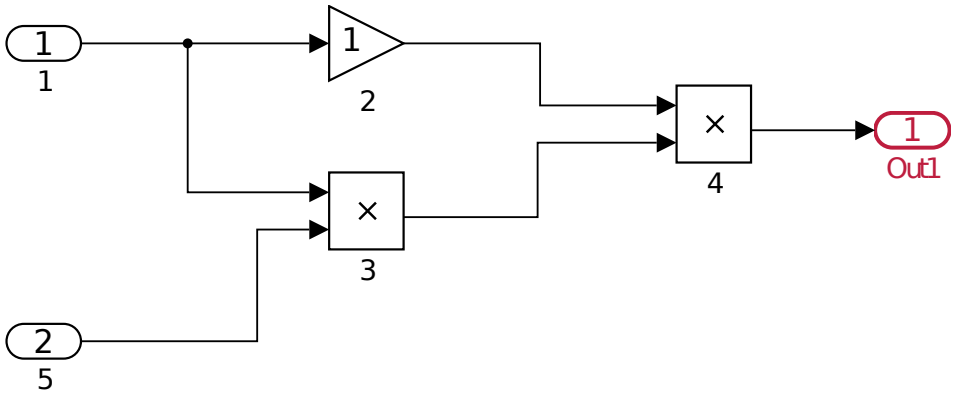

1
1
1
2
×
4
1
Out1
×
3
2
5

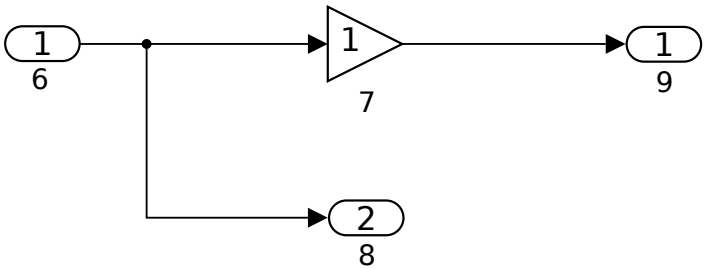

1
6
1
7
1
9
2
8

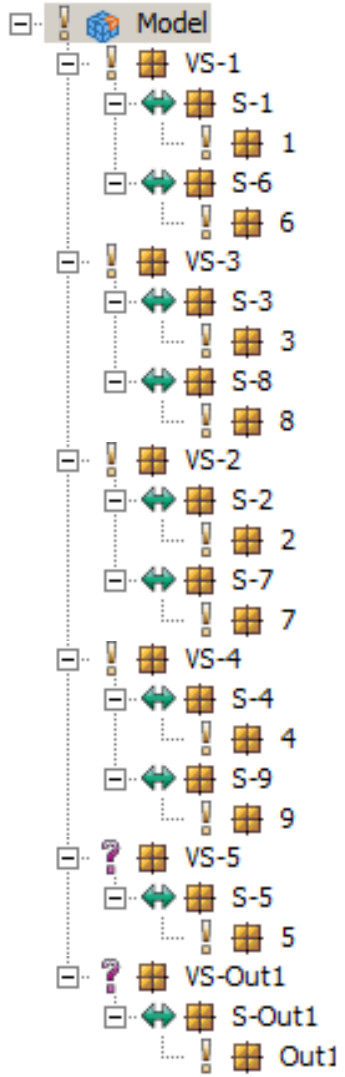

Model
VS-1
S-1
1
S-6
6
VS-3
S-3
3
S-8
8
VS-2
S-2
2
S-7
7
VS-4
S-4
4
S-9
9
VS-5
S-5
5

Merging


e1
1
6
90%
e6
2
7
85%
e7
4
9
5%
e8
Out1
0%
e3
3
8
10%
e9
5
0%


1
1
1
2
×
4
1
Out1
×
3
2
5


1
6
1
7
1
9
2
8


Model
VS-1
S-1
1
S-6
6
VS-3
S-3
3
S-8
8
VS-2
S-2
2
S-7
7
VS-4
S-4
4
S-9
9
VS-5
S-5
5
VS-Out1
S-Out1
Out1

Summary

What has been done?

