Structuring Interaction in BIP

Workshop on

Foundations of Component-based Design

Embedded Systems Week

Salzburg, September 28, 2007

Simon Bliudze and Joseph Sifakis

VERIMAG

Motivation for BIP

Provide a **unified composition framework** for describing and analyzing interaction between components in terms of **tangible, well-founded and organized concepts** instead of using dispersed mechanisms including semaphores, monitors, message passing, remote call etc.

Requirements: The framework

• relies on a minimal set of constructs and principles

• treats interaction and system architecture as first class entities that can be composed and analyzed - independently of the behavior of individual components

• is expressive enough to directly encompass heterogeneity of synchronization (rendezvous and broadcast) and execution mechanisms (synchronous and asynchronous) – not just the usual product of automata

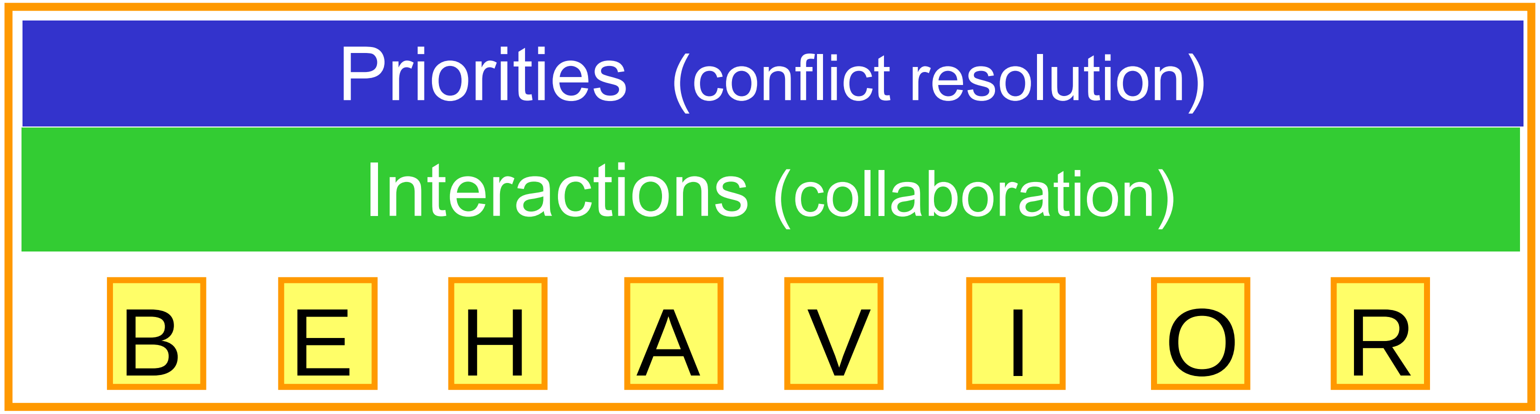
• provides automated support for component integration and generation of glue code meeting given requirements

Overview

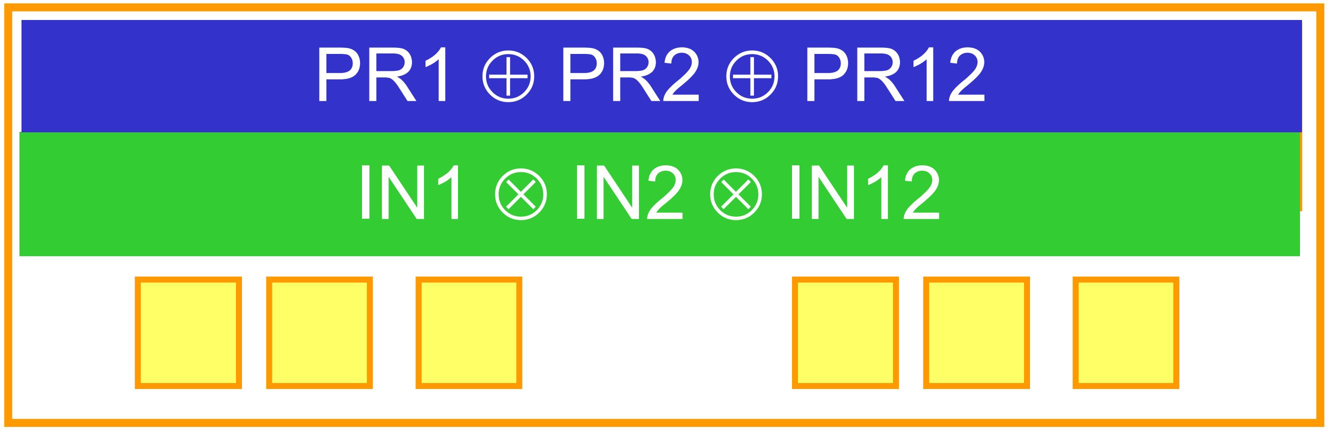
- BIP: Basic Concepts
- The Algebra of Connectors
- One-shot vs. Multi-shot semantics
- Discussion

BIP: Basic Concepts

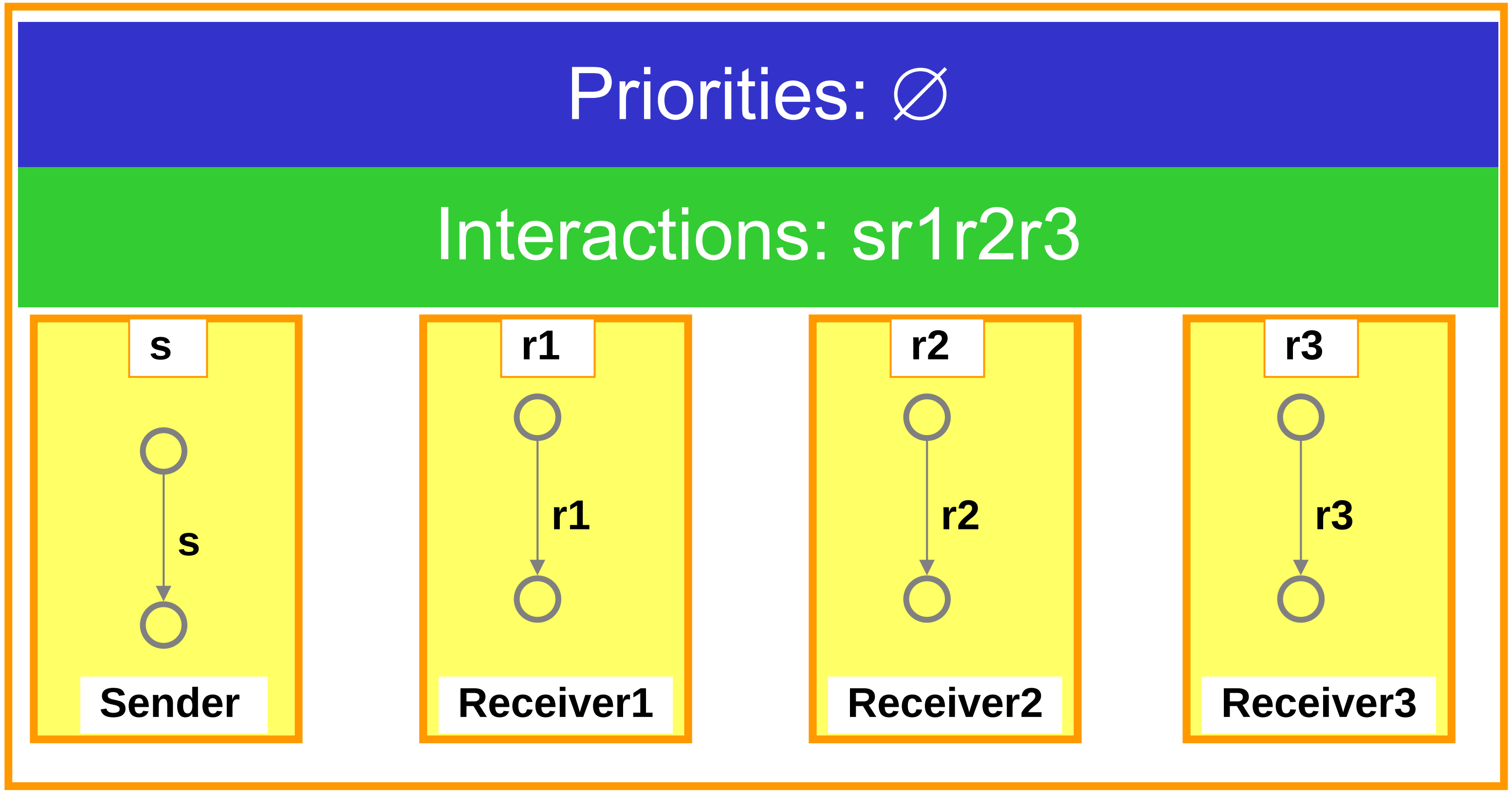
Layered component model



Composition (incremental description)

