# Improved Parallel Composition

A. Alekseyev[1] V. Kkomenko[2] A. Mokhov[2] D. Wist[2]
A. Yakovlev[1]

[1]School of Electrical, Electronic and Computer Engineering
Newcastle University

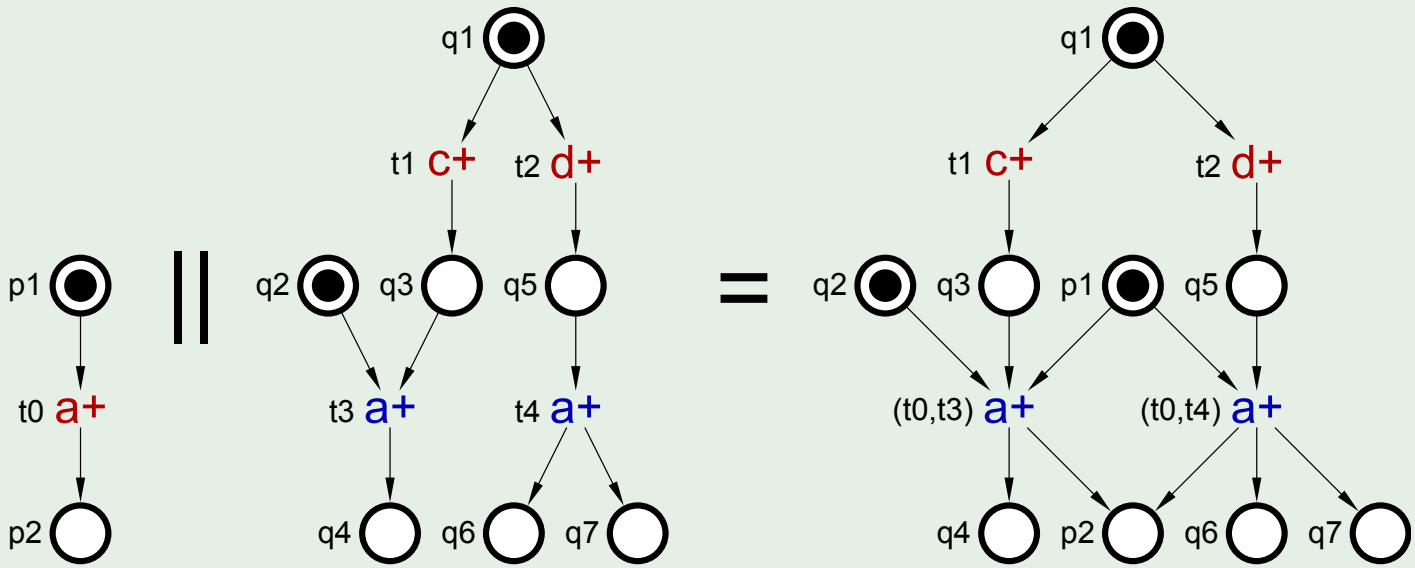
[2]School of Computer Science
Newcastle University

ACSD 2011

# Parallel composition

- Also known as Synchronous Product
- Used to combine components models into a system model.

## Parallel composition



# 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.

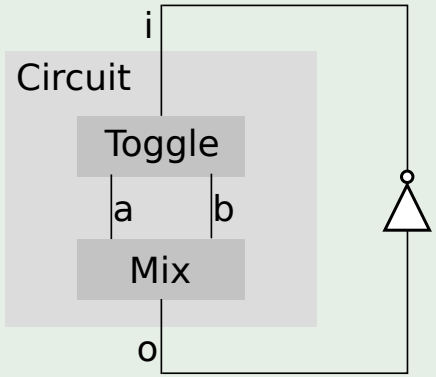
# Use case: Circuit Resynthesis

- 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.