- Approach applied to MATLAB/Simulink models
- Create family models:
  - Understand relations between compared models
  - Improves maintainability of models

Current work

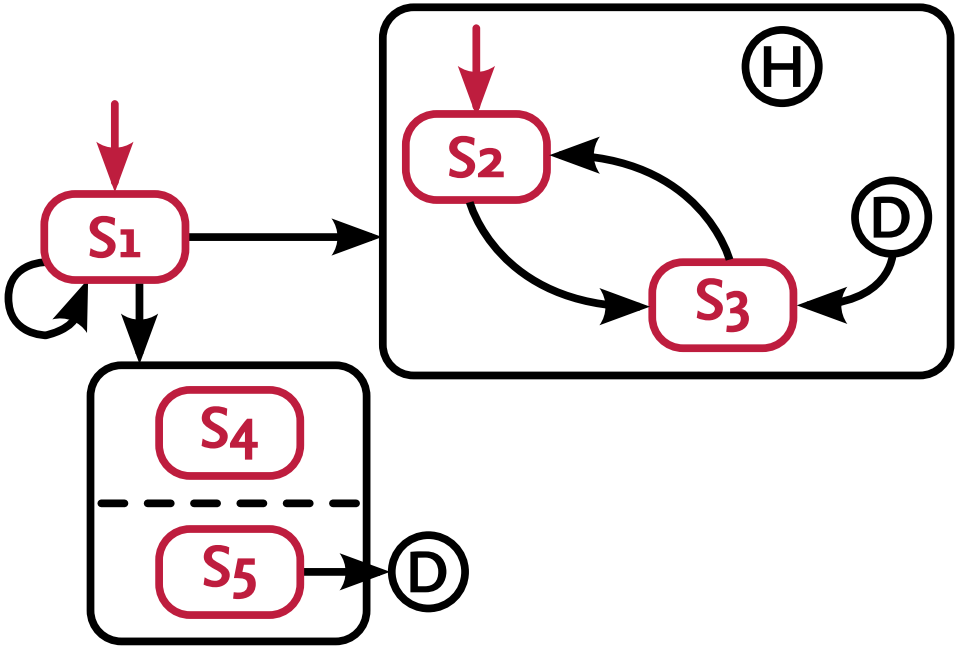
- Validation of the approach with industrial scale models:
  - The general approach, the metric, ...
- Logging for large models
- Refactoring in order to support multiple block-based languages:
  - e.g., MATLAB/Simulink, CoDeSys, ...

Analysis

- Analyze statechart representations of the following tools:
  - Esterel Technologies *SCADE*
  - Math Works *Stateflow*
  - ETAS *ASCET*
  - IBM *Rational Rhapsody*
- Also consider the journal article by *Harel*[1]

[1] David Harel, Statecharts: a visual formalism for complex systems, 1987

Identified concepts



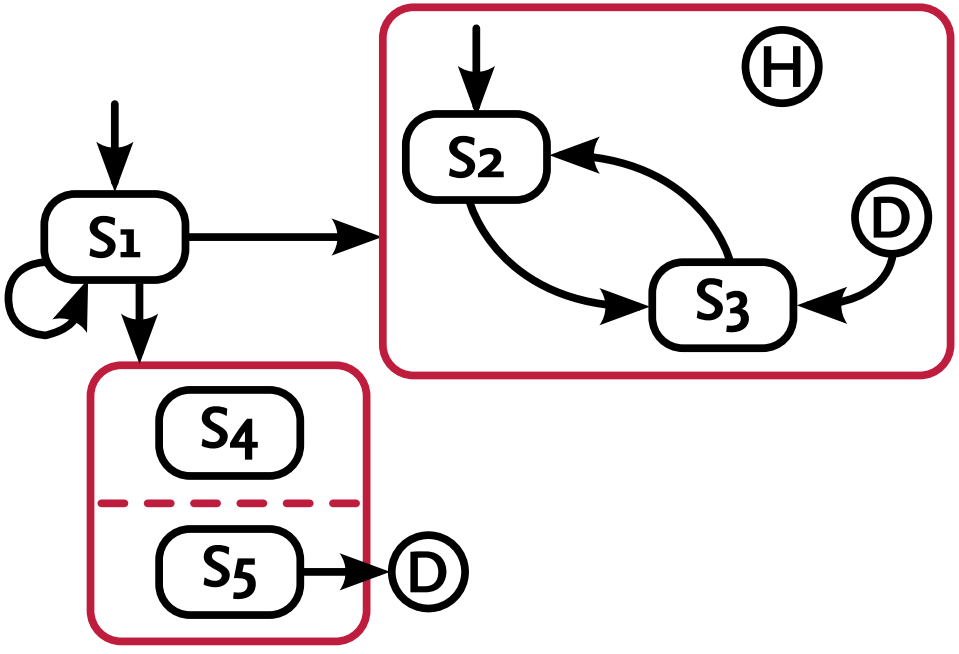
States

- Represented as a box
- Name is distinctive
- State actions:
  - e.g., entry, exit, during, ...
  - in *SCADE* defined by block-based models

