On the Compositionality of Round Abstraction

Mohamed N. Menaa

University of Birmingham

LICS 2010 Short Presentation
13 July 2010

Introduction

Problem

Building synchronous representations of asynchronous behaviours compositionally.

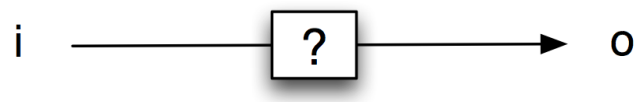
Introduction

Problem

Building synchronous representations of asynchronous behaviours compositionally.

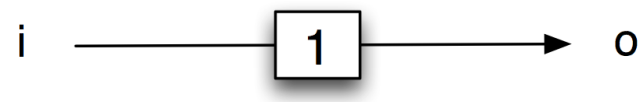
Naive Approach

Asynchronous behaviour



$(i.o)^*$

Synchronous representation



$(i.o)^*$

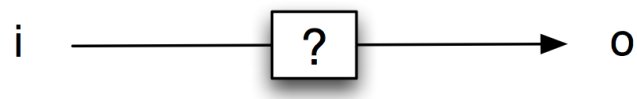
Introduction

Problem

Building synchronous representations of asynchronous behaviours compositionally.

Naive Approach

Asynchronous behaviour



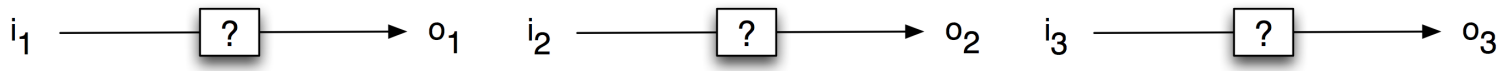
$(i.o)^*$

Synchronous representation

$(i.o)^*$
$(i.\tau.o)^*$

In Composition

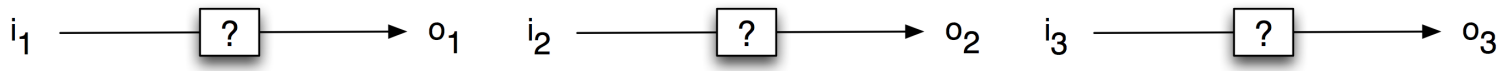
- Composition of asynchronous identities



$$(i_1 . o_3)^*$$

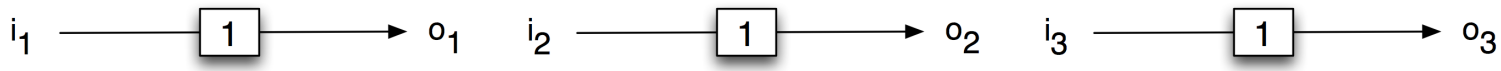
In Composition

- Composition of asynchronous identities



$$(i_1.o_3)^*$$

- Composition of synchronous identities

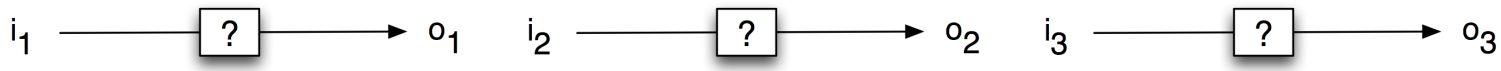


$$(i_1.o_3)^* = (i_1.\tau.\tau.\tau.o_3)^*$$

  - Correct but very inefficient
  - High latency
  - High resource usage

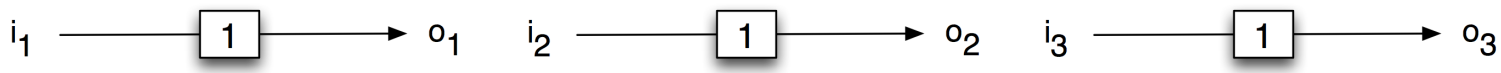
In Composition

- Composition of asynchronous identities



$$(i_1.o_3)^*$$

- Composition of synchronous identities



$$(i_1.o_3)^* = (i_1.\tau.\tau.\tau.o_3)^*$$

  - Correct but very inefficient
  - High latency
  - High resource usage
- Desideratum: abstraction from $(i_1.\tau.\tau.\tau.o_3)^*$ to $\langle i_1, o_3 \rangle^*$
- Round abstraction after similar technique by Alur and Henzinger (*Reactive Modules*, LICS'96)