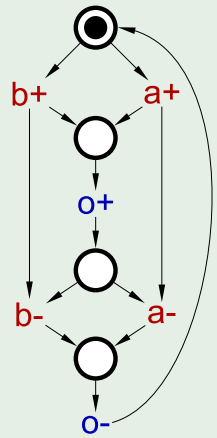
# Working example

## Two interconnected components and an environment specification

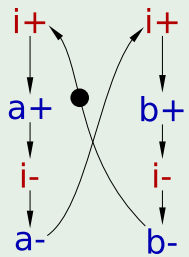
### Circuit



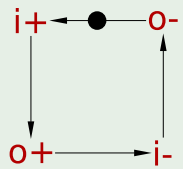

### Mix




### Toggle

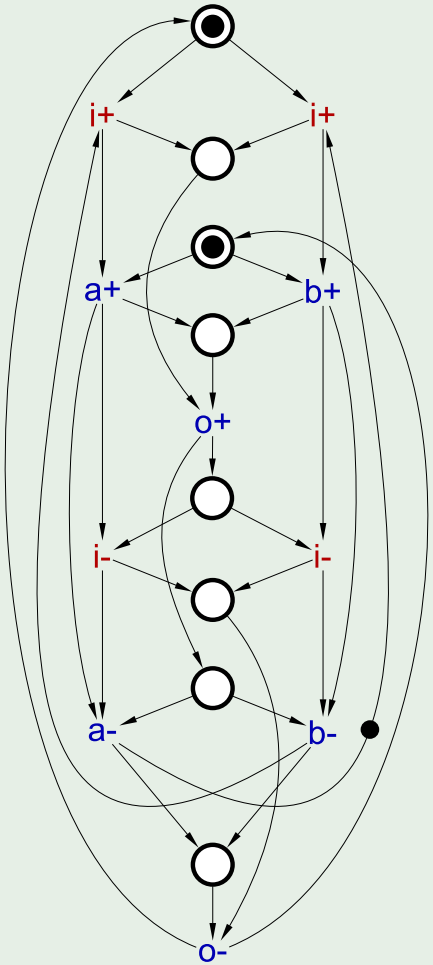



### Environment



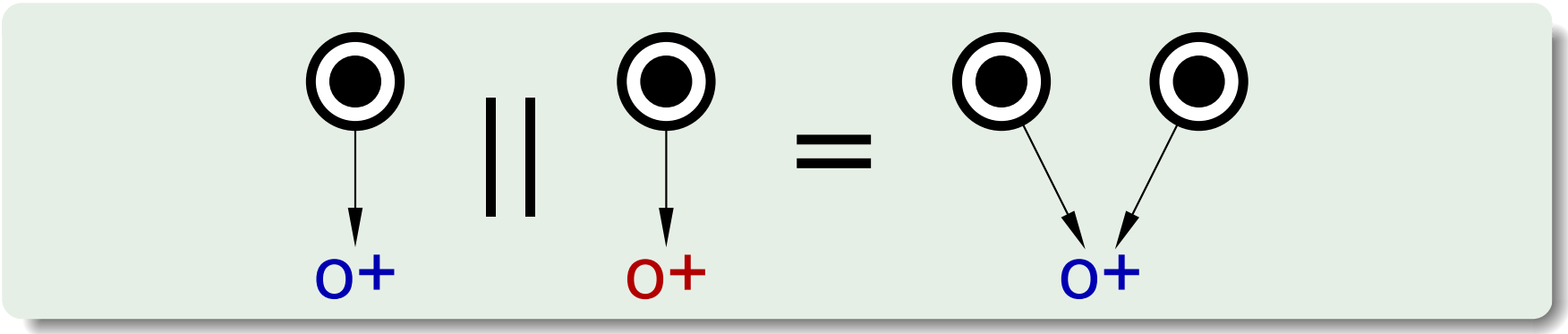
