

The Algebra of Connectors — Structuring Interaction in BIP



Simon Bludze, Joseph Sifakis
{bliudze, sifakis}@imag.fr
VERIMAG, Grenoble



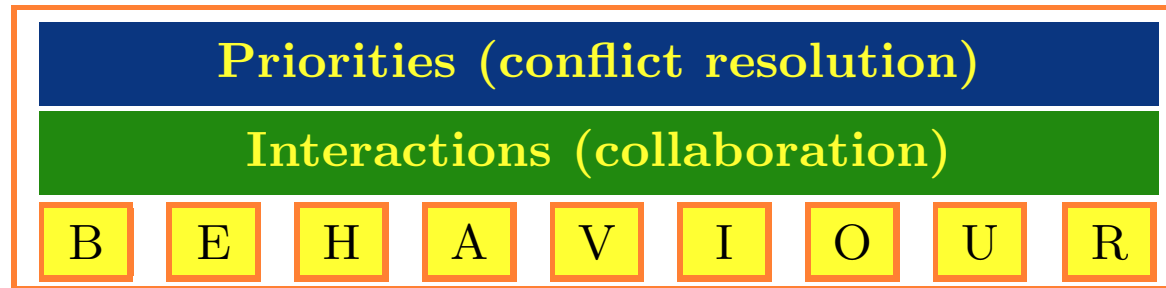
Presentation outline

- Overview of BIP
- Interactions and Connectors
- The Algebra of Connectors
- Applications
- Conclusion



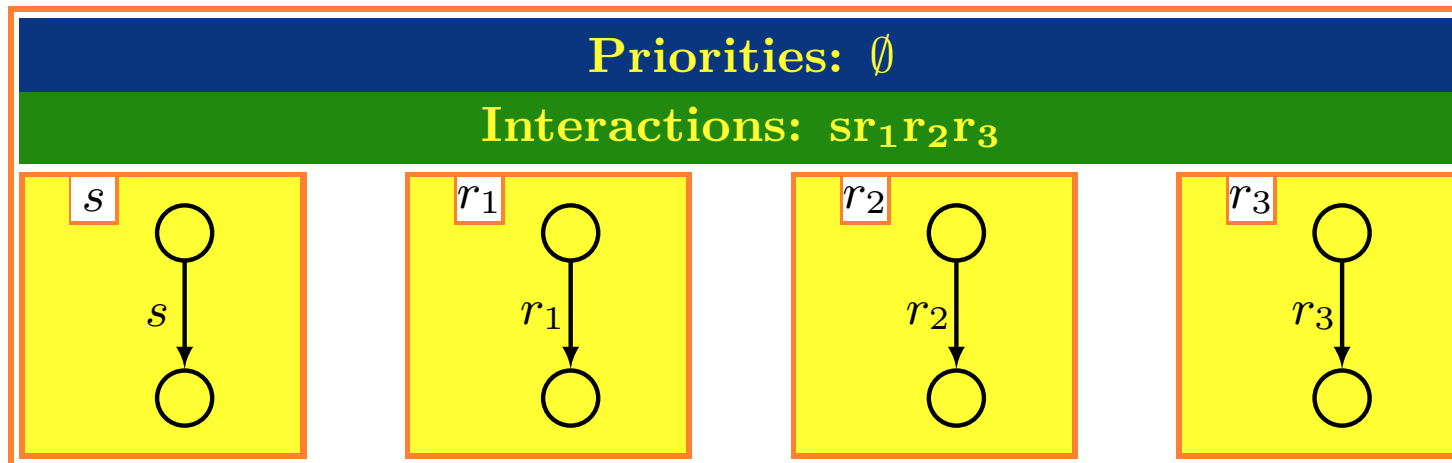
Motivation

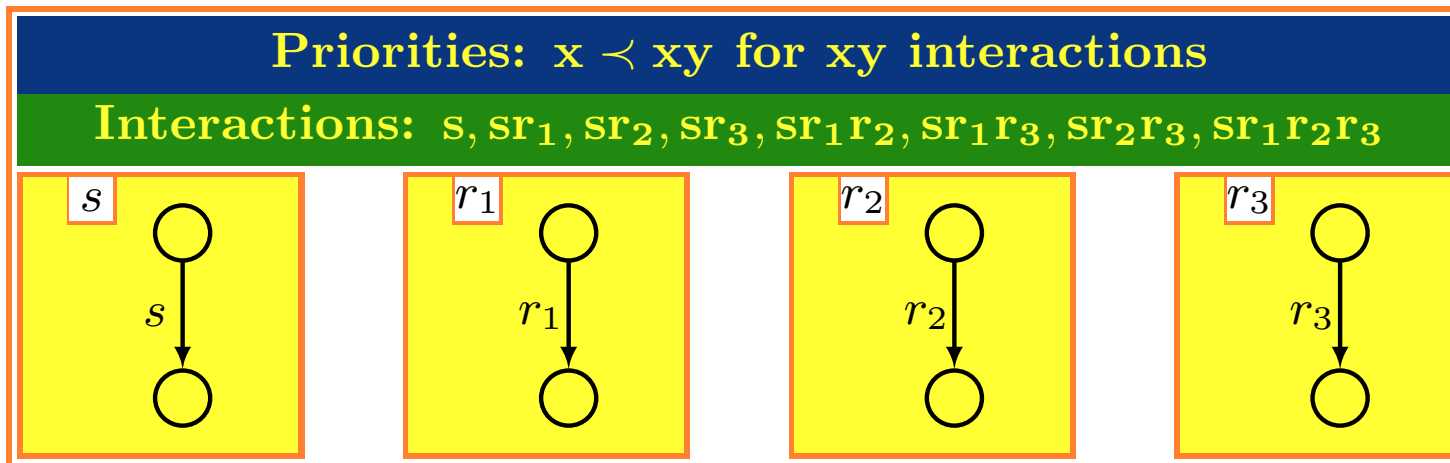
- Develop a unified compositional framework for describing and analysing the interaction between components
 - interaction and system architecture — first class entities
 - minimal set of constructs and principles
 - tangible, well-founded, and organised concepts
(instead of using dispersed mechanisms such as semaphores, monitors, message passing, remote call etc.)
 - full separation of concerns: computation and coordination
- Encompass different kinds of heterogeneity
 - strong and weak synchronisation,
 - synchronous and asynchronous execution.
- Provide automated support for component integration and generation of glue code.

