The Algebra of Connectors — Structuring Interaction in BIP

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

Presentation outline

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

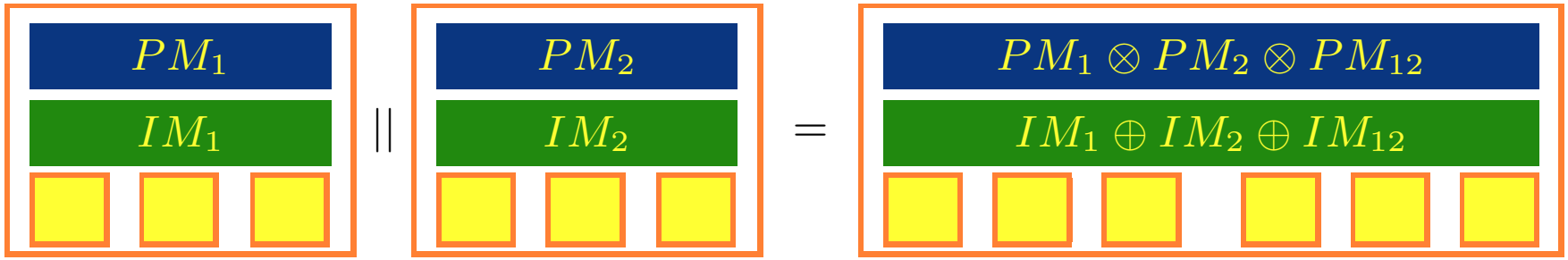
Motivation

- 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.