# Parallel Composition Example

## Standard 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.

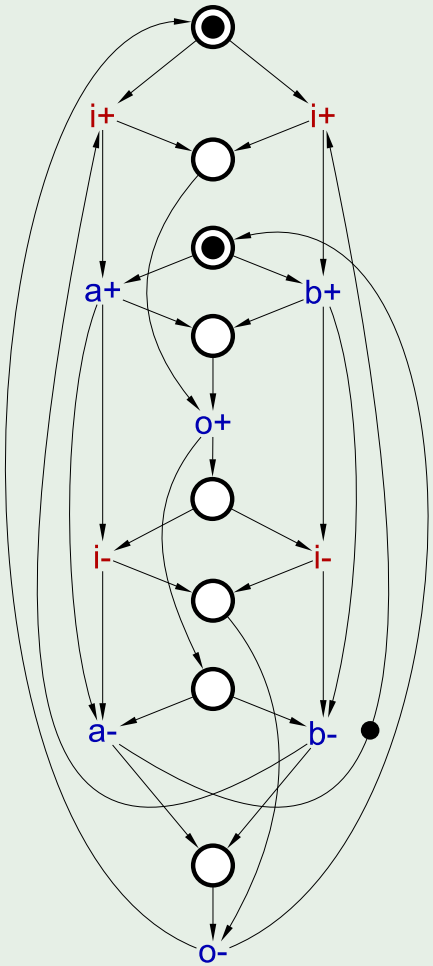


# Implicit places effect

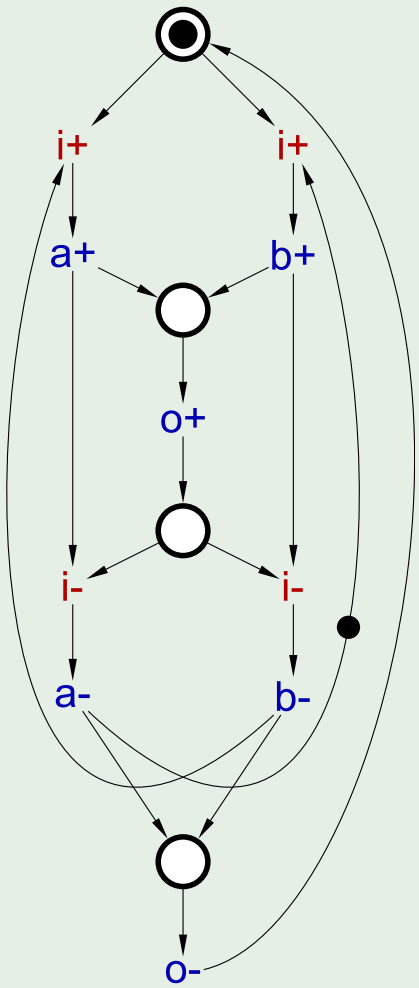
- Can affect the tools performance.
  - Model checking tools are generally not affected (there do exist some heuristics that are);
  - Structural tools are affected.
- Can be found if duplicate or shortcut.
- Hard to find in general.