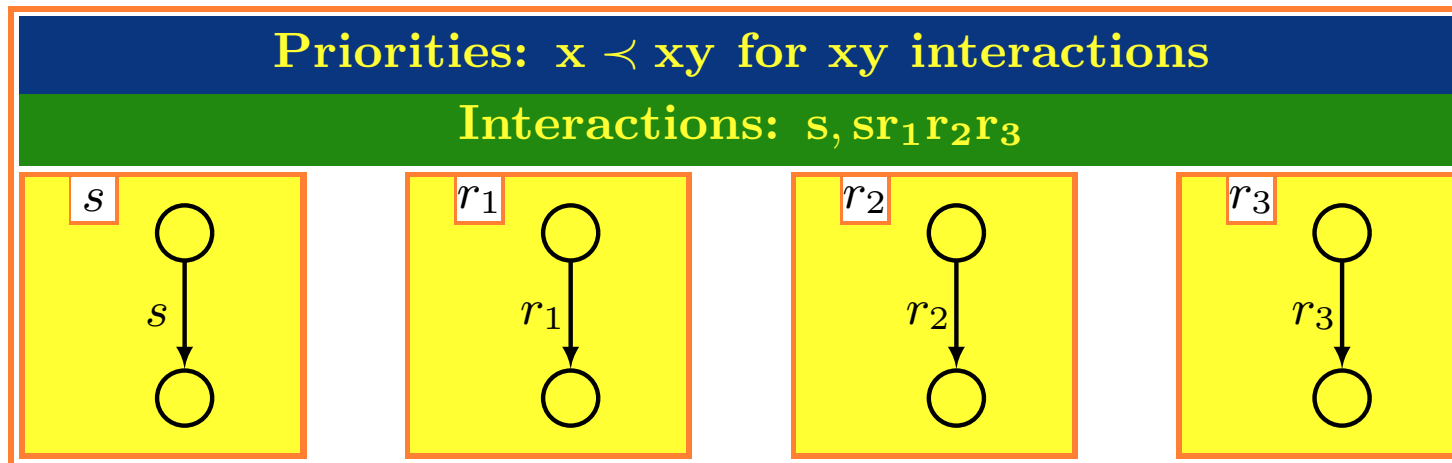


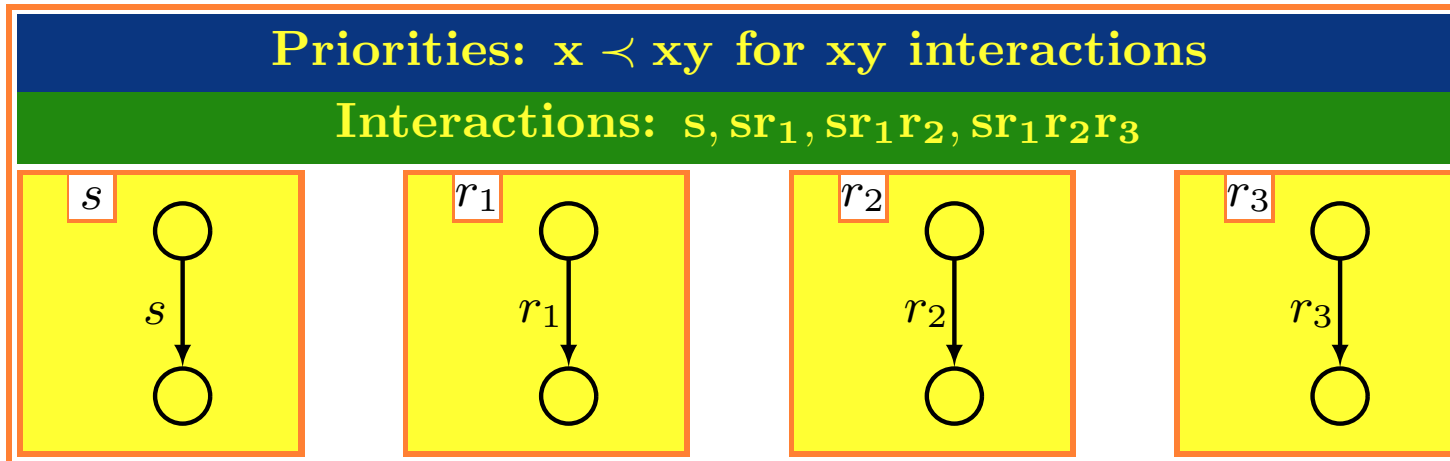
Layered component model

- **Behaviour** — labelled transition systems with communication ports
- **Interaction** — set of interactions (interaction = set of ports)
- **Priorities** — order on interactions









A family of atomic components $\{B_i\}_{i=1}^n$ with $B_i = (Q_i, 2^{P_i}, \rightarrow_i)$ | \rightsquigarrow $\gamma(B_1, \dots, B_n)$ — product automaton
 A set of interactions $\gamma \in 2^{2^P}$

Interactions (n -ary strong synchronisation)

$$\frac{a \in \gamma \quad \wedge \quad \forall i \in I, q_i \xrightarrow{a \cap P_i} q'_i}{(q_1, \dots, q_n) \xrightarrow{a} (q'_1, \dots, q'_n)}$$

Priorities (π is an order on 2^{2^P})

$$\frac{q \xrightarrow{a} q' \quad \wedge \quad \nexists a' : (a \prec a' \wedge q \xrightarrow{a'} q')}{q \xrightarrow{\pi} q'}$$

Other parallel composition operators (e.g. CCS, CSP) can be expressed in BIP.