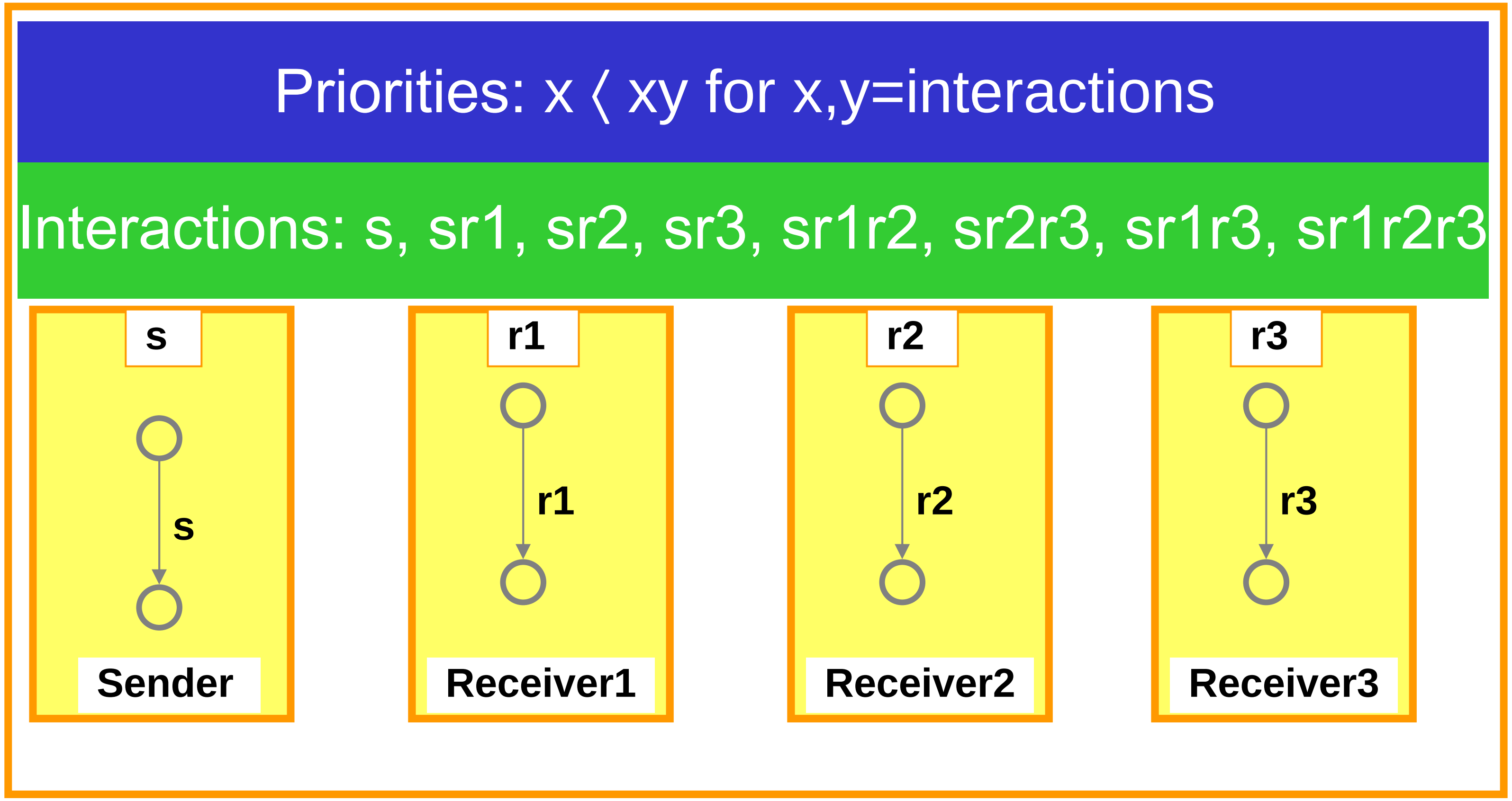


BIP: Basic Concepts



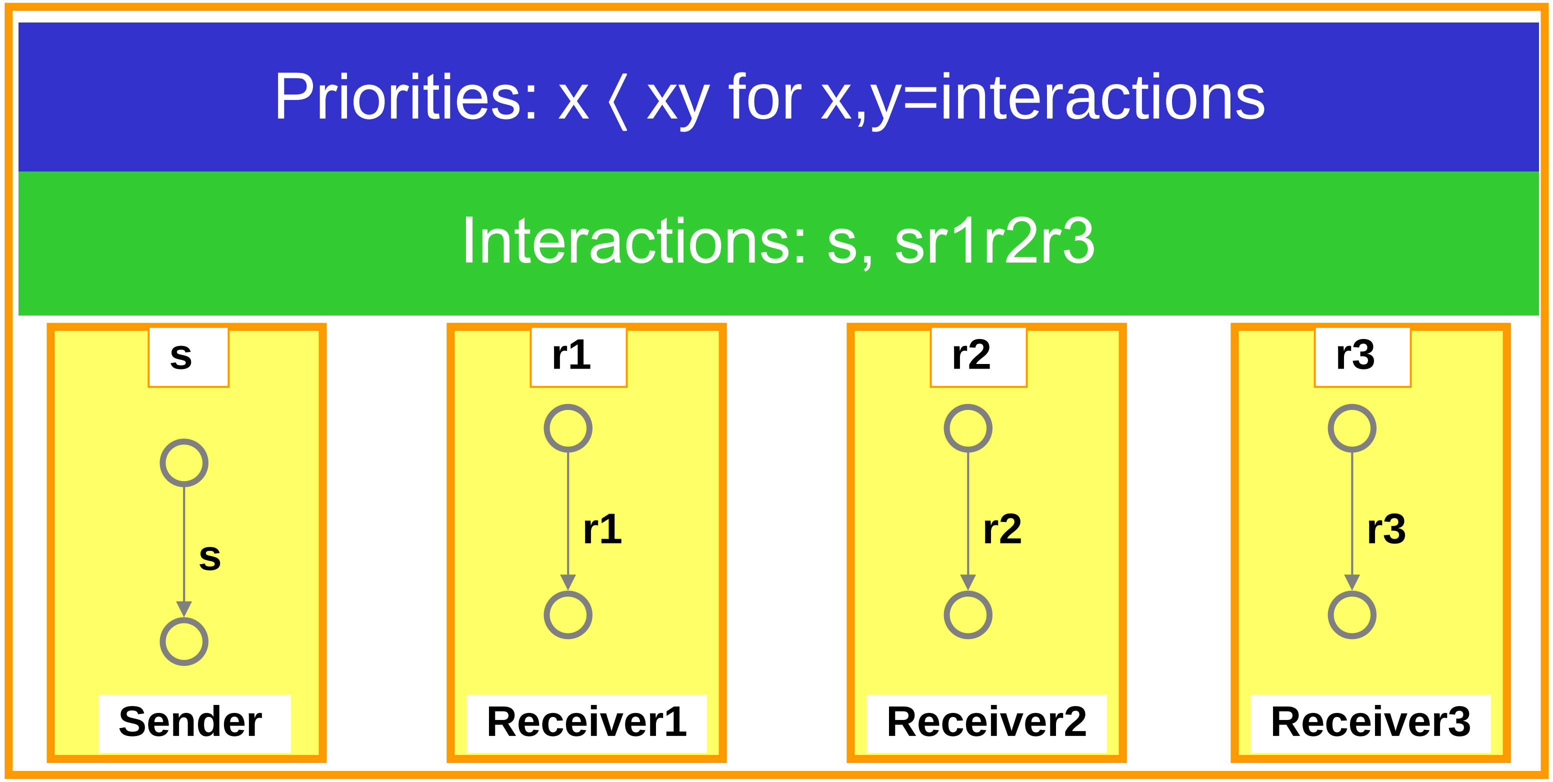
Rendezvous

BIP: Basic Concepts



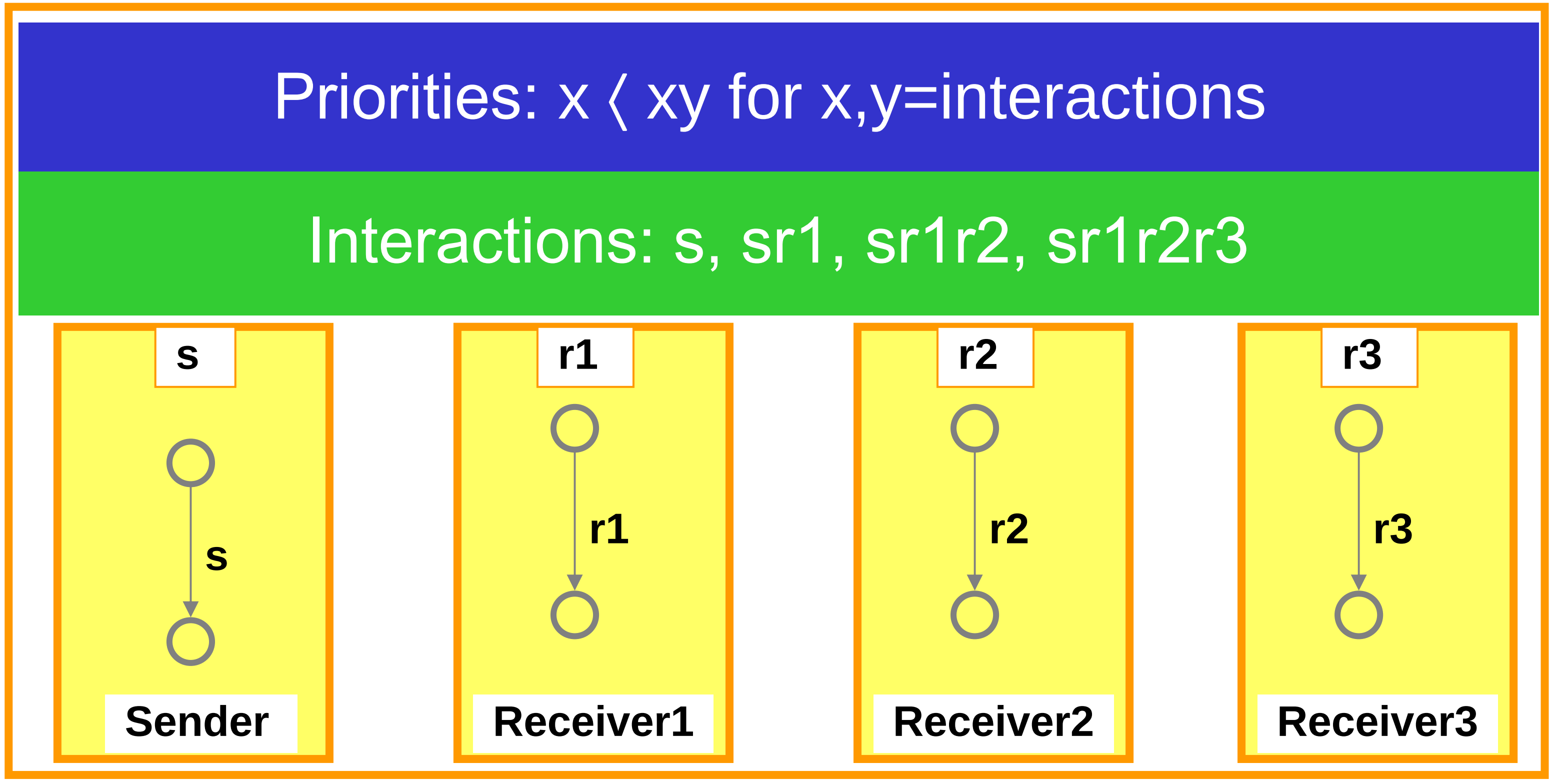
Broadcast

BIP: Basic Concepts



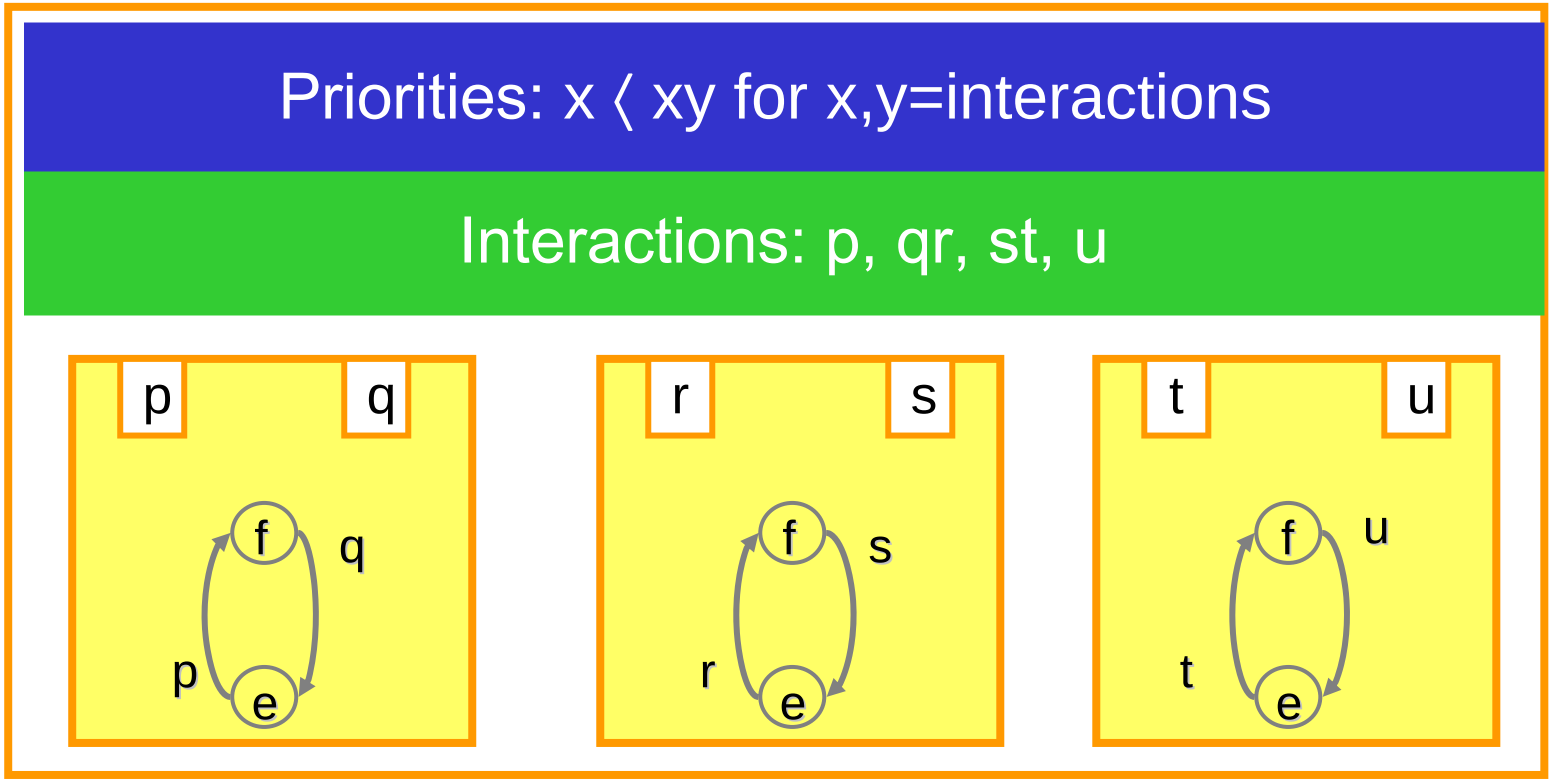
Atomic Broadcast

BIP: Basic Concepts



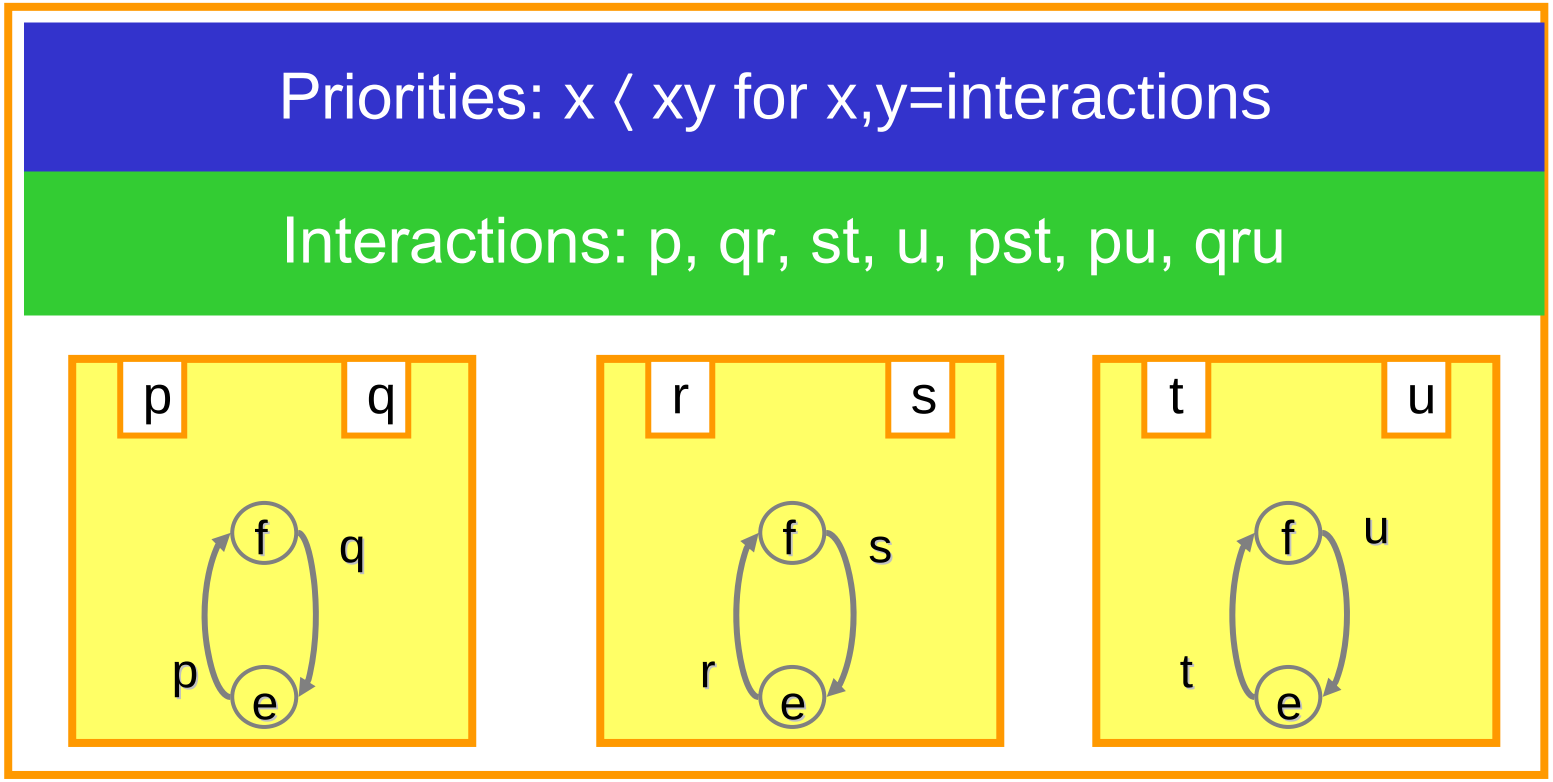
Causal Chain