# Parallel Composition Example

## Standard parallel composition

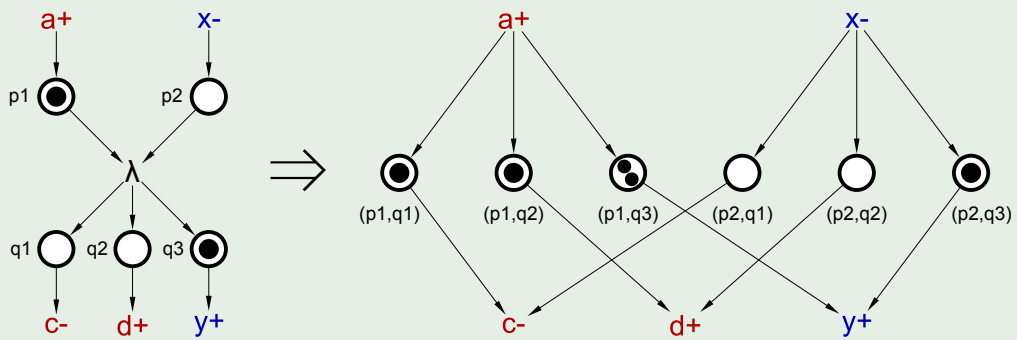



## With implicit places removed



# Transition contraction

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

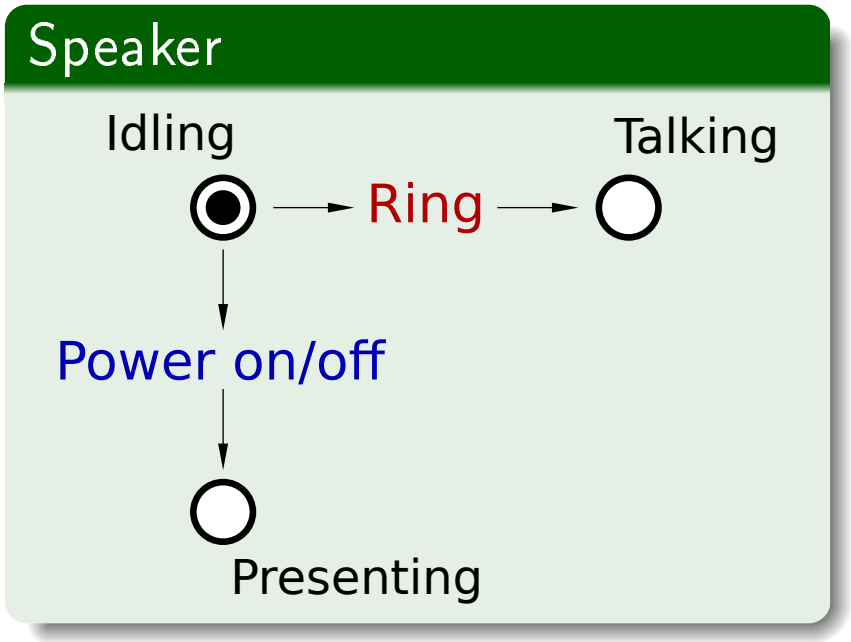


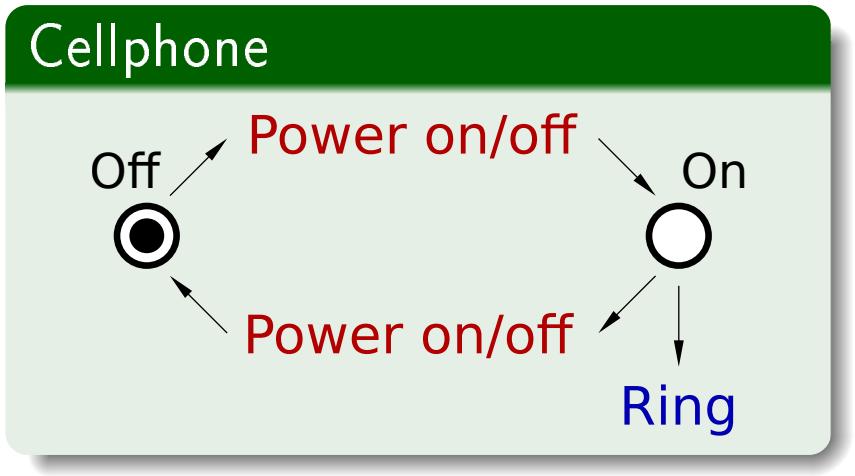
# Our goal

- 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.



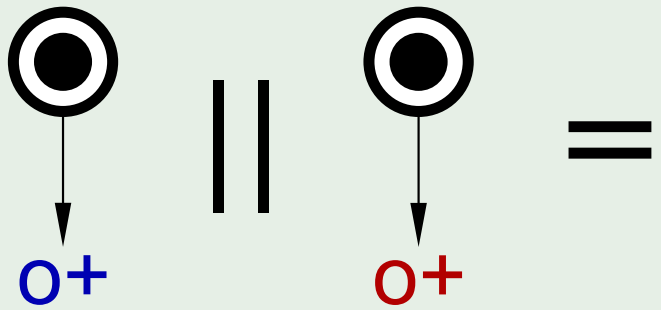



# Our Method of Place Removal

FCI

$\Rightarrow$ components do not have to control their inputs

$\Rightarrow$ places preceding inputs can be removed.