Presentation outline

- Overview of BIP
- Interactions and Connectors
- The Algebra of Connectors
- Applications
- Conclusion



The Algebra of Interactions $\mathcal{AI}(P)$

Syntax: $x ::= 0 \mid 1 \mid p \mid x \cdot x \mid x + x \mid (x)$

Axioms:

- + union idempotent, associative, commutative, identity 0
- synchronisation idempotent, associative, commutative, identity 1, absorbing 0
distributes over union

Examples: $s + s r_1 + s r_2 + s r_1 r_2 = s(1 + r_1)(1 + r_2)$ broadcast
 $s + s r_1 + s r_1 r_2 = s(1 + r_1(1 + r_2))$ causality chain

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

$$\|0\| = \emptyset,$$

$$\|1\| = \{\emptyset\},$$

$$\|p\| = \left\{ \{p\} \right\}, \text{ for any } p \in P,$$

$$\|x_1 + x_2\| = \|x_1\| \cup \|x_2\|, \text{ for any } x_1, x_2 \in \mathcal{AI}(P),$$

$$\|x_1 \cdot x_2\| = \left\{ a_1 \cup a_2 \mid a_1 \in \|x_1\|, a_2 \in \|x_2\| \right\}, \text{ for any } x_1, x_2 \in \mathcal{AI}(P).$$

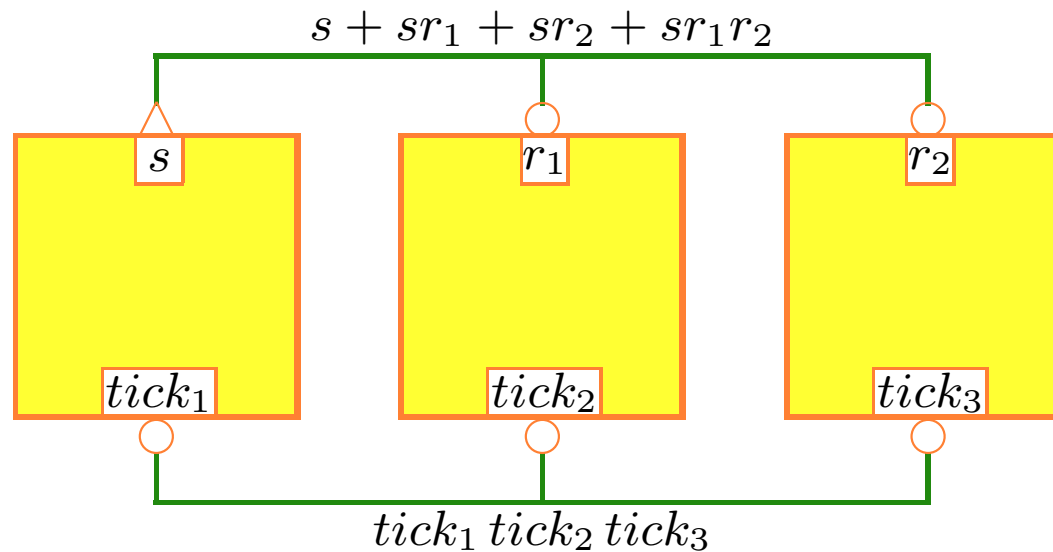
$\mathcal{AI}(P)$	$\mathbb{B}[P]$
0	<i>false</i>
1 p q pq	$\bar{p}\bar{q}$ $p\bar{q}$ $\bar{p}q$ pq
$p+1$ $q+1$ $pq+1$ $p+q$ $p+pq$ $q+pq$	\bar{q} \bar{p} $\bar{p}\bar{q} \vee pq$ $p\bar{q} \vee \bar{p}q$ p q
$p+q+1$ $pq+p+1$ $pq+q+1$ $pq+p+q$	$\bar{p} \vee \bar{q}$ $p \vee \bar{q}$ $\bar{p} \vee q$ $p \vee q$
$pq+p+q+1$	<i>true</i>

Boolean function representation depends on the set P :

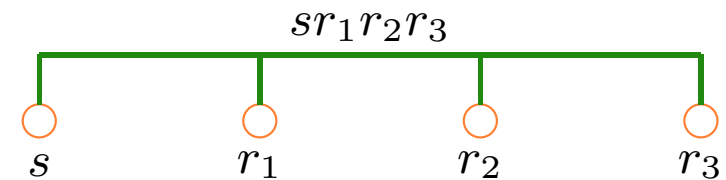
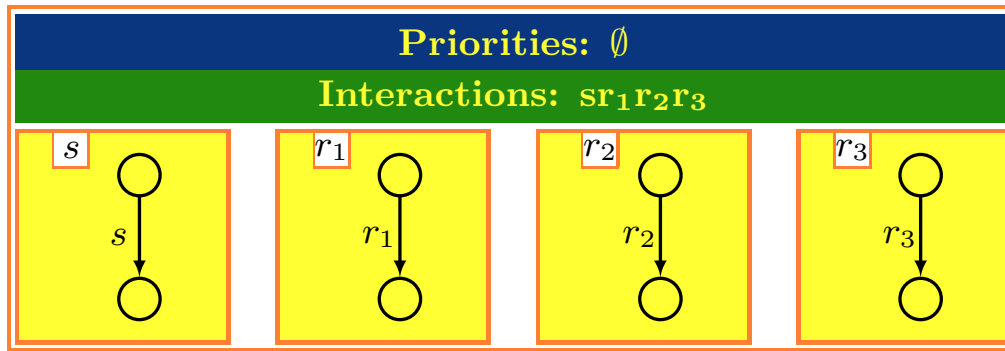
$$\mathcal{AI}(P) : pq \begin{cases} \rightarrow pq, & \text{if } P = \{p, q\} \\ \rightarrow pq\bar{r}\bar{s}, & \text{if } P = \{p, q, r, s\} \end{cases}$$