Start States

- Different notations:
  - Marked with “S”
  - Default transition

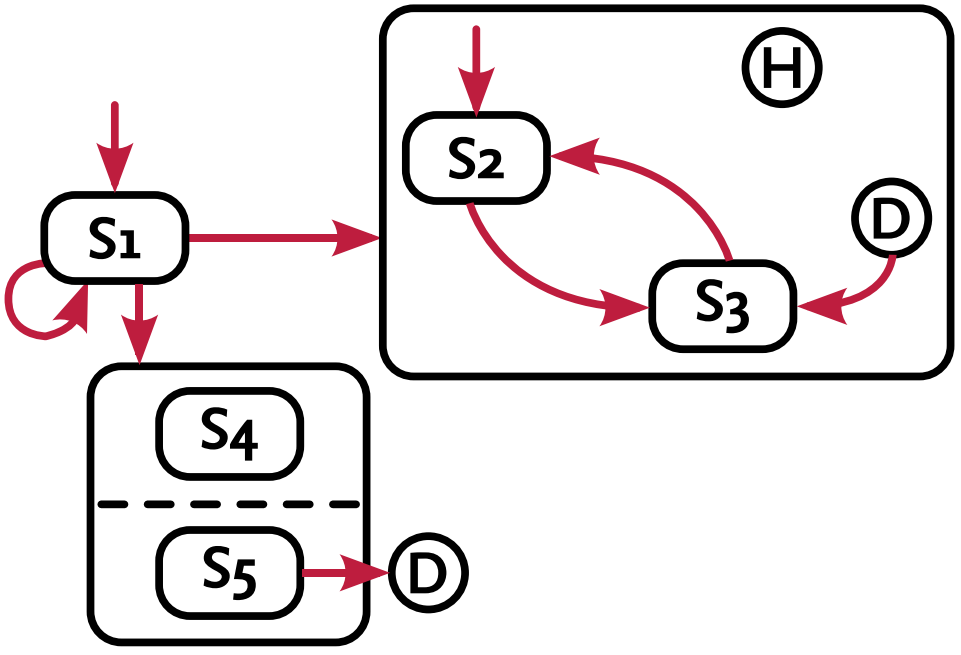
Identified concepts



Special States

- Hierarchical states
- Parallel states

Identified concepts



Transitions

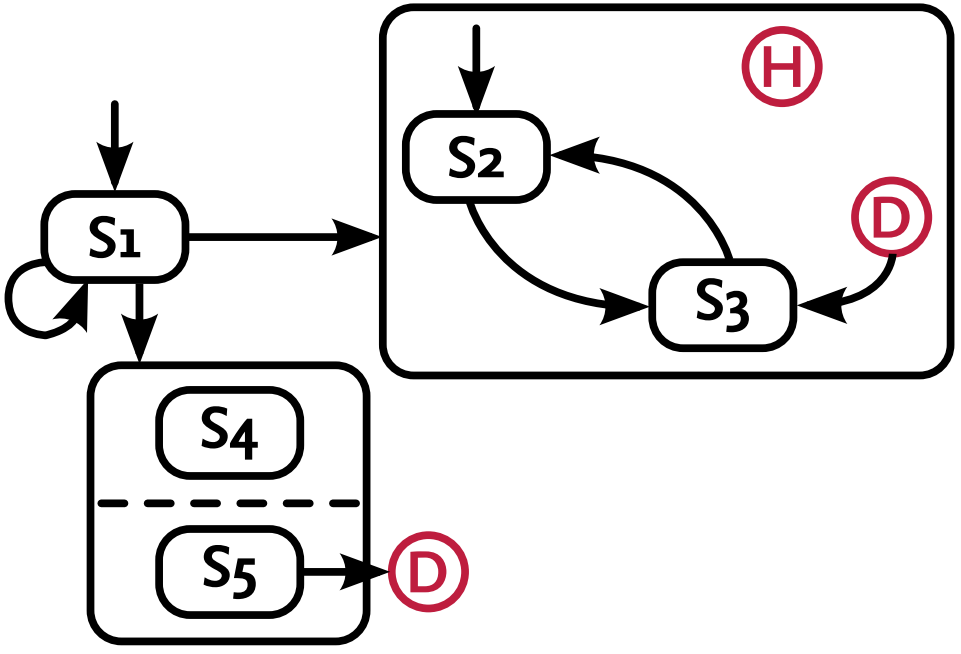
event [condition] / action

- *event:* triggering the transition
- *condition:* e.g., `x < 2`
- *action:* e.g., `x = 2`