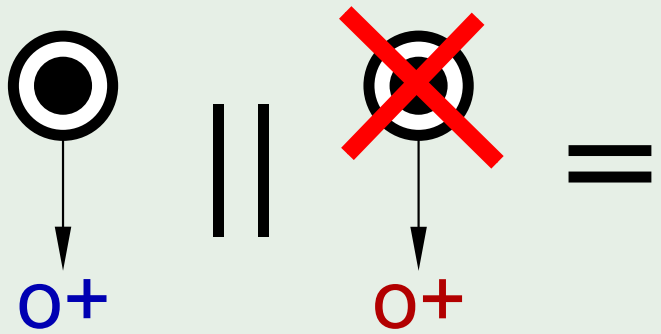


# Our Method of Place Removal

FCI

⇒ components do not have to control their inputs

⇒ places preceding inputs can be removed.

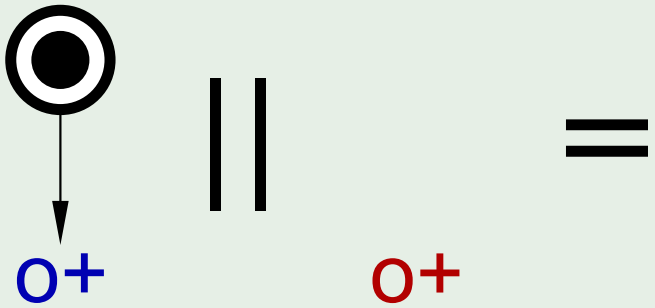


# Our Method of Place Removal

FCI

$\Rightarrow$ components do not have to control their inputs

$\Rightarrow$ places preceding inputs can be removed.

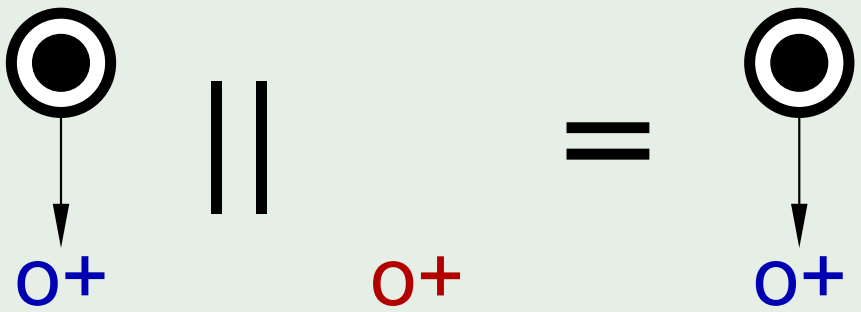


# Our Method of Place Removal

FCI

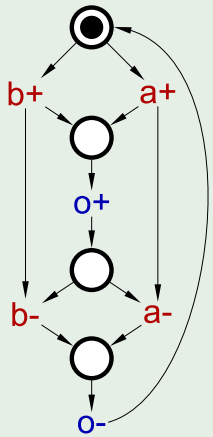
$\Rightarrow$ components do not have to control their inputs

$\Rightarrow$ places preceding inputs can be removed.