Synchronisation in $\mathcal{AI}(P)$ is represented by simple concatenation

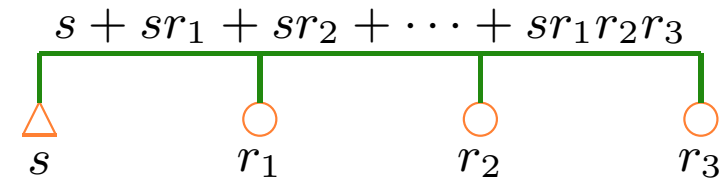
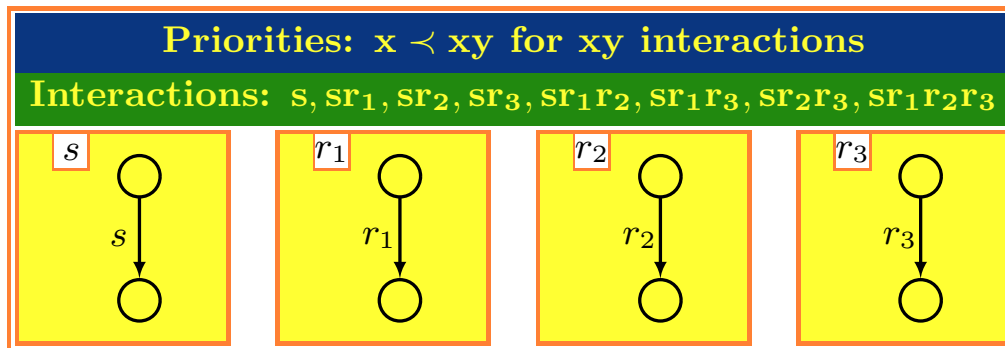
- A **connector** is a set of ports which can be involved in an interaction.
- Port attributes (**trigger** Δ , **synchron** \bigcirc) determine the synchronisation type.
- An **interaction** in a connector is a subset of ports such that either it contains a trigger or it is maximal.



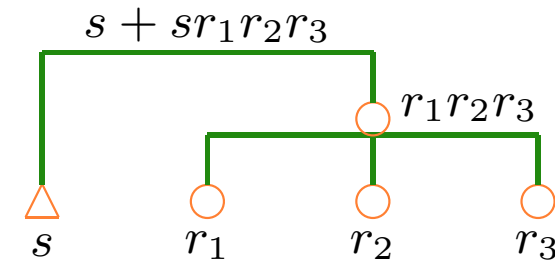
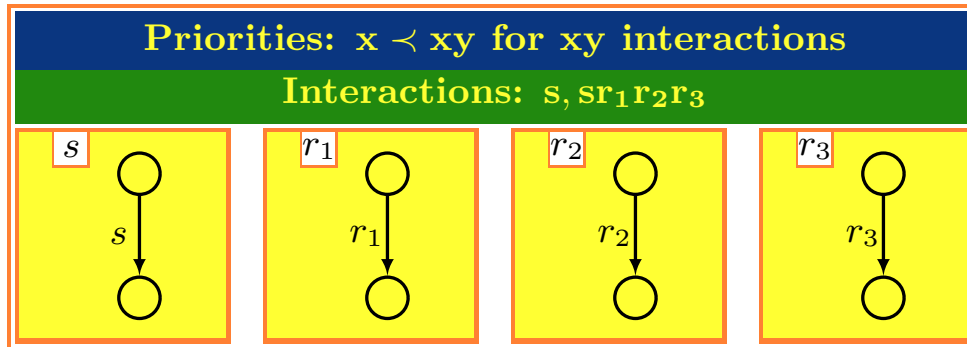
Rendezvous



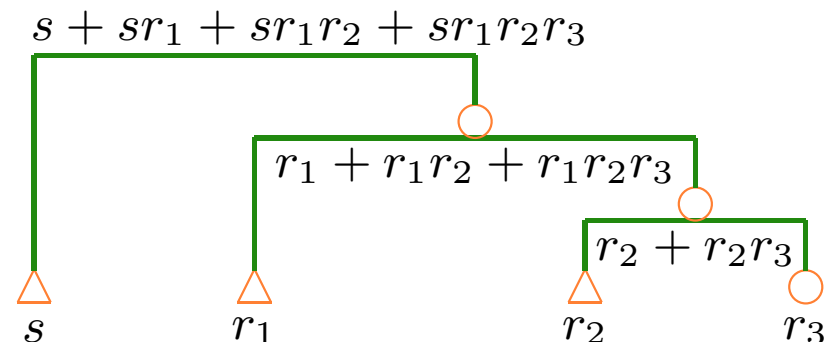
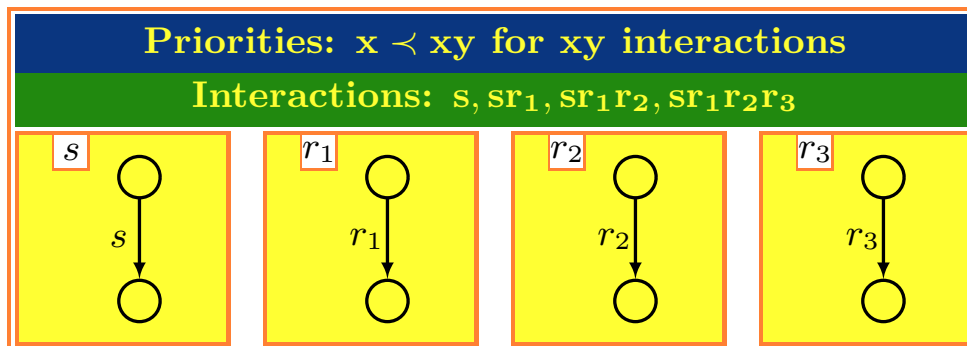
Broadcast