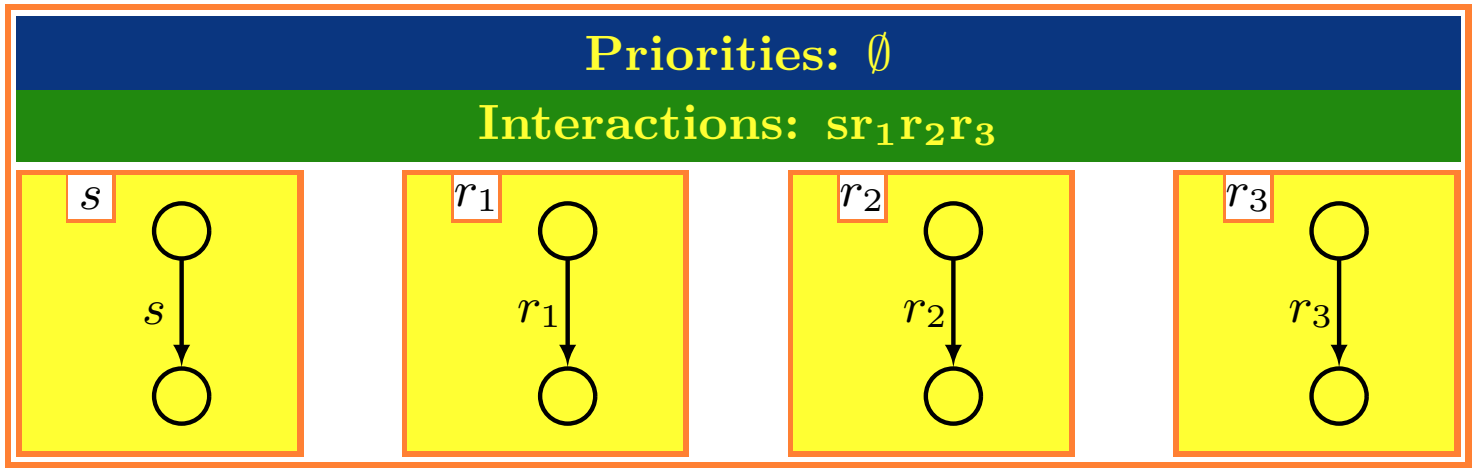
Basic model of BIP

Layered component model

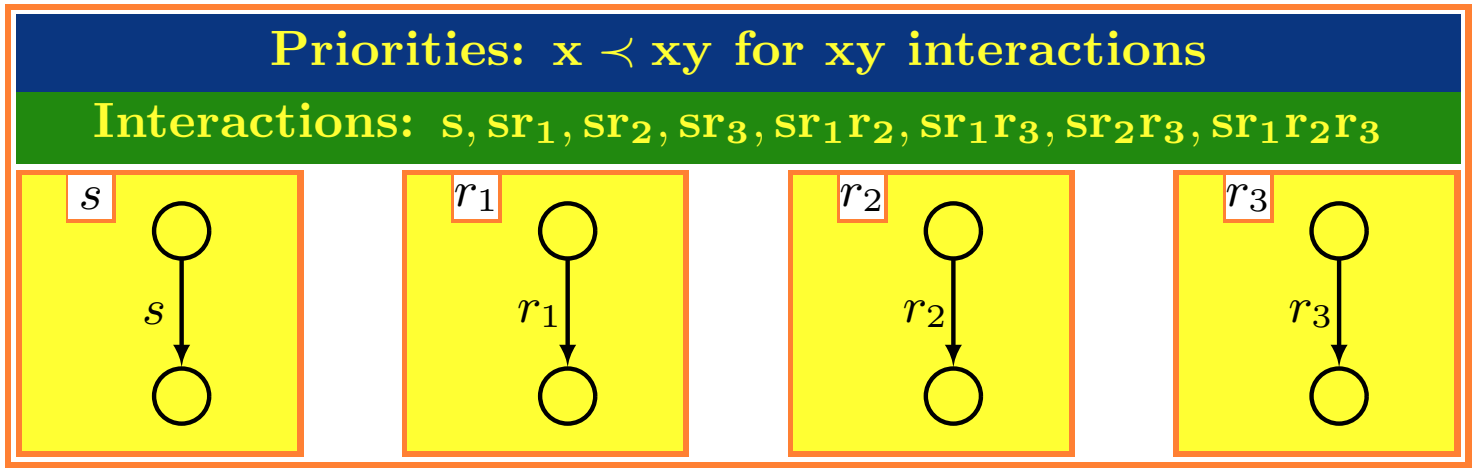
- **Behaviour** — labelled transition systems with communication ports
- **Interaction** — set of interactions (interaction = set of ports)
- **Priorities** — order on interactions