BIP: Basic Concepts



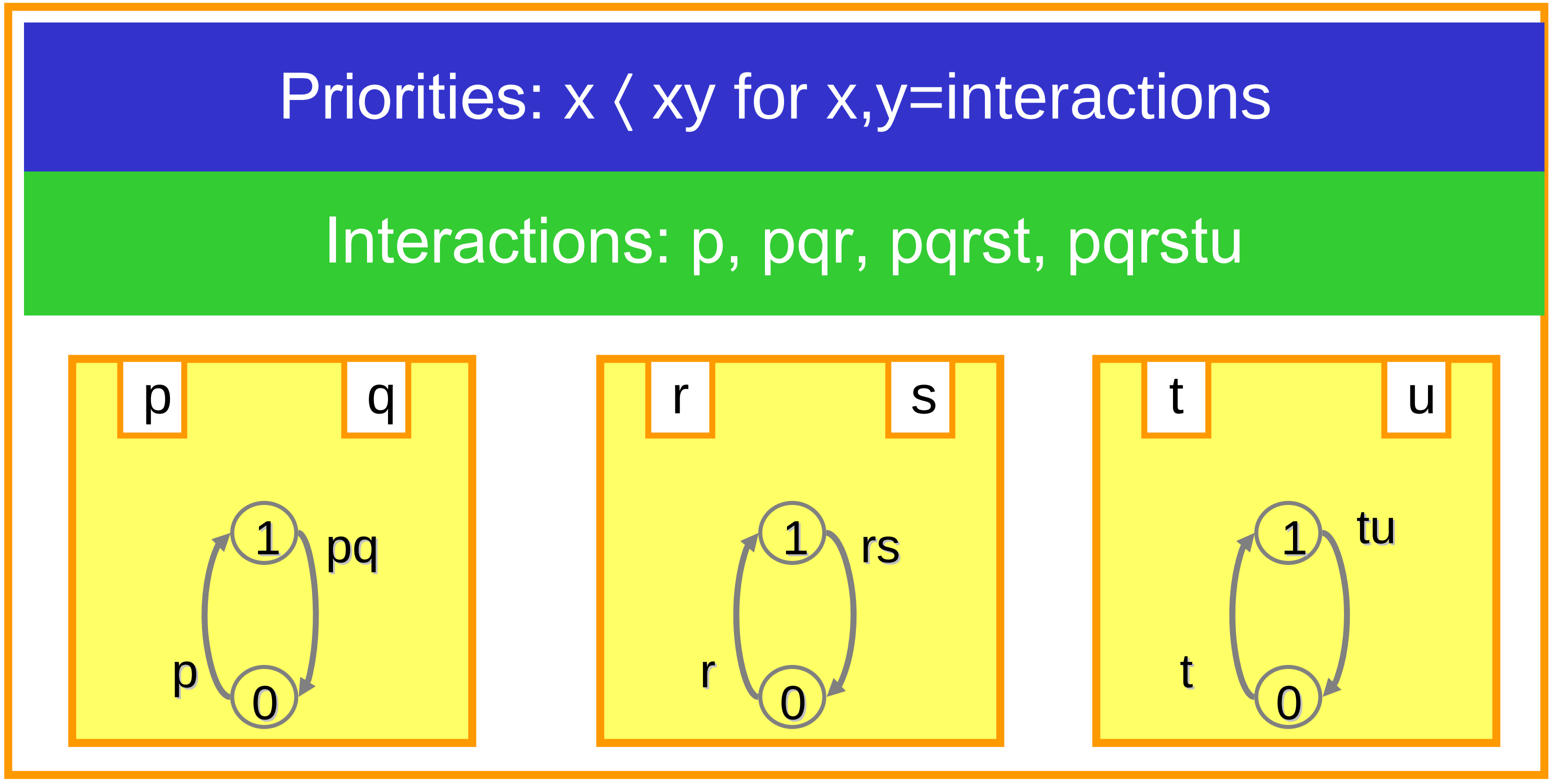
Buffer

BIP: Basic Concepts



Buffer with max progress

BIP: Basic Concepts



Mod8 Counter

BIP: Basic Concepts - Semantics

- a set of atomic components $\{B_i\}_{i=1..n}$ where $B_i = (Q_i,\ 2^{P_i},\ \rightarrow_i)$
- a set of interactions $\gamma \in 2^P$ with $P = \cup_{i=1..n} P_i$ and $P_i \cap P_j = \emptyset$ $P = \cup_{i=1..n} P_i$
- a strict partial order $\pi \subseteq 2^P \times 2^P$

$\Big\}\ \pi\ \gamma\ (B1,.,\ Bn)$

Interactions

$$\frac{a \in \gamma \ \wedge\ \forall i \in [1,n]\ q_i \xrightarrow{a \cap P_i}_i q'_i}{(q_1,.,q_n) \xrightarrow{a}_{\gamma} (q'_1,.,q'_n) \ \text{ where } q'_l = q_l \text{ if } a \cap P_l = \emptyset}$$