Atomic broadcast



Causality chain





Presentation outline

- Overview of BIP
- Interactions and Connectors
- The Algebra of Connectors
- Applications
- Conclusion



The Algebra of Connectors $\mathcal{AC}(P)$

Syntax:

$s ::= [0] \mid [1] \mid [p] \mid [x] \quad (\textit{synchrons})$

$t ::= [0]' \mid [1]' \mid [p]' \mid [x]' \quad (\textit{triggers})$

$x ::= s \mid t \mid x \cdot x \mid x + x \mid (x)$

Operators:

$+$ union idempotent, associative, commutative, identity $[0]$

\cdot fusion idempotent, associative, commutative, identity $[1]$
distributes over union ($[0]$ is **not** absorbing)

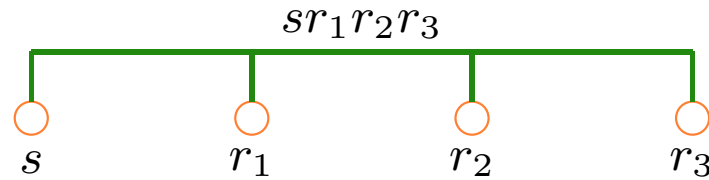
$[\cdot], [\cdot]'$ typing (often denoted $[\cdot]^\alpha$ for some trigger/synchron typing α)

Semantics: is given by a function $|\cdot| : \mathcal{AC}(P) \rightarrow \mathcal{AI}(P)$.