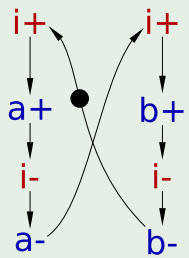


# Place Removal Applied

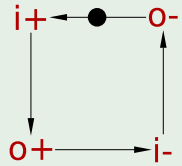
## Mix




## Toggle

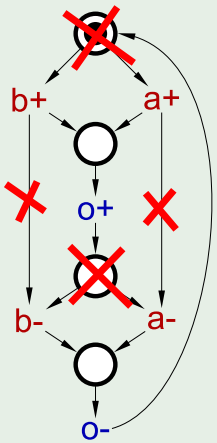



## Environment

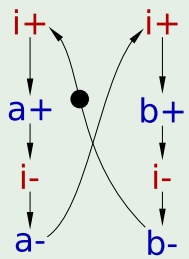


# Place Removal Applied

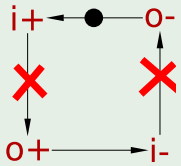
## Mix




## Toggle

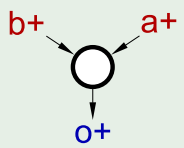


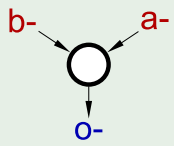

## Environment



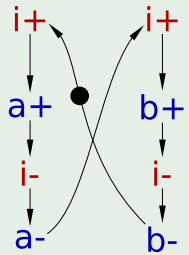
# Place Removal Applied

## Mix



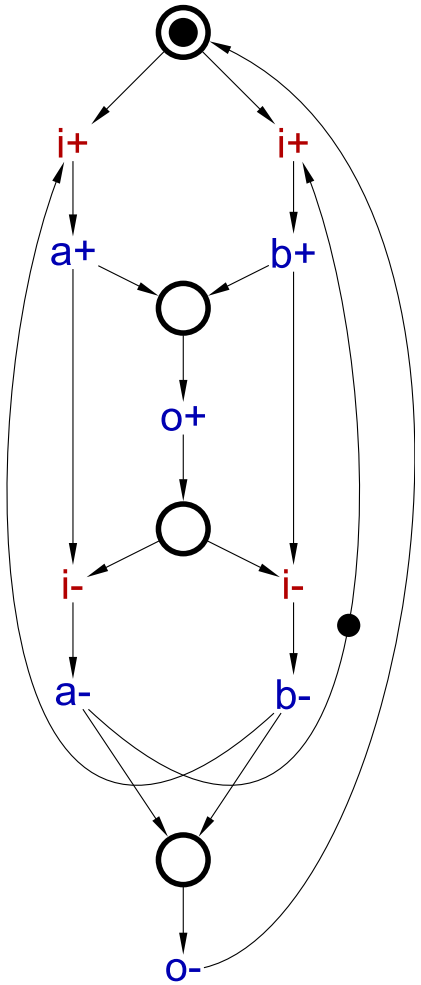




## Toggle



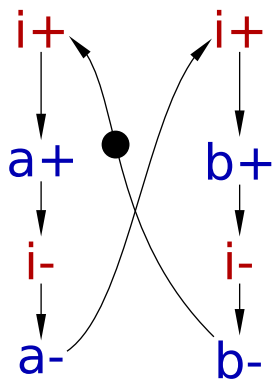

## Environment


# Place Removal Applied



# Non-injective labelling

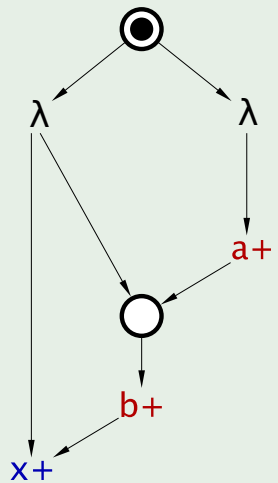
- Injective labelling – signal label occurs at most once.
- Can be enforced with:
  - automated structural transformation;
  - manual transformations;
  - Petri net synthesis.
- Places preceding non-unique labels still carry information of which transition to fire.
- Such places can be left in the composition.