Priorities

$$\frac{q \xrightarrow{a}_{\gamma} q' \wedge \neg (\exists\ q \xrightarrow{b}_{\gamma} \wedge\ a\ \pi\ b)}{q \xrightarrow{a}_{\pi} q'}$$

Other parallel composition operators (CCS, SCCS, CSP) can be expressed in BIP

Overview

- BIP: Basic Concepts
- The Algebra of Connectors
- One-shot vs. Multi-shot semantics
- Discussion

The Algebra of Interactions *AI(P)*

Broadcast	$s+sr_1+sr_2+sr_1r_2$	$= s(1+r_1)(1+r_2)$
Causality Chain	$s+sr_1+sr_1r_2$	$= s(1+r_1(1+r_2))$

Syntax: $x ::= 0 \mid 1 \mid p \in P \mid x.x \mid x + x$

where P is a set of ports, such that $0,1 \notin P$

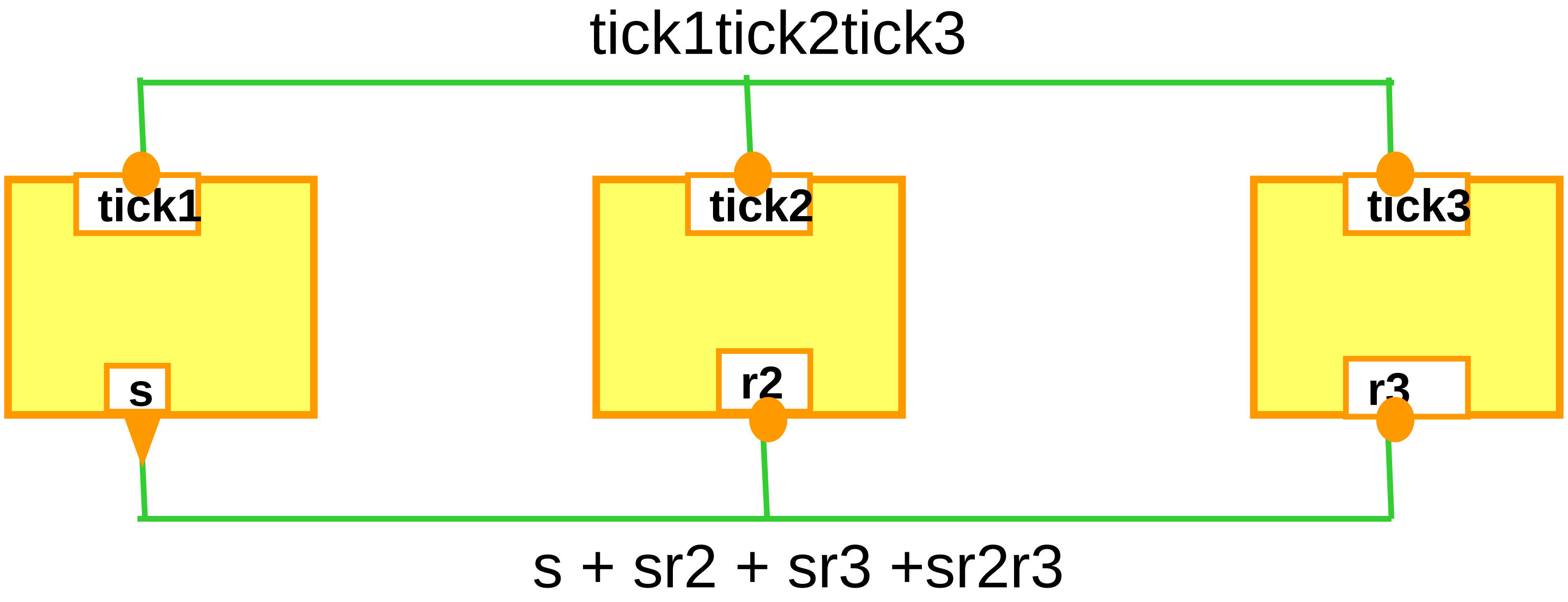
+	*union*	idempotent, associative, commutative, identity 0
.	*synchronization*	idempotent, associative, commutative, identity 1, absorbing 0, distributive wrt +

Semantics: defined by the function $\|\ \|: AI(P) \rightarrow 2^{2^P}$

$$\|0\| = \emptyset$$
$$\|1\| = \{\emptyset\}$$
$$\|p\| = \{\{p\}\}$$
$$\|x_1 + x_2\| = \|x_1\| \cup \|x_2\|$$
$$\|x_1.x_2\| = \{a_1 \cup a_2 \mid a_1 \in \|x_1\| \ \ a_2 \in \|x_2\|\}$$

Simple Connectors

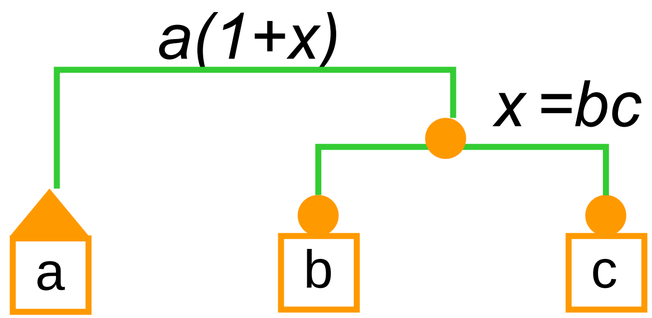
- A ***connector*** is a set of ports which can be involved in an interaction
- Port attributes **(*trigger* ▼, *synchron* ●)** are used to model rendezvous and broadcast.
- An ***interaction*** of a connector is a set of ports such that: either it contains some trigger or it is maximal.



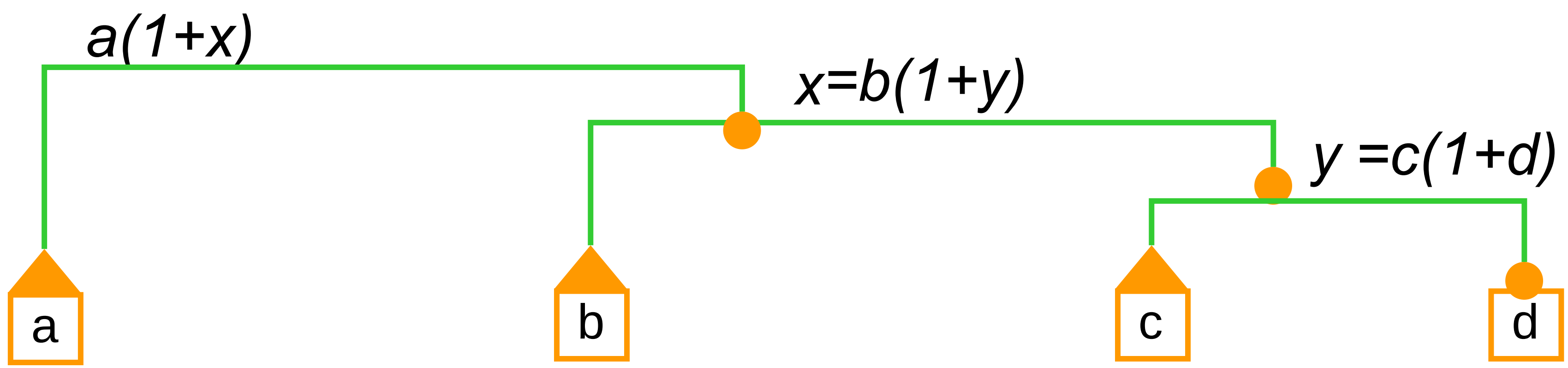
Hierarchical Connectors

Atomic Broadcast:

a+abc



Causality chain: a+ab+abc+abcd



The Algebra of Connectors *AC(P)*

Syntax:

s ::= [0] | [1] | [p] |[x] (synchrons)

t ::= [0]' | [1]' | [p]' | [x]' (triggers)

x ::= s | t | x.x | x + x

where P is a set of ports, such that $0,1 \notin P$

+	*union*	idempotent, associative, commutative, identity [0]
.	*fusion*	idempotent, associative, commutative, identity [1], distributive wrt + ([0] is not absorbing)
[ ], [ ]'	*typing*	unary operators