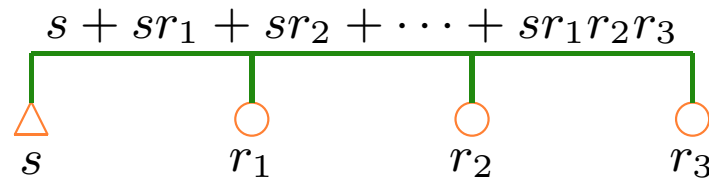
Examples

Rendezvous



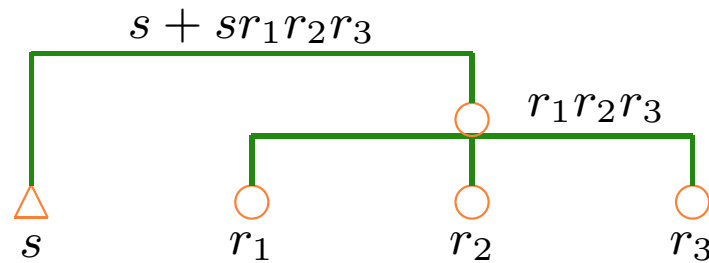
$sr_1r_2r_3$

Broadcast



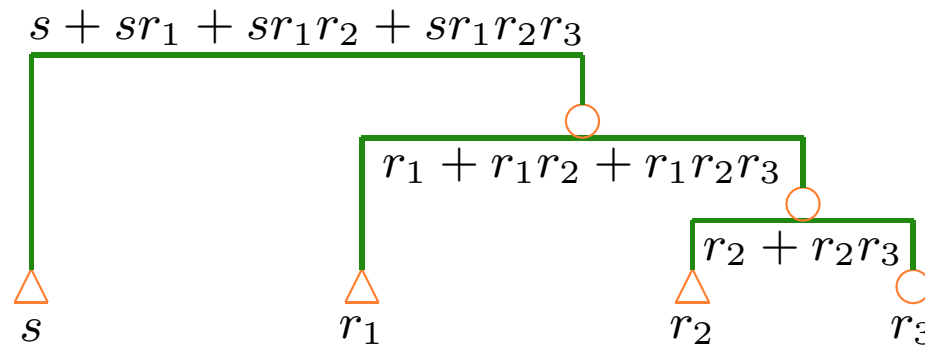
$s'r_1r_2r_3$

Atomic broadcast



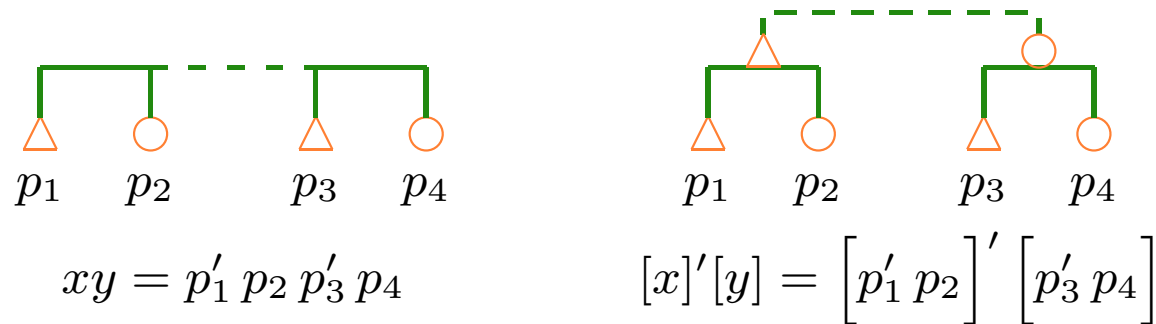
$s'[r_1r_2r_3]$

Causality chain



$s' \left[r'_1 \left[r'_2 r_3 \right] \right]$

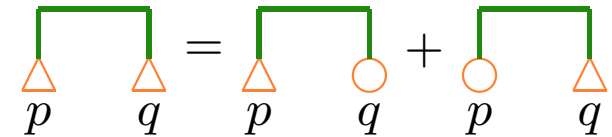
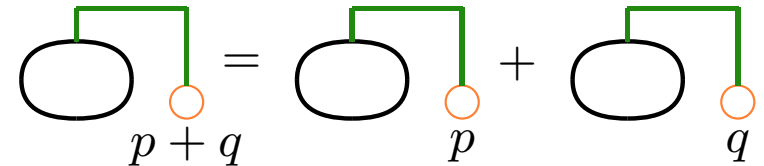
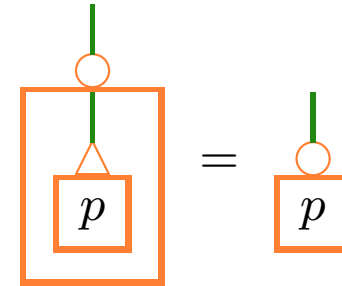
Consider two connectors: $x = p'_1 p_2$ and $y = p'_3 p_4$.



$$\begin{aligned}
 xy &\rightsquigarrow p_1 + p_1 p_2 + p_1 p_3 + p_1 p_4 + p_1 p_2 p_3 + p_1 p_2 p_4 + p_1 p_3 p_4 \\
 &\quad + p_3 + p_2 p_3 + p_3 p_4 + p_2 p_3 p_4 + p_1 p_2 p_3 p_4 \\
 [x]'[y] &\rightsquigarrow p_1 + p_1 p_2 + p_1 p_3 + p_1 p_2 p_3 + p_1 p_3 p_4 + p_1 p_2 p_3 p_4
 \end{aligned}$$

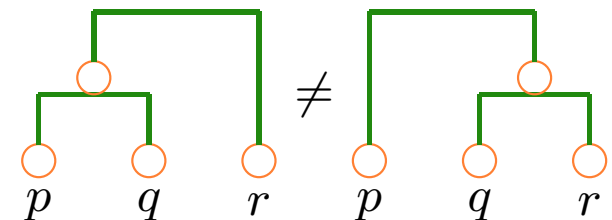
Axioms for **typing** (for arbitrary typing α, β):

1. $[0]' = [0]$,
2. $\left[[x]^\alpha \right]^\beta = [x]^\beta$,
3. $[x + y]^\alpha = [y]^\alpha + [x]^\alpha$,
4. $[x]' \cdot [y]' = [x]' \cdot [y] + [x] \cdot [y]'$.



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

$$[p q] r \neq p [q r]$$





Equivalence vs. Congruence

- $x \simeq y \stackrel{def}{\iff} |x| = |y|$, i.e. they represent the same sets of interactions
 - The axiomatisation of $\mathcal{AC}(P)$ is semantically sound, i.e. $x = y \Rightarrow x \simeq y$.
 - Semantic equivalence is not a congruence (not preserved by fusion)

$$p + pq \simeq p'q, \quad \text{but} \quad pr + pqr \not\simeq p'qr \quad / \quad \simeq p + pq + pr + pqr \quad / ,$$
$$p[qr] \simeq [pq]r, \quad \text{but} \quad s'p[qr] \not\simeq s'[pq]r .$$

- \cong is the largest congruence contained in \simeq
 - A criterion to infer congruence from equivalence is available.
 - Similarly typed semantically equivalent elements are congruent, i.e. for any two connectors $x, y \in \mathcal{AC}(P)$, and any typing α , we have

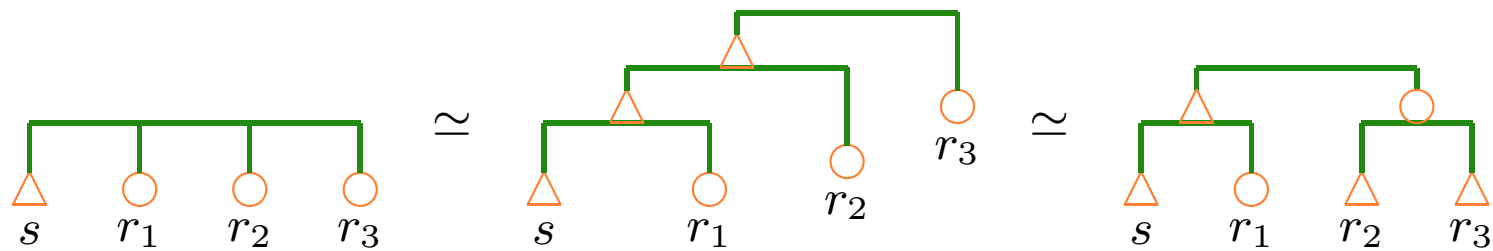
$$x \simeq y \implies [x]^\alpha \cong [y]^\alpha .$$



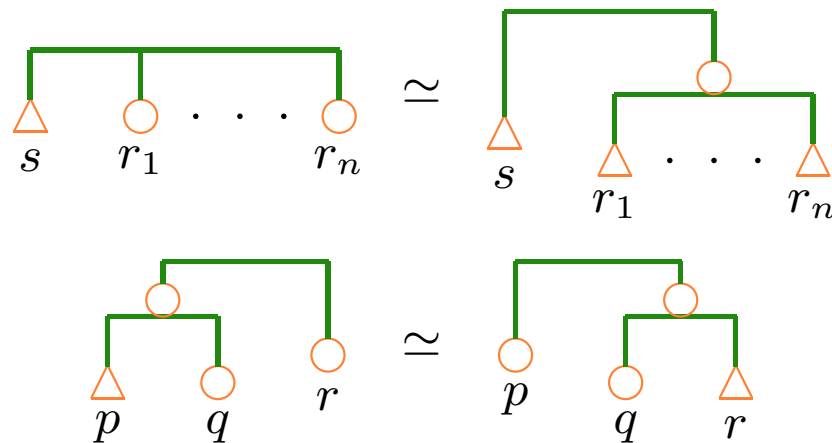
Presentation outline

- Overview of BIP
- Interactions and Connectors
- The Algebra of Connectors
- Applications
- Conclusion