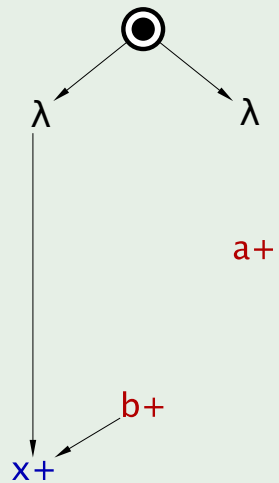
# Dummy Transitions

- $a^+b^+$ 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



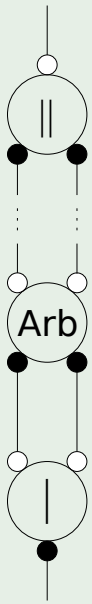
# Experiment Set Up

- 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

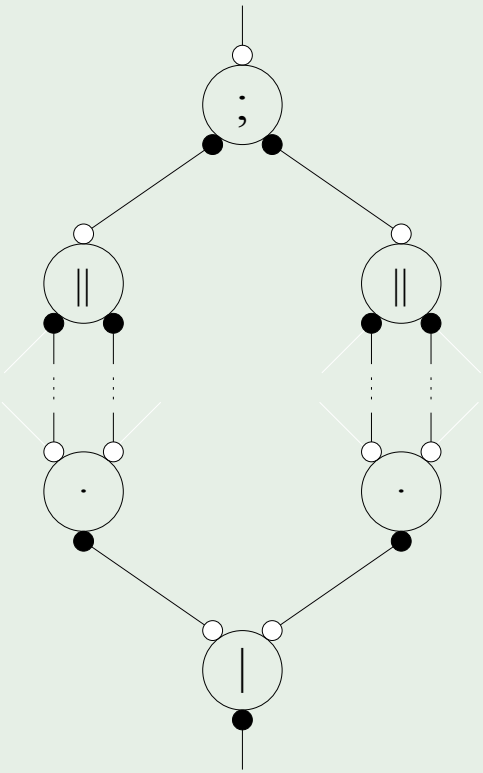
# Balsa Benchmarks

## Seq, Mix


## Par, Arb, Mix

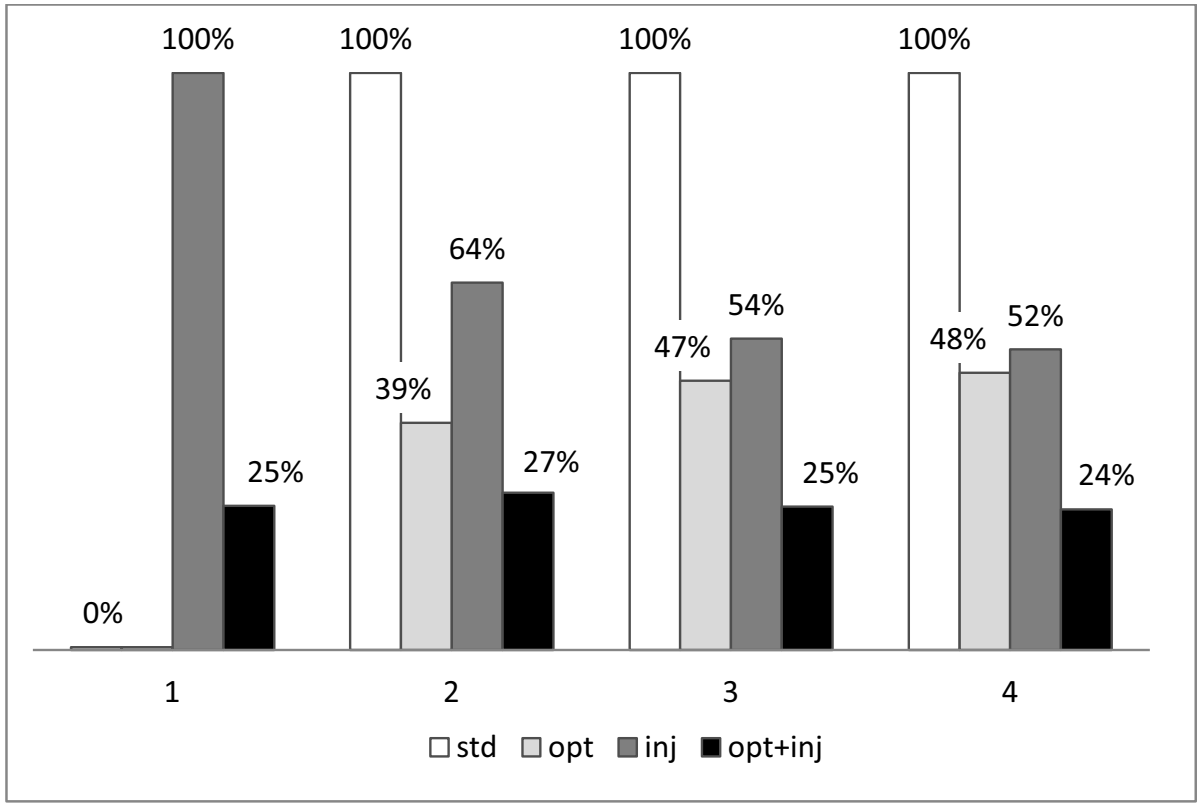