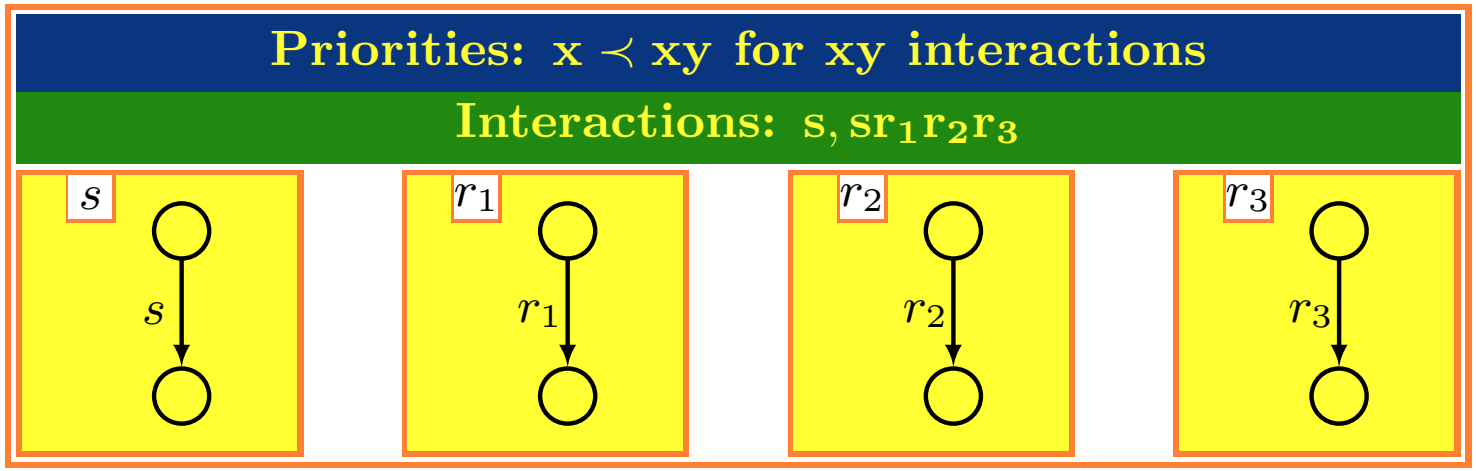
Rendezvous



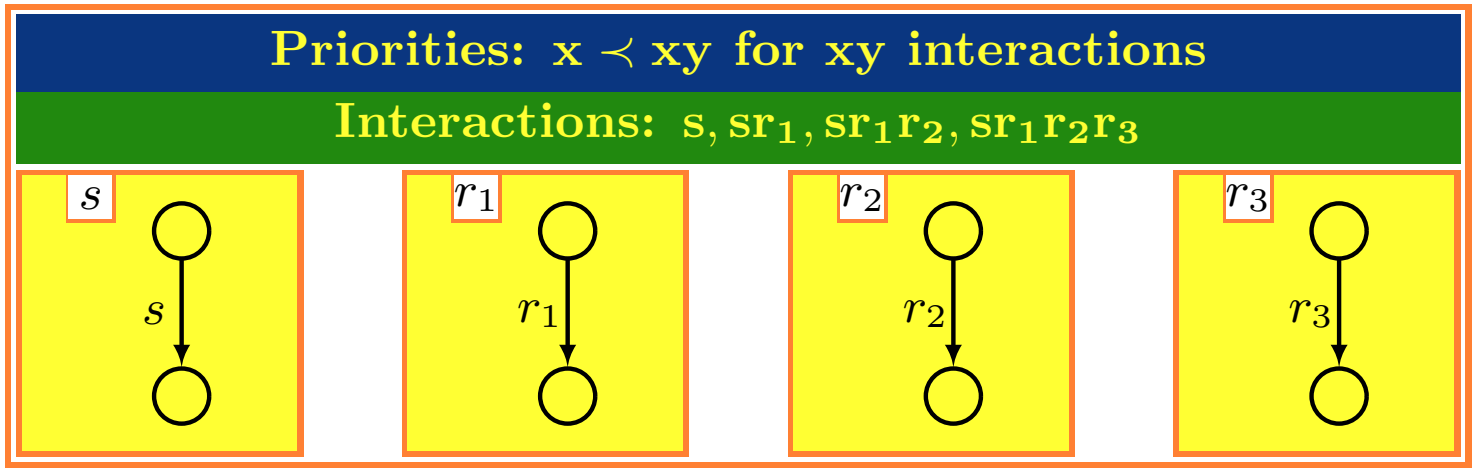
Broadcast


Priorities: x ≺ xy for xy interactions
Interactions: s, sr1, sr2, sr3, sr1r2, sr1r3, sr2r3, sr1r2r3
s
s
r1
r1
r2
r2
r3
r3