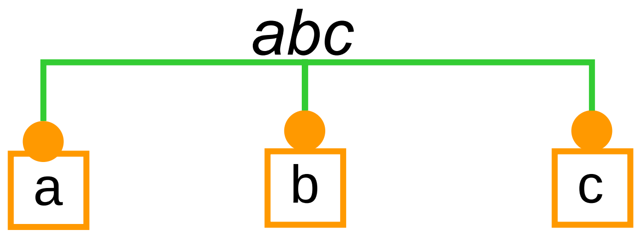
Semantics:

The semantics of AC(P) is given by a function | |: AC(P) $\rightarrow$AI(P)

The Algebra of Connectors *AC(P)* : Examples

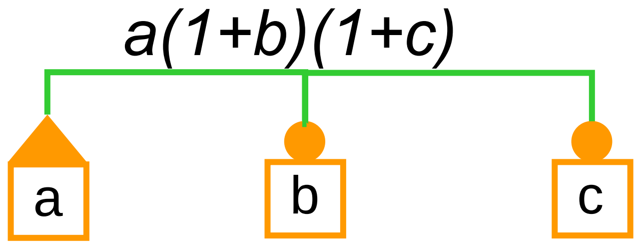
Rendezvous

abc



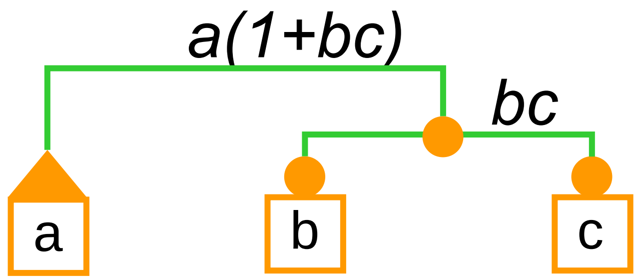
Broadcast

a'bc



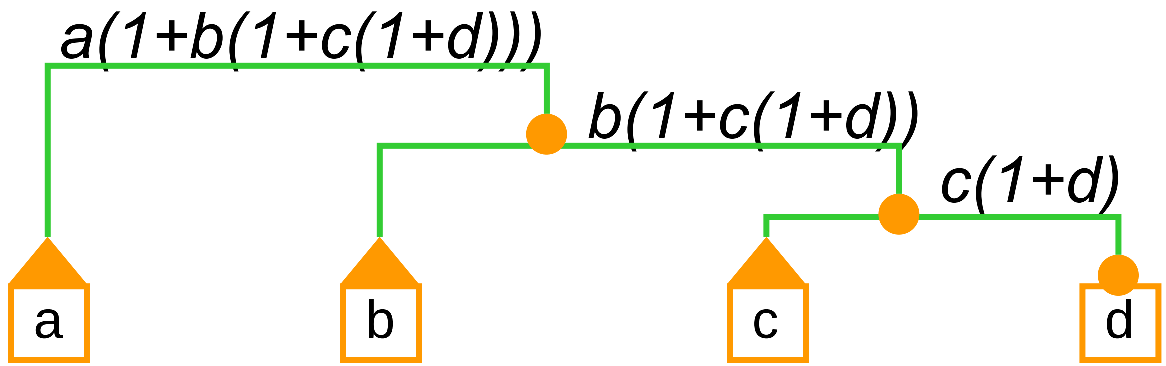
Atomic Broadcast

a'[bc]



Causality chain

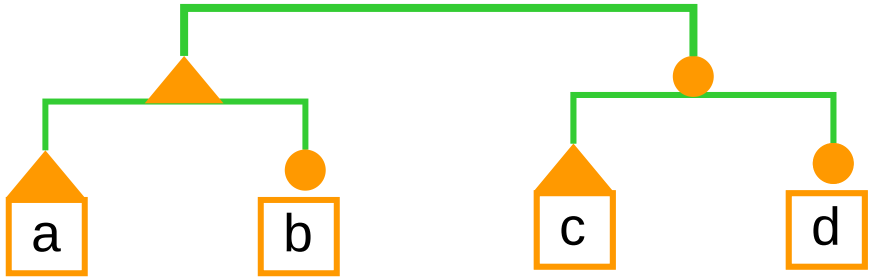
a'[b'[c'd]]



The Algebra of Connectors: Fusion vs. Typing

For two connectors x=a'.b and y=c'.d

xy=a'bc'd= a'bcd + abc'd =(a+b+ab)(1+c)(1+d)



[x]'[y]=[a'b]'[c'd]= a(1+b)(1+c(1+d))

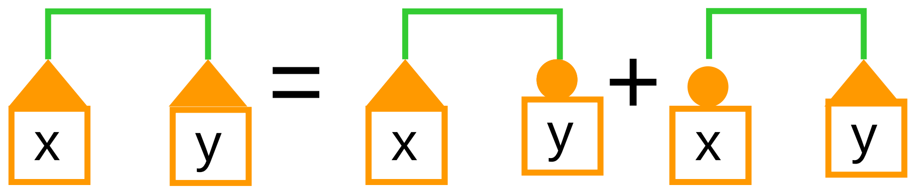
The Algebra of Connectors: Axioms for typing

$[0]' = [0]$

$[[x]^{\alpha}]^{\beta} = [x]^{\beta}$

$[x+y]^{\alpha} = [x]^{\alpha} + [y]^{\alpha}$

$[x]'[y]' = [x]'[y] + [x][y]'$



Fusion for typed connectors is not associative, e.g.

$x[yz] \neq [xy]z$

The Algebra of Connectors: Equivalence vs.Congruence

$x \sim y$ if $|x| = |y|$ i.e. they represent the same set of interactions

- The axiomatization of AC(P) is semantically sound, i.e.

  $x = y \Rightarrow x \sim y$

- $\sim$ is not a congruence (not preserved by fusion)

  $a'b \sim a + ab$ but $a'bc \sim a + ab + ac + abc \nsim ac + abc$

$\approx$ is the largest congruence contained in $\sim$

- $x \sim y \Rightarrow [x]^{\alpha} \approx [y]^{\alpha}$
- Results for inferring congruence from equivalence
- Causal semantics not reducing triggers to rendezvous - the equivalence is a congruence

The Algebra of Connectors: Boolean representation

$\beta: AC(P) \rightarrow B(P)$ where $B(P)$ the boolean calculus on P

For $P=\{p,q,r,s,t\}$

$\beta(pq) = p \wedge q \wedge \neg r \wedge \neg s \wedge \neg t$

$\beta(p'qr) = p \wedge \neg s \wedge \neg t$

$\beta(p+q) = (p \wedge \neg q \vee \neg p \wedge q) \wedge \neg r \wedge \neg s \wedge \neg t$

$\beta(0) = false$

$\beta(1) = \neg p \wedge \neg q \wedge \neg r \wedge \neg s \wedge \neg t$

$\beta(1+p'q'r's't') = true$

Boolean representation depends on the set of ports P, in particular the expression of *fusion* and *typing* in terms of boolean operations .

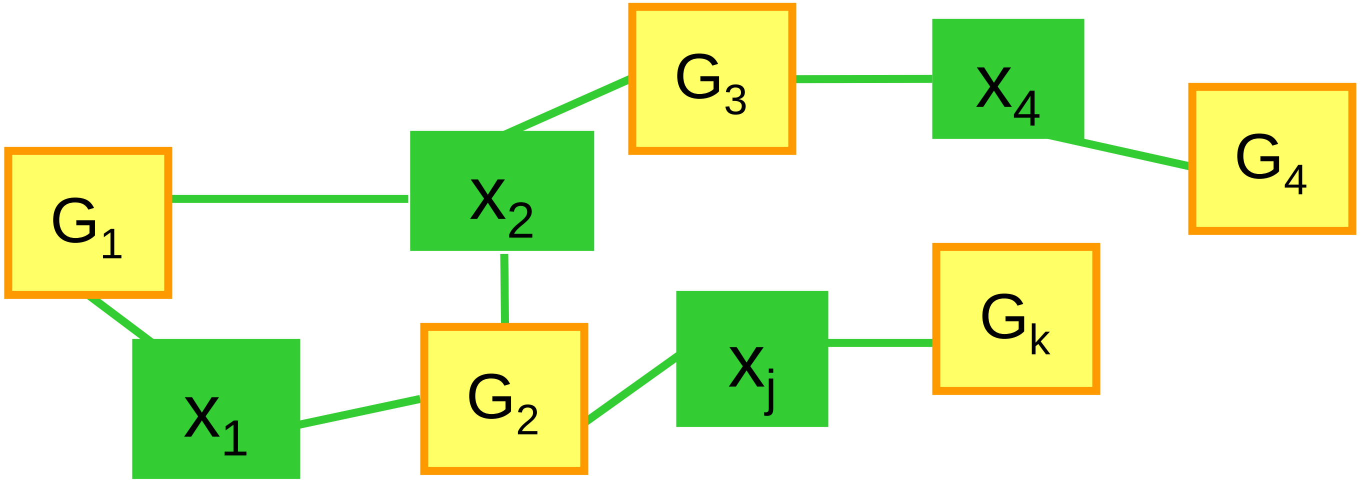
Overview

- BIP: Basic Concepts
- The Algebra of Connectors
- One-shot vs. Multi-shot semantics
- Discussion