## Seq, Par, Sync, Mix



# Experimental results

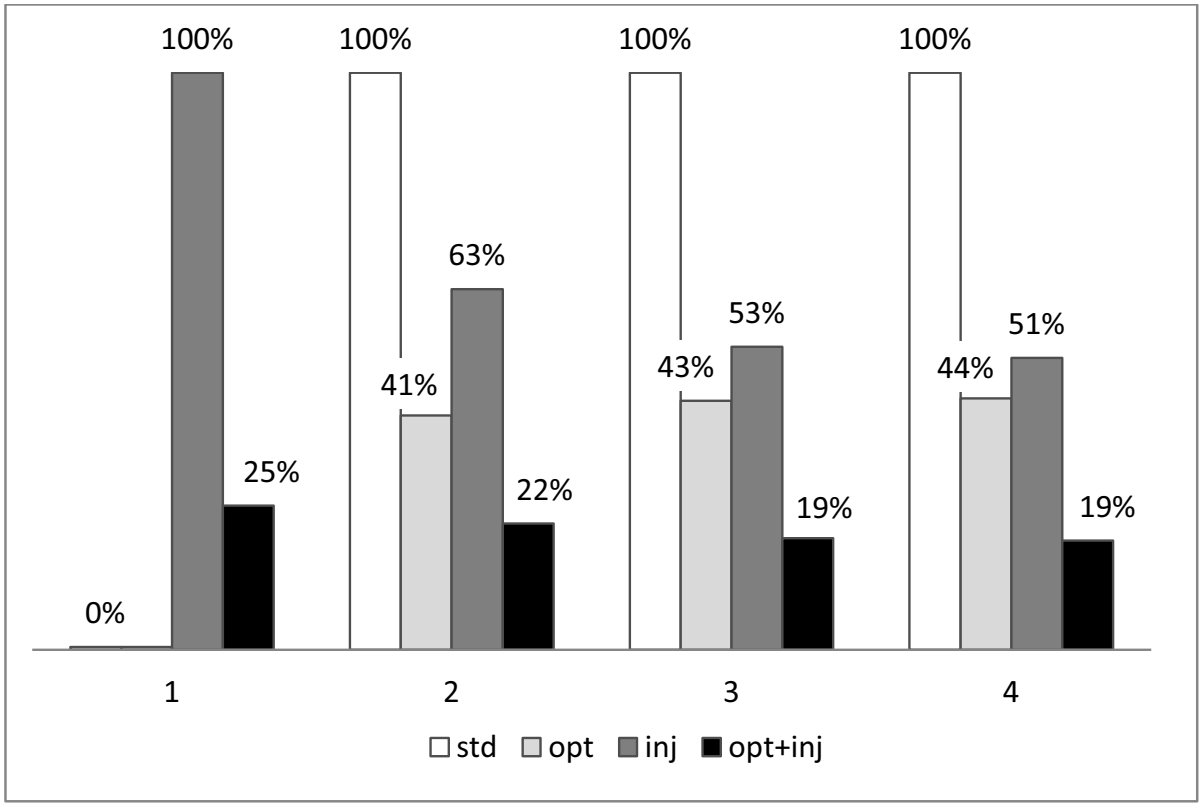
## Dummy transitions remaining after contraction in SeqCallParSync



# Experimental results

## Dummy transitions remaining after contraction in SeqCallParSync

All contractions



# Summary

- The parallel composition improvement was developed and implemented in software.
- Significant improvement in some cases.
- Improvement comes at negligible computational cost.