Special Transitions

- Self-Loops
- Junctions
- Spontaneous transitions

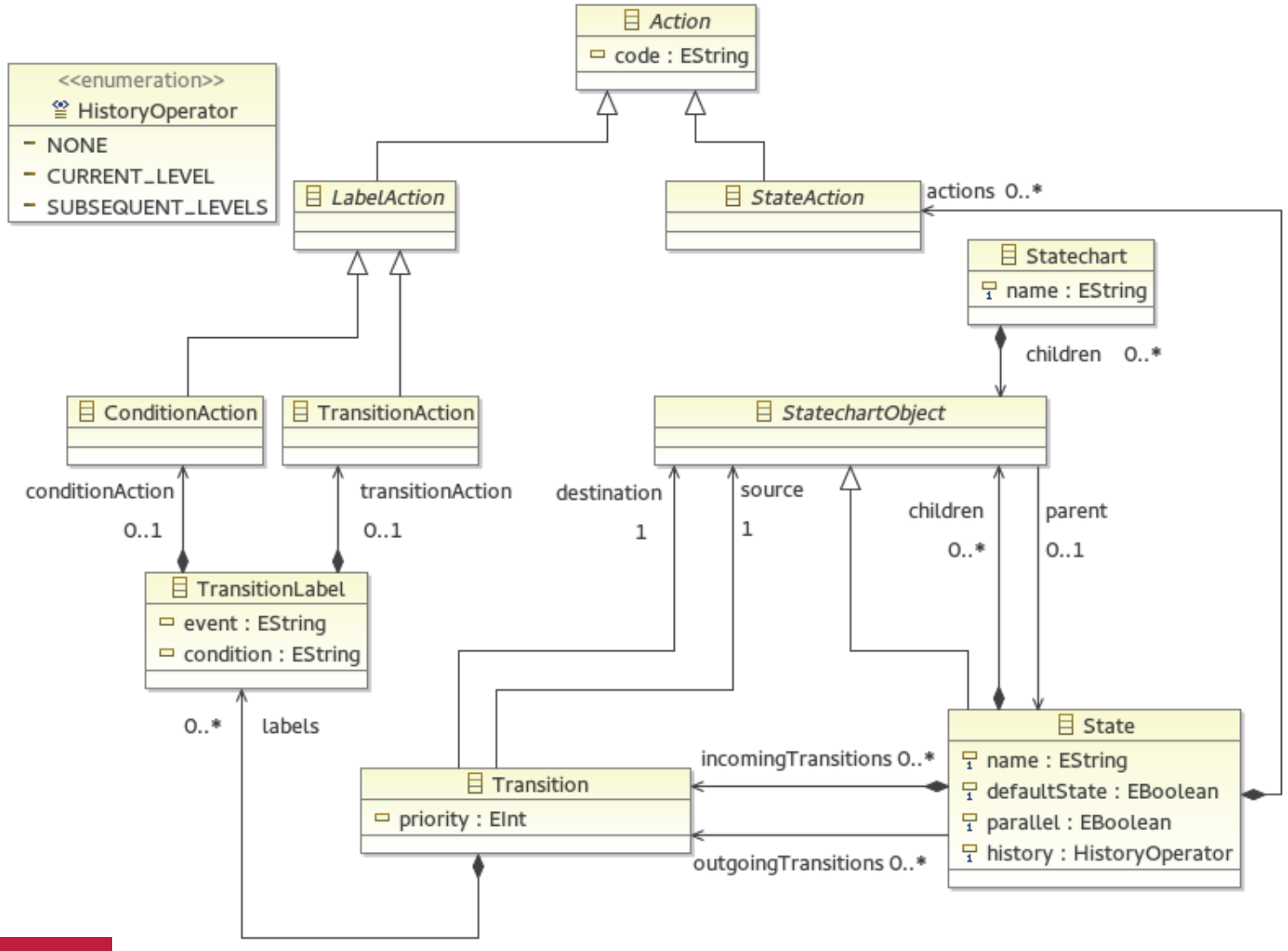
Identified concepts



Special concepts

- History junctions
- Diagram connectors “goto”
- Forks / Joins for parallel states

Create a Meta-Model



Identified challenges

No distinct attributes

- Names can change
- All elements have the same type (“state”) in contrast to Simulink with different types (e.g., Gain, Sum, Product, ...)
- Vocabulary can change between models (e.g., different names for the same event)
- Actions are defined by code and not by unique block types

$\Rightarrow$ Find a way to compare ambiguous elements and identify relations

Next steps: Approach & Evaluation

Metric

- Find a suitable metric to identify the variability of:
  - States
  - Transitions
- Should work with hierarchies and differing interfaces

Approach

- Find an approach with following requirements:
  - Efficient (preferably no $n \times m$ comparison)
  - Correct (i.e., meeting the stakeholders' expectations)
- Should work with all identified concepts

Next steps: Approach & Evaluation

Evaluation

- Extend the *architecture generator* developed at the ISF to generate
  - ... different related statecharts.
  - ... statecharts with differing complexity and size.
- Use the generated statecharts to evaluate the results of the approach

Thank you!

Thank you for your attention!