One-shot vs. Multi-shot Semantics

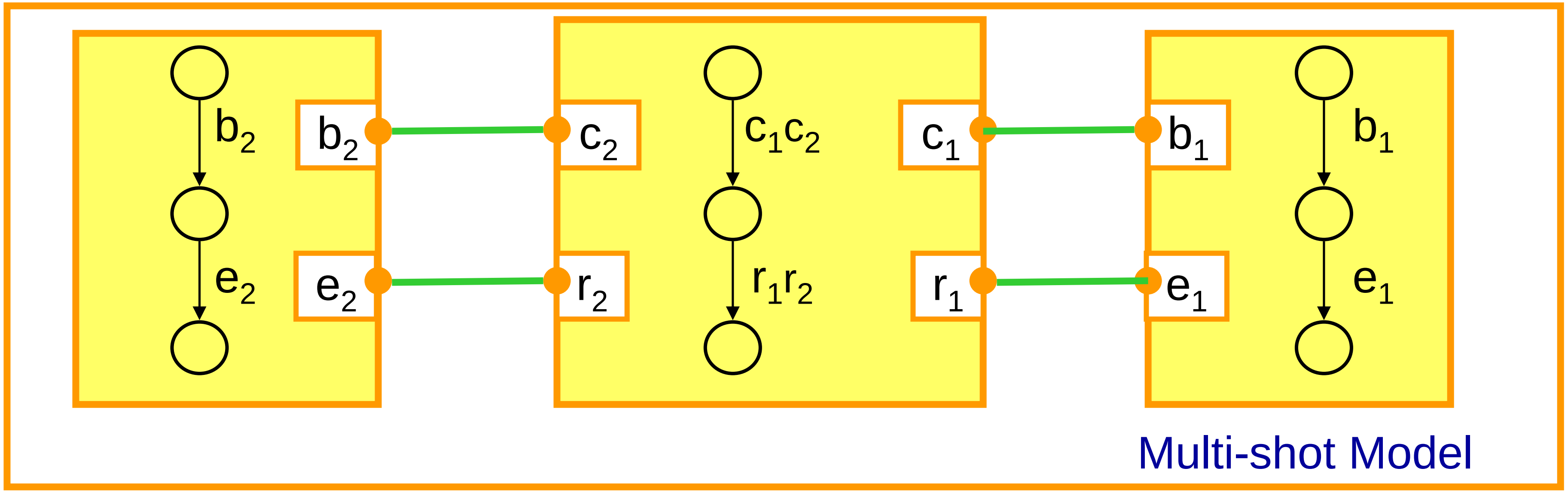
Interactions

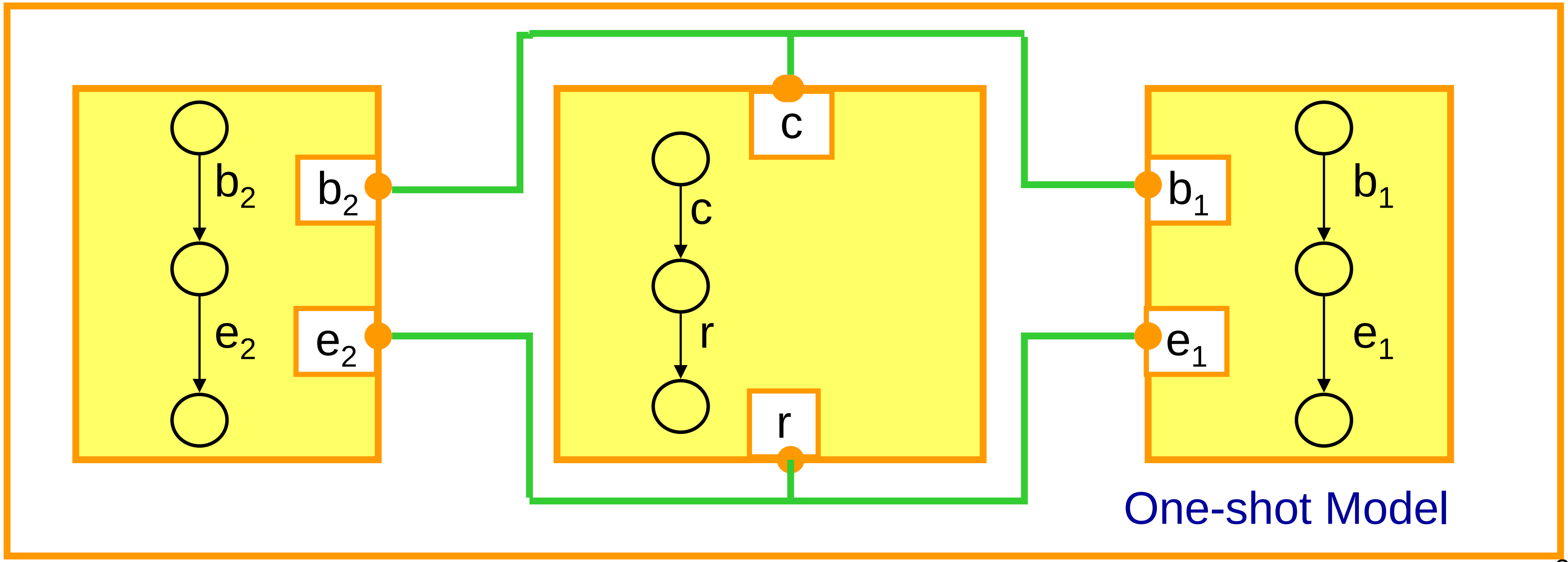
$$\frac{a \in \gamma \wedge \forall i \in [1,n]\ q_i - a \cap P_j \rightarrow_I q'_i}{(q_1 ,.., q_n) - a \rightarrow_{\gamma} (q'_1 ,.., q'_n) \text{ where } q'_I = q_I \text{ if } i \notin I}$$



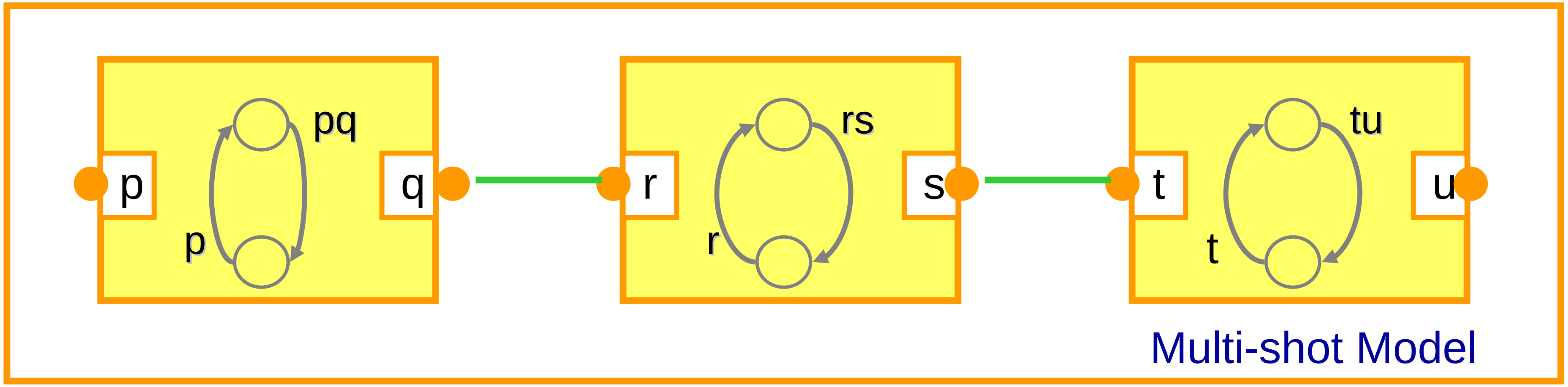
- A set of components *I* offering interactions G_i for $i \in I$
- A set of connectors X_j for $j \in J$
- one-shot: $\gamma = \Pi_{i \in I} G_i' \cap \Sigma_{j \in J} X_j$ multi-shot : $\gamma = \Pi_{i \in I} G_i' \cap \Pi_{j \in J} X_j'$