Incremental construction

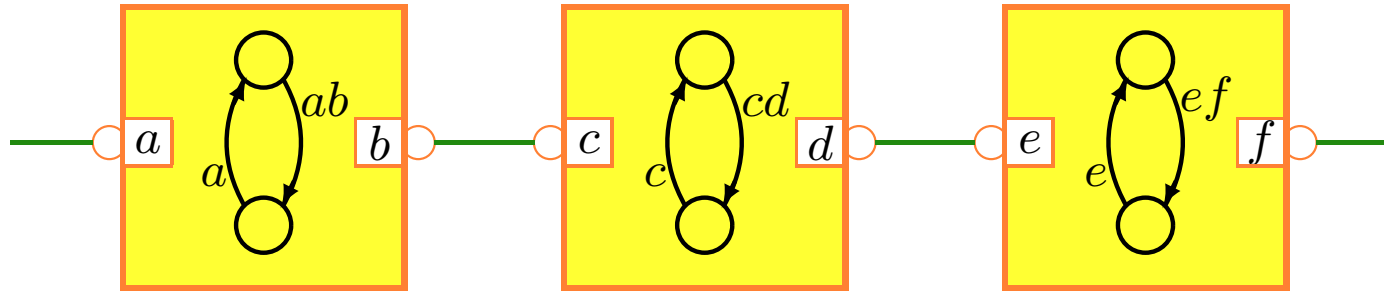


Transformation (separate one port in a connector)



Connector synthesis: Modulo-8 counter

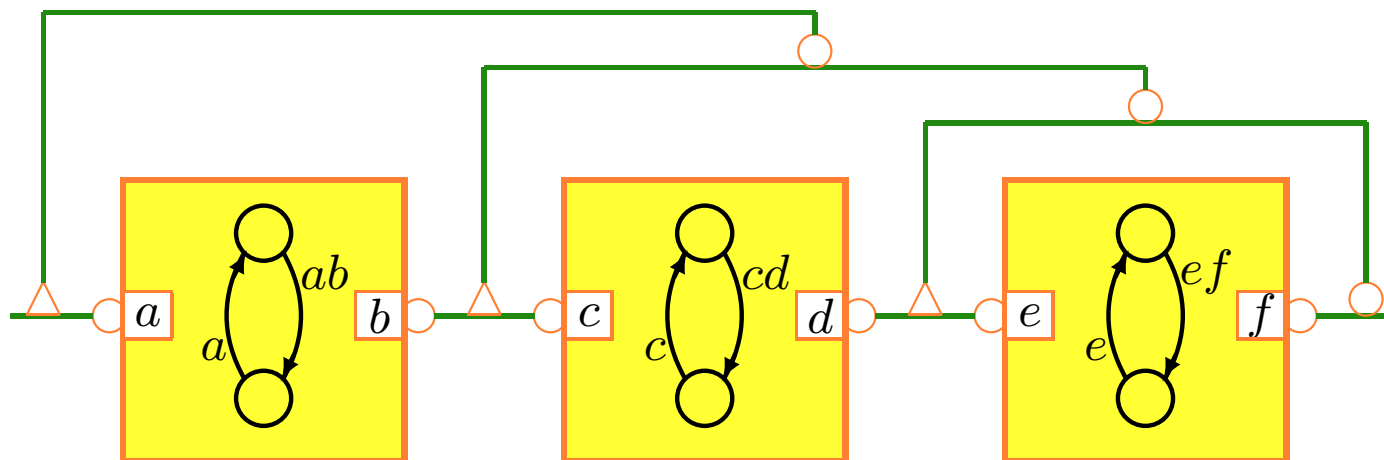
Multi-shot semantics: several connectors can be fired simultaneously