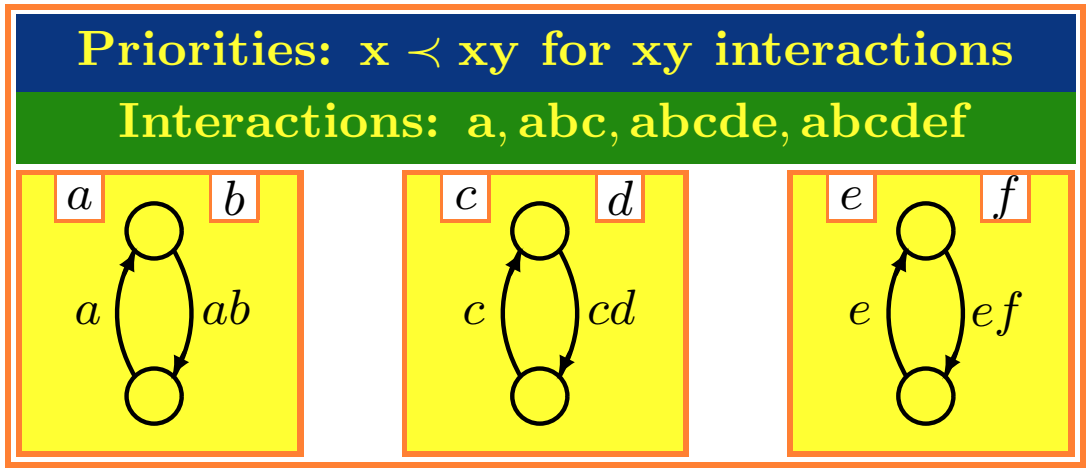
Atomic broadcast



Causality chain



Modulo-8 counter



Semantic model of BIP

A family of atomic components $\{B_i\}_{i=1}^n$ with $B_i = (Q_i, 2^{P_i}, \rightarrow_i)$

A set of interactions $\gamma \in 2^{2^P}$

A strict partial order π on 2^P

$\rightsquigarrow$ $\pi\gamma(B_1, \ldots, B_n)$ — product automaton

$/ P = \bigcup_{i=1}^n P_i /$

Interactions (n-ary strong synchronisation)

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

Priorities

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

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: $x ::= 0 \mid 1 \mid p \mid x \cdot x \mid x + x \mid (x)$, with $p \in P$

Axioms:

$+$	union	idempotent, associative, commutative, identity 0
$\cdot$	synchronisation	idempotent, associative, commutative, identity 1, absorbing 0
		distributes over union

Examples:

$$s + s\,r_1 + s\,r_2 + s\,r_1\,r_2 = s(1 + r_1)(1 + r_2) \quad \text{broadcast}$$

$$s + s\,r_1 + s\,r_1\,r_2 = s(1 + r_1(1 + r_2)) \quad \text{causality chain}$$

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

$$\begin{aligned}
\|0\| &= \emptyset, \\
\|1\| &= \{\emptyset\}, \\
\|p\| &= \Big\{\{p\}\Big\}, \text{ for any } p \in P, \\
\|x_1 + x_2\| &= \|x_1\| \cup \|x_2\|, \text{ for any } x_1, x_2 \in \mathcal{AI}(P), \\
\|x_1 \cdot x_2\| &= \Big\{a_1 \cup a_2 \,\Big|\, a_1 \in \|x_1\|, a_2 \in \|x_2\|\Big\}, \text{ for any } x_1, x_2 \in \mathcal{AI}(P)\,.
\end{aligned}$$

Correspondence with boolean functions

$\mathcal{AI}(P)$	$\mathbb{B}[P]$
0	$false$
$1 \quad p \quad q \quad p\,q$	$\overline{p}\,\overline{q} \quad p\,\overline{q} \quad \overline{p}\,q \quad p\,q$
$p+1 \quad q+1 \quad p\,q+1 \quad p+q \quad p+p\,q \quad q+p\,q$	$\overline{q} \quad \overline{p} \quad \overline{p}\,\overline{q} \vee p\,q \quad p\,\overline{q} \vee \overline{p}\,q \quad p \quad q$
$p+q+1 \quad p\,q+p+1 \quad p\,q+q+1 \quad p\,q+p+q$	$\overline{p} \vee \overline{q} \quad p \vee \overline{q} \quad \overline{p} \vee q \quad p \vee q$
$p\,q+p+q+1$	$true$

