Improved Parallel Composition

A. Alekseyev\textsuperscript{1} V. Kkomenko\textsuperscript{2} A. Mokhov\textsuperscript{2} D. Wist\textsuperscript{2} A. Yakovlev\textsuperscript{1}

\textsuperscript{1}School of Electrical, Electronic and Computer Engineering
Newcastle University

\textsuperscript{2}School of Computer Science
Newcastle University

ACSD 2011
Parallel composition

- Also known as Synchronous Product
- Used to combine components models into a system model.
Use case: Circuit Resynthesis

- Circuit components are specified as labelled Petri nets (STG).
- Circuit model is produced with parallel composition of components, hiding the shared signals.
- The model STG is simplified using structural methods.
- Gate-level implementation of the circuit is derived.
Use case: Circuit Resynthesis

- Circuit components are specified as labelled Petri nets (STG).
- Circuit model is produced with parallel composition of components, hiding the shared signals.
- The model STG is simplified using structural methods.
- Gate-level implementation of the circuit is derived.
Working example

Two interconnected components and an environment specification

Circuit

Mix

Toggle

Environment
Standard parallel composition

Motivation
Our approach
Summary

Parallel Composition Example

Standard parallel composition

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Implicit places

- Also known as redundant places.
- Not necessary.
- Can appear in parallel composition even when there were none in the original Petri nets.
Implicit places effect

- Can affect the tools performance.
  - Model checking tools are generally not affected (there do exist some heuristics that are);
  - Structural tools are affected.
- Can be found if duplicate or shortcut.
- Hard to find in general.
Parallel Composition Example

Standard parallel composition

With implicit places removed

Motivation
Our approach
Summary

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Transition contraction

- Structural operation.
- Reduces the number of transitions.
- Has some conditions of applicability.
- Affected by implicit places.
Our goal

- Remove as many implicit places as possible.
- Perform it cheaply.
Computation interference

- CI means that a component produces output without others expecting it.
- We need FCI – Freedom from Computation Interference.
- The composition does not make sense if FCI is violated.
- FCI is assumed anyway and can be guaranteed by model construction.
Motivation

Our approach

Summary

Our Method of Place Removal

FCI
⇒ components do not have to control their inputs
⇒ places preceding inputs can be removed.
Our Method of Place Removal

FCI

⇒ components do not have to control their inputs
⇒ places preceding inputs can be removed.

\[
\text{component 1} \parallel \text{component 2}
\]
Our Method of Place Removal

FCI
⇒ components do not have to control their inputs
⇒ places preceding inputs can be removed.
FCI
⇒ components do not have to control their inputs
⇒ places preceding inputs can be removed.
Place Removal Applied

Motivation

Our approach

Summary

Place Removal Applied

Mix

Toggle

Environment

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Place Removal Applied
Place Removal Applied

Motivation

Our approach

Summary

Mix

Toggle

Environment

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Motivation
Our approach
Summary

Place Removal Applied
Non-injective labelling

Injective labelling – signal label occurs at most once.

Can be enforced with:
- automated structural transformation;
- manual transformations;
- Petri net synthesis.

Places preceding non-unique labels still carry information of which transition to fire.

Such places can be left in the composition.
Motivation

Our approach

Summary

Dummy Transitions

- $a^+ b^+$ does not produce any output in the original component.
- $a^+ b^+ x^+$ is possible if the place is removed.
- We do not remove such places.

Original component

With a place removed

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Experiment Set Up

- Choose a Balsa benchmark and its size
- Take individual component STGs
  - injectively labelled
  - with label-splitting
- Perform parallel composition
  - standard
  - improved
- Perform dummy contraction with DesiJ
  - safeness-preserving
  - all contractions
Balsa Benchmarks

Motivation
Our approach
Summary

Seq, Mix

Par, Arb, Mix

Seq, Par, Sync, Mix

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Experimental results

Dummy transitions remaining after contraction in SeqCallParSync

Safeness-preserving

Alekseyev, Khomenko, Mokhov, Wist, Yakovlev

Improved Parallel Composition
Experimental results
Dummy transitions remaining after contraction in SeqCallParSync

All contractions

<table>
<thead>
<tr>
<th></th>
<th>std</th>
<th>opt</th>
<th>inj</th>
<th>opt+inj</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>100%</td>
<td>22%</td>
<td>63%</td>
<td>41%</td>
</tr>
<tr>
<td>3</td>
<td>100%</td>
<td>19%</td>
<td>53%</td>
<td>43%</td>
</tr>
<tr>
<td>4</td>
<td>100%</td>
<td>19%</td>
<td>51%</td>
<td>44%</td>
</tr>
</tbody>
</table>
The parallel composition improvement was developed and implemented in software.

Significant improvement in some cases.

Improvement comes at negligible computational cost.