Interactions: $a + abc + abcde + abcdef$

One-shot semantics: one connector can be fired at a time

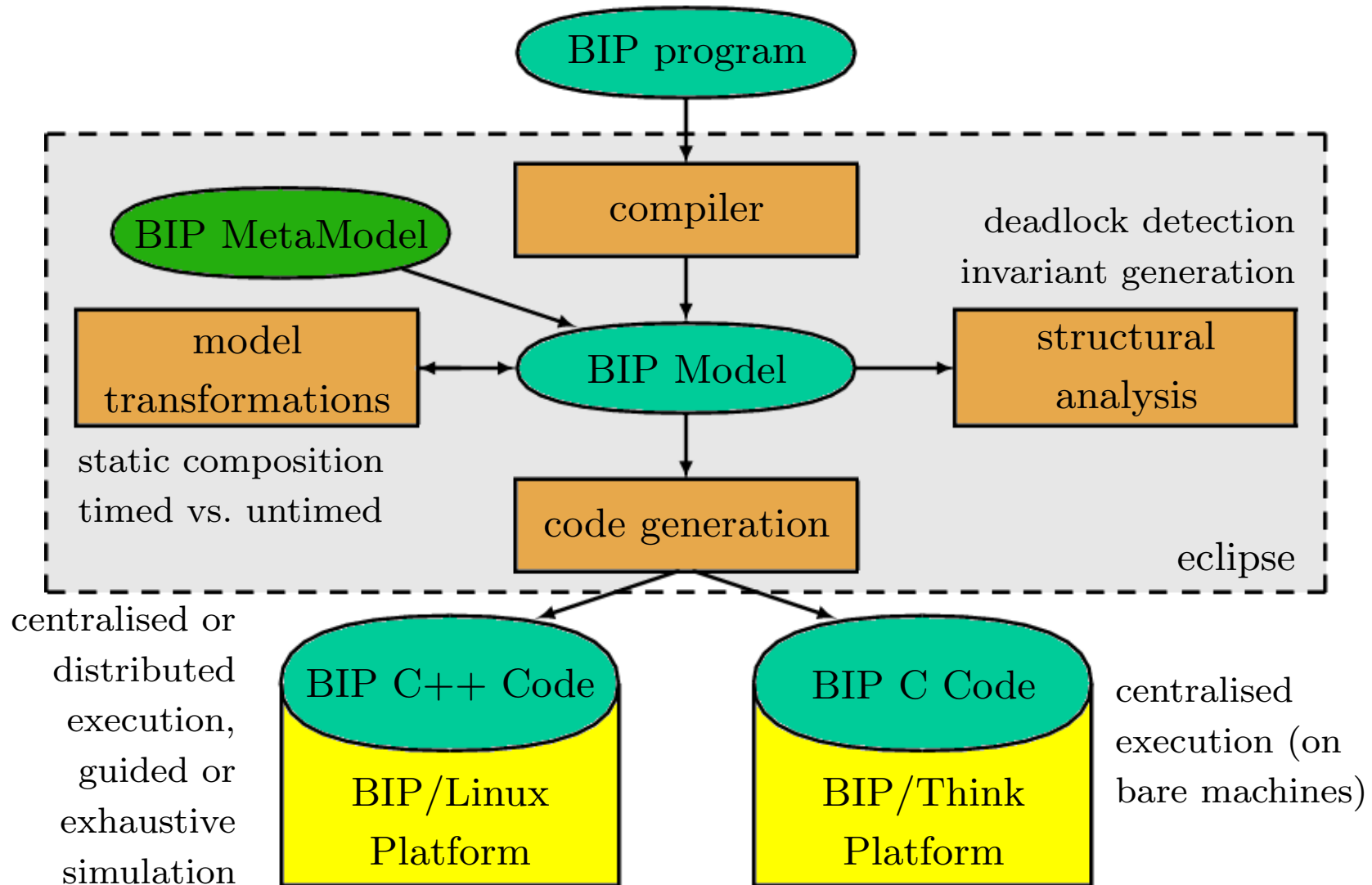
Connector synthesis: $[a + ab]' [c + cd]' [e + ef]' \cap a' [bc]' [de]' f' \simeq a' [[bc]' [[de]' f]]$.





Presentation outline

- Overview of BIP
- Interactions and Connectors
- The Algebra of Connectors
- Applications
- Conclusion





The algebra of connectors

- Allows compact and structured description and analysis of interactions, in terms of two operators admitting a very intuitive interpretation: typing and fusion.
- Differs from process algebras, which do not study interactions as such — these are only a means to compose behaviour.
- Provides basis for the symbolic comparison, transformation, and synthesis of connectors, which can be directly implemented.
- Boolean representation provides powerful techniques for manipulation, implementation, and synthesis.
- Application in BIP to model interaction in non trivial case studies, e.g. TinyOS-based wireless networks, autonomous robot software.



- Causal semantics for $\mathcal{AC}(P)$
- Temporised connectors
- Algebraic approach to constraints and priorities
- Application to other formalisms encompassing event-based interaction, in particular coordination languages
- Efficient implementation in future versions of BIP.



Annex: Correspondence with boolean functions

A mapping $\beta : \mathcal{AI}(P) \rightarrow \mathbb{B}[P]$

$$\beta(0) = \text{false},$$

$$\beta(p_{i_1} \dots p_{i_k}) = \bigwedge_{j=1}^k p_{i_j} \wedge \bigwedge_{i \notin \{i_j\}} \bar{p}_i, \quad \text{for any } p_{i_1}, \dots, p_{i_k} \in P,$$

$$\beta(x + y) = \beta(x) \vee \beta(y), \quad \text{for any } x, y \in \mathcal{AI}(P),$$

$$\beta(1) = \bigwedge_{p \in P} \bar{p},$$



Annex: Full axiomatisation of \cong

For $x, y, z \in \mathcal{AC}(P)$ and arbitrary typing α, β :

Basic:

1. $[0]' = [0]$,
2. $\left[[x]^\alpha\right]^\beta = [x]^\beta$,
3. $[x + y]^\alpha = [y]^\alpha + [x]^\alpha$,

Additional:

1. $[x]' [0] = [x]'$,
2. $\left[[x] [0]\right] = [0]$,
3. $[x]' + [x] = [x]'$,
4. $[x]' [y]' = [x]' [y] + [y]'$,
5. $[x]' [y] [z] = [x]' \left[[y]' [z]'\right]$,
6. $[x]' [y] = \left[[x]' [y]\right]' + [0] [y]$,
7. $[x] [y] = \left[[x] [y]\right] + [0] [x] [y]$,
8. $\left[[x]' [y]\right] = [x] + \left[[x] [y]\right]$,
9. $\left[\left[[x] [y]\right] [z]\right] = \left[[x] [y] [z]\right]$.



Annex: Semantics of $\mathcal{AC}(P)$

The semantics of $\mathcal{AC}(P)$ is given by the function $|\cdot| : \mathcal{AC}(P) \rightarrow \mathcal{AI}(P)$, defined by the rules

$$\begin{aligned} |[p]| &= p, \text{ for } p \in P \cup \{0, 1\} \\ |x_1 + x_2| &= |x_1| + |x_2|, \\ \left| \prod_{i=1}^n [x_i] \right| &= \prod_{i=1}^n |x_i|, \\ \left| \prod_{i=1}^n [x_i]' \cdot \prod_{j=1}^m [y_j] \right| &= \sum_{i=1}^n |x_i| \cdot \left(\prod_{k \neq i} (1 + |x_k|) \cdot \prod_{j=1}^m (1 + |y_j|) \right), \end{aligned}$$

for $x, x_1, \dots, x_n, y_1, \dots, y_m \in \mathcal{AC}(P)$.