Correctness Criteria

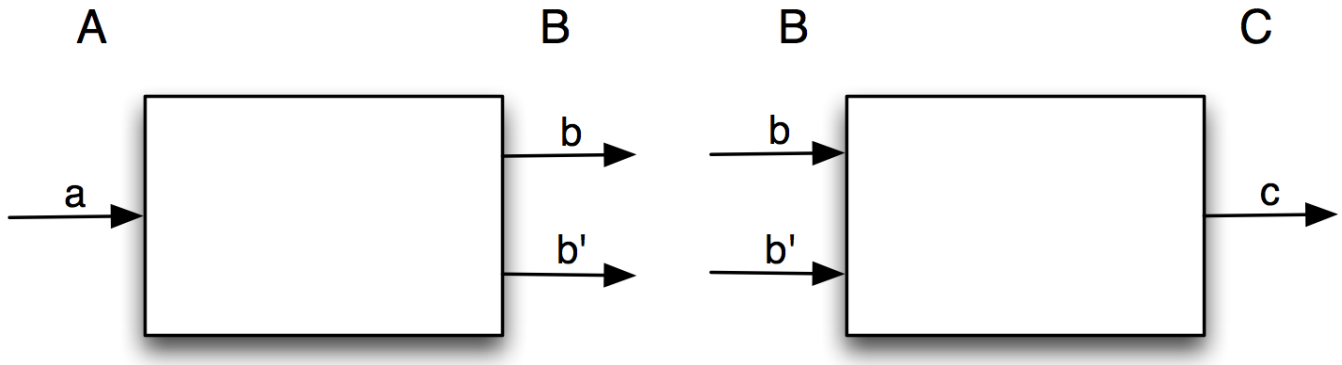
Soundness

All traces in the composition of round abstractions stem from the composition of their asynchronous counterparts (no traces are introduced).

Adequacy

All traces in the composition of asynchronous behaviours have an equivalent in the composition of their round abstractions (no traces are lost)

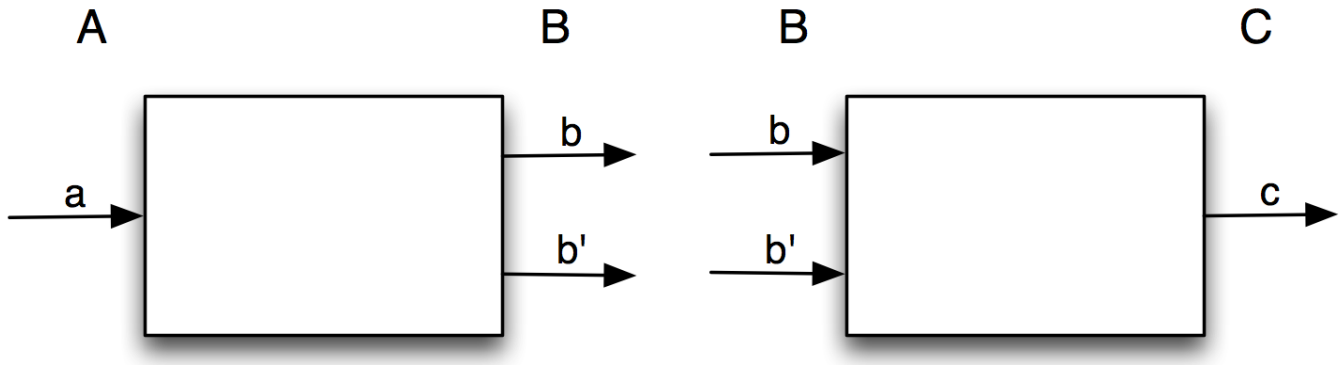
Round Abstraction on Traces



Asynch	$\{a.b.b'\}$	;	$\{b'.b.c\}$	=	$\emptyset$
	$\wr$		$\wr$		
Synch	$\{a.\langle b, b'\rangle\}$	;	$\{\langle b', b\rangle.c\}$	=	$\{a.c\}$