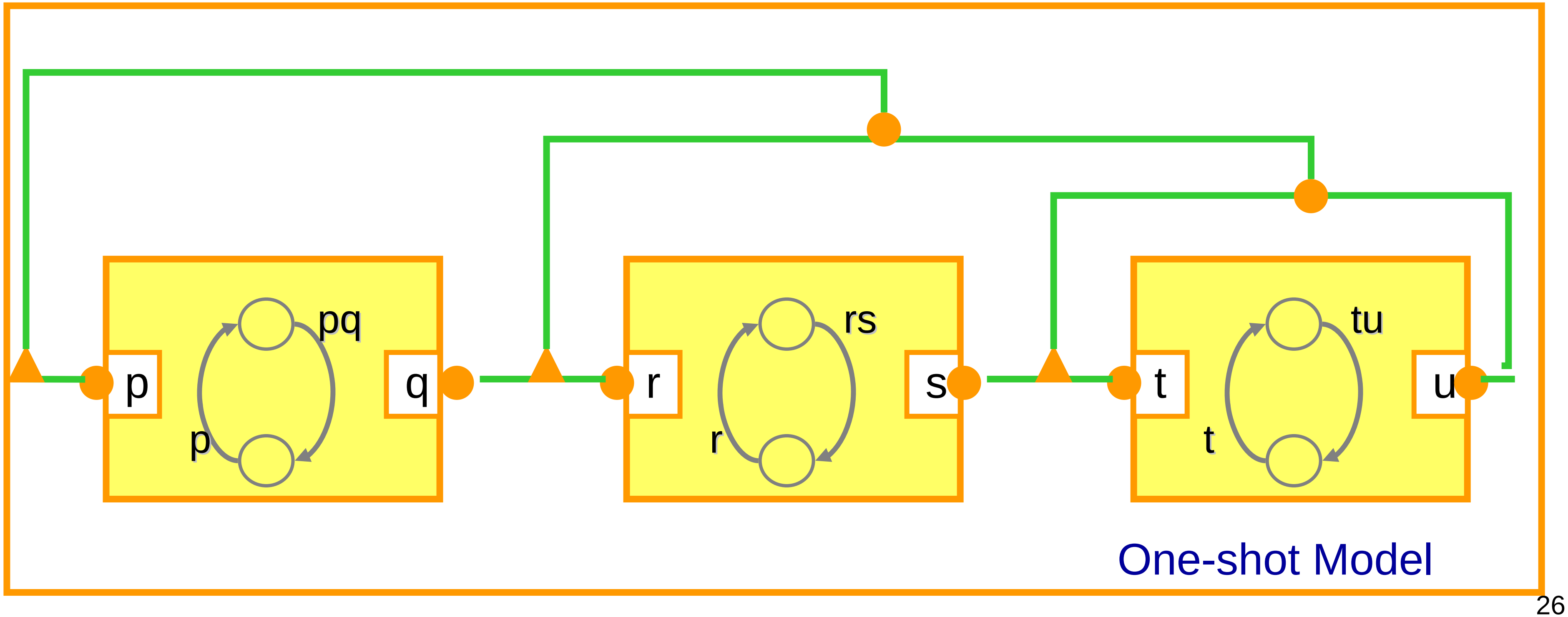
One-shot vs. Multi-shot Semantics: Joint Function Call





One-shot vs. Multi-shot Semantics: mod8 Counter

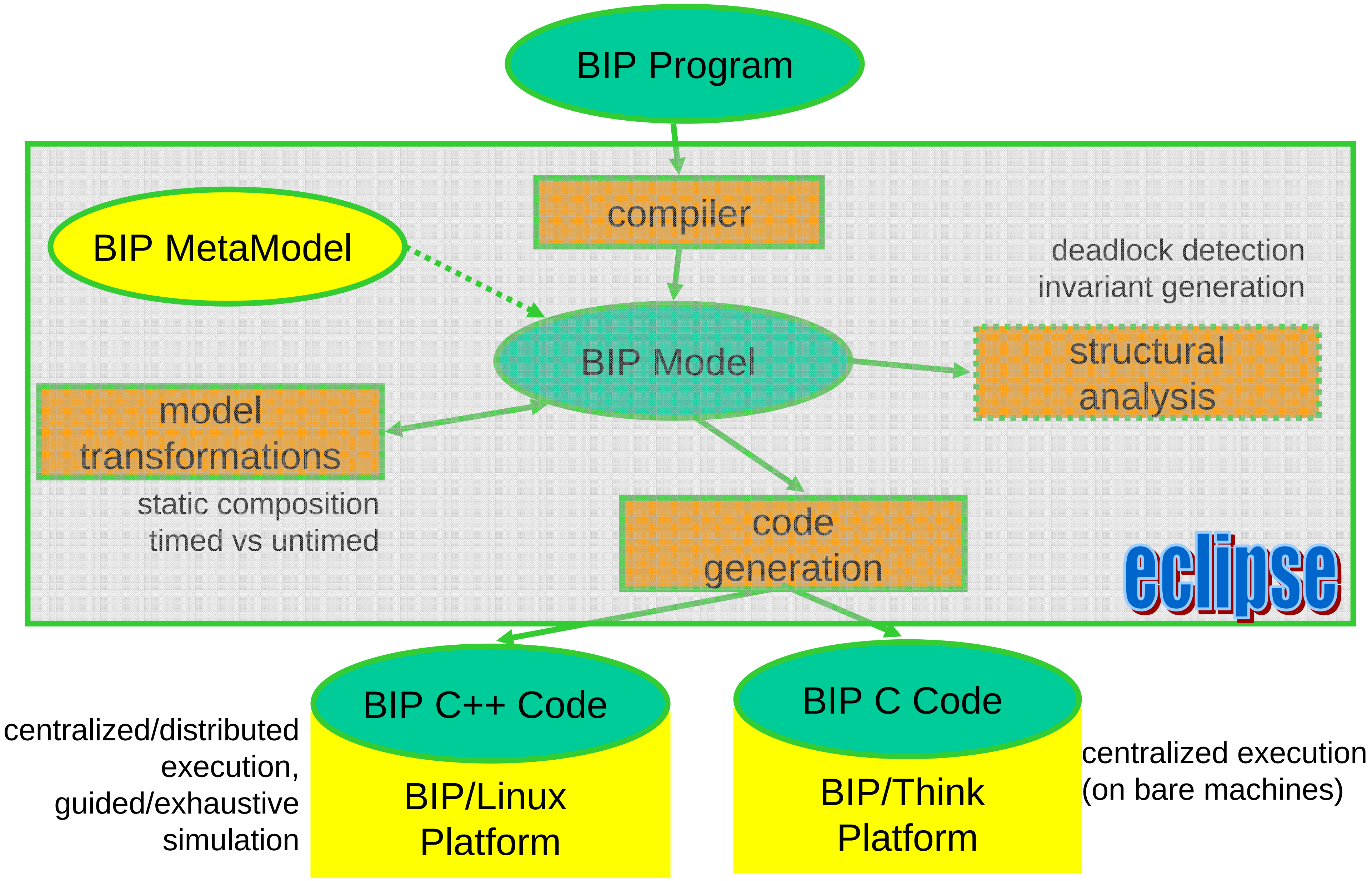




Overview

- BIP: Basic Concepts
- The Algebra of Connectors
- One-shot vs. Multi-shot semantics
- Discussion

Discussion : Implementation



Discussion: the Algebra of Conectors

- Allows compact and structured description of interactions as the structured composition of rendezvous and broadcast by using two operators : typing and fusion.

- Clear separation between behavior and interaction – NOT a process algebra!!

- Framework for studying composability in heterogeneous systems

- Boolean representation allows powerful manipulation, implementation and synthesis - Application for efficient execution of BIP