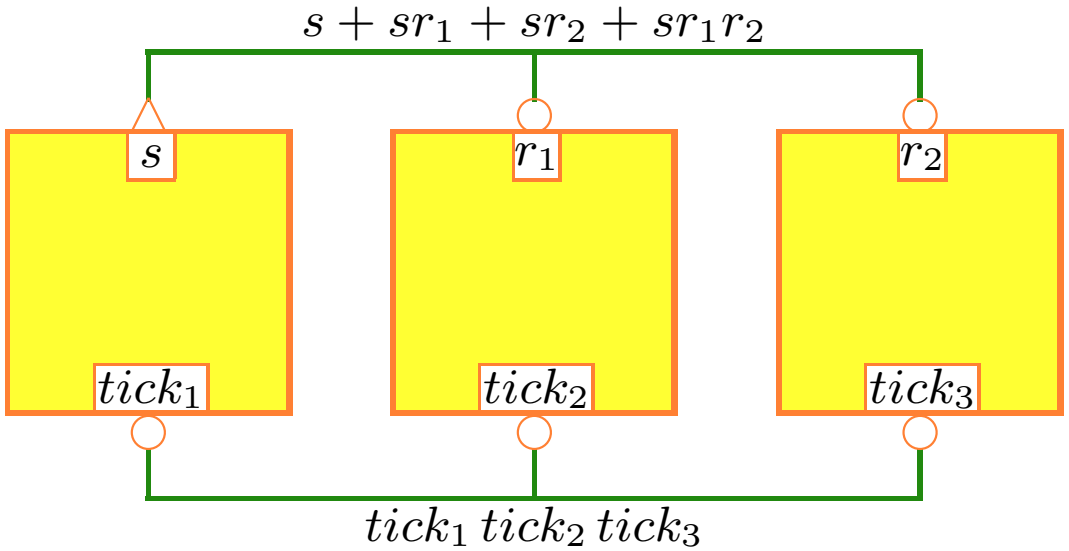
Boolean function representation depends on the set P:

$$\mathcal{AI}(P):\ p\,q \begin{cases} \nearrow\ p\,q\,, & \text{if } P=\{p,q\} \\ \searrow\ p\,q\,\overline{r}\,\overline{s}\,, & \text{if } P=\{p,q,r,s\} \end{cases}$$

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

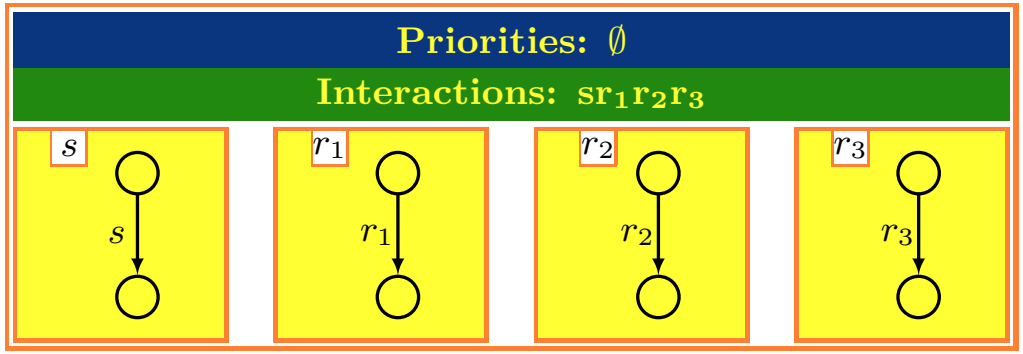
Interaction modelling: Basic connectors

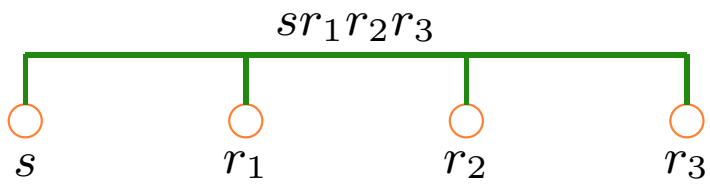
- A **connector** is a set of ports which can be involved in an interaction.
- Port attributes (**trigger** △, **synchron** ○) determine the synchronisation type.
- An **interaction** in a connector is a subset of ports such that either it contains a trigger or it is maximal.



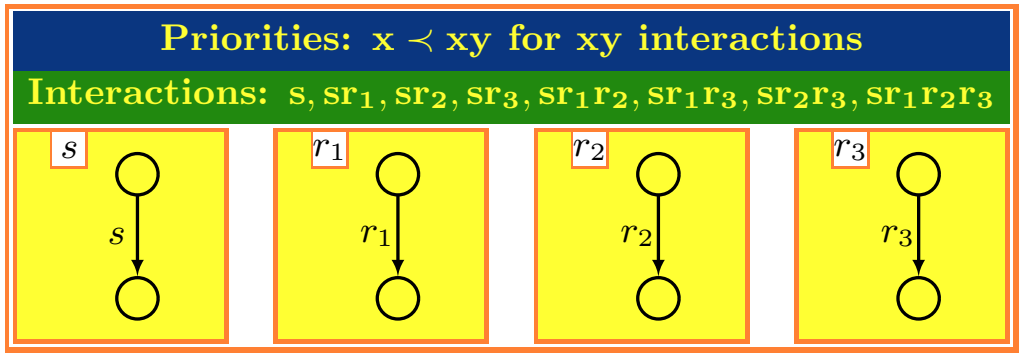
Interaction modelling: Flat connectors

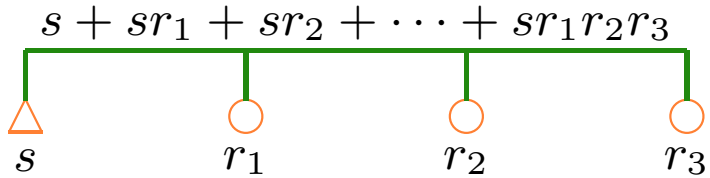
Rendezvous





Broadcast

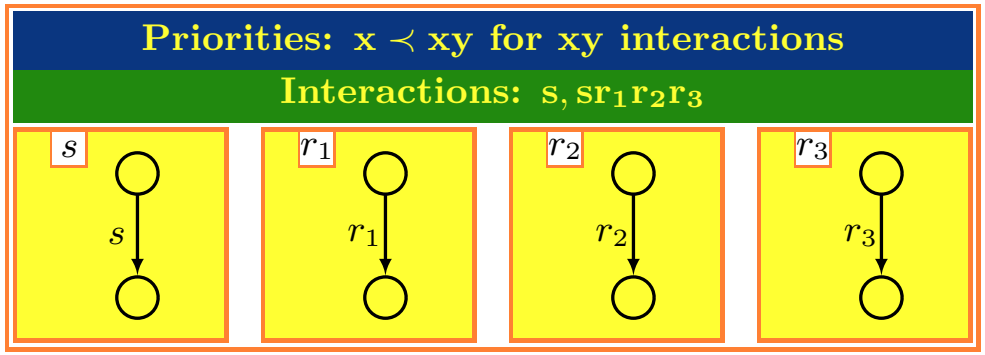


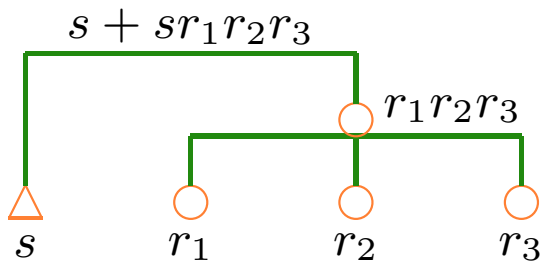


$$/s(1+r_1)(1+r_2)(1+r_3)/$$

Interaction modelling: Hierarchical connectors