Not sound! (Deadlock is resolved)

Round Abstraction on Traces



Asynch	$\{a.b.b'\}$	;	$\{b.b'.c\}$	=	$\{a.c\}$
	$\wr$		$\wr$		
Synch	$\{\langle a, b\rangle.b'\}$	;	$\{\langle b, b'\rangle.c\}$	=	$\emptyset$

Not adequate! (Deadlock is introduced)

Conditions for Soundness and Adequacy

Soundness

- **Compatibility:** two behaviours are compatible if
  - Pre: they do not deadlock.
  - Post: they deadlock and their resp. round abstractions do not resolve it.

Conditions for Soundness and Adequacy

Soundness

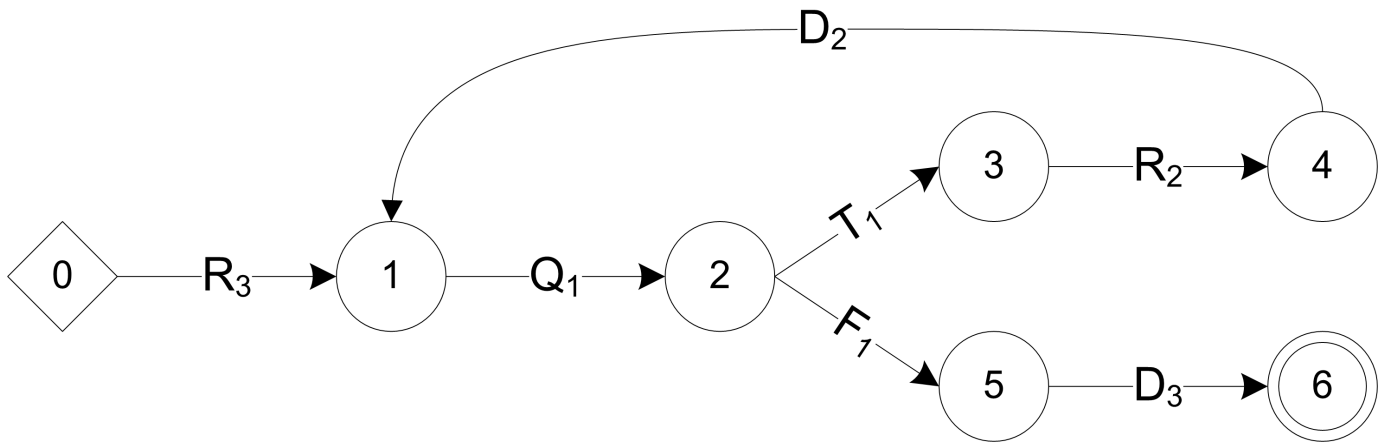
- **Compatibility:** two behaviours are compatible if
  - Pre: they do not deadlock.
  - Post: they deadlock and their resp. round abstractions do not resolve it.

Adequacy

- **Safety:** the composition of two asynchronous behaviours is safe if any output produced by one of them, at any time, can be handled by the other.
- **Receptivity:** a round abstraction is receptive when it satisfies
  - Input receptivity: successive inputs can be received in succession as well as simultaneously,
  - Instant feedback receptivity: an input following an output may also be received simultaneously.

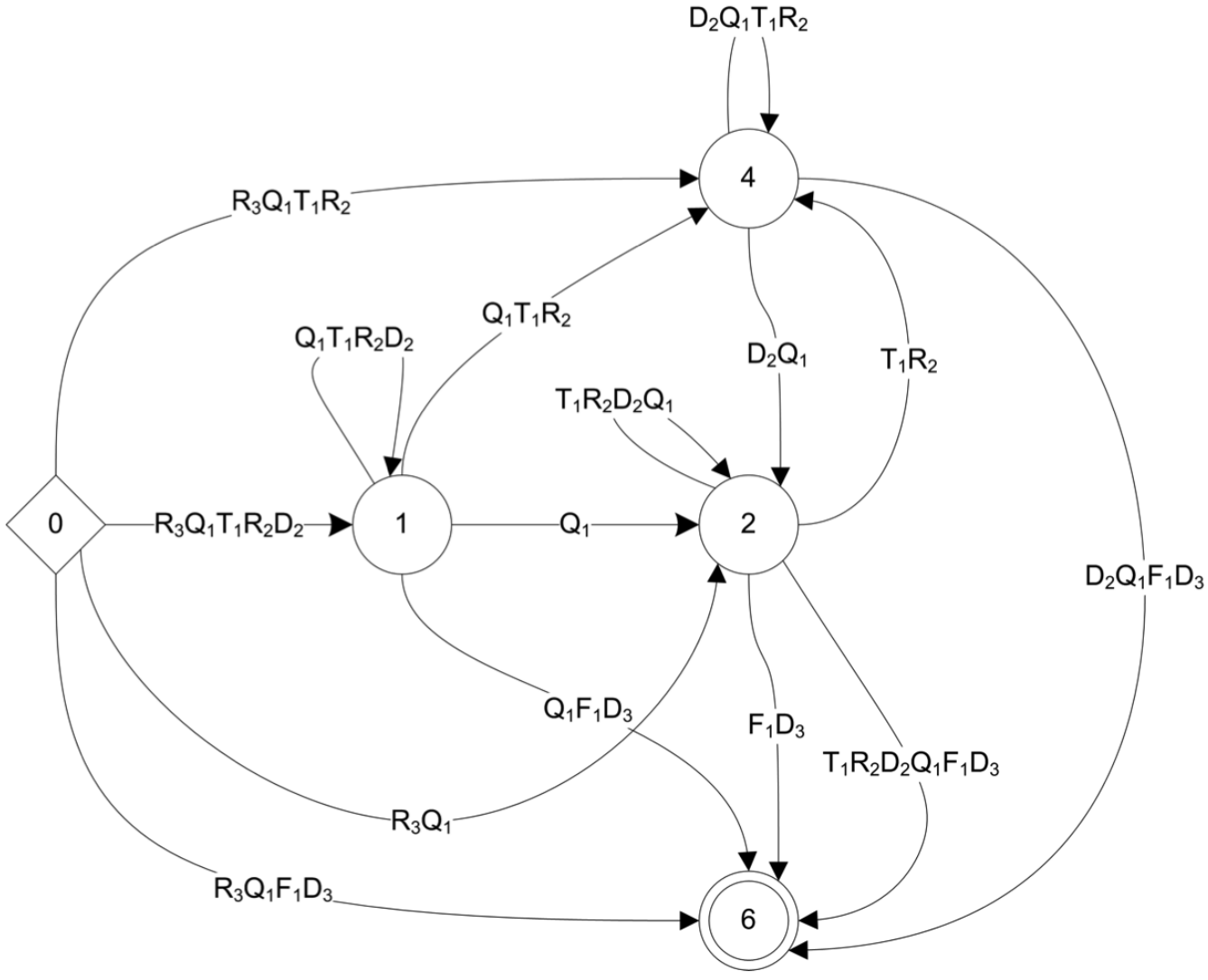
Application to Geometry of Synthesis (Ghica, POPL'07)

GoS: hardware compilation technique from programming languages to hardware via game semantics.



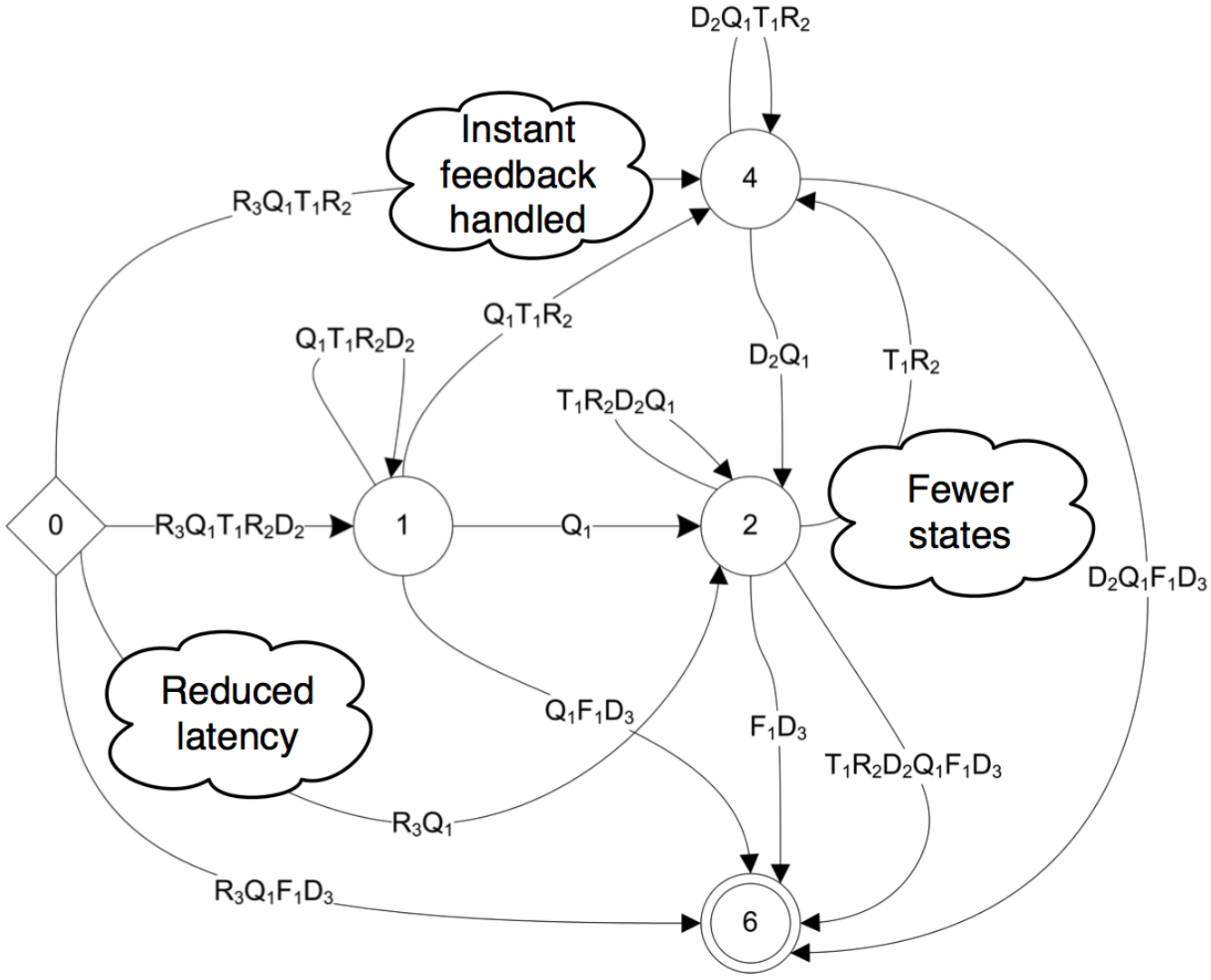
'Original' asynchronous representation of the iteration circuit

Application to Geometry of Synthesis (Ghica, POPL'07)



A synchronous representation of the iteration circuit

Application to Geometry of Synthesis (Ghica, POPL'07)



A synchronous representation of the iteration circuit

Conference Paper

"On the compositionality of round abstraction"
In CONCUR'10 (with Dan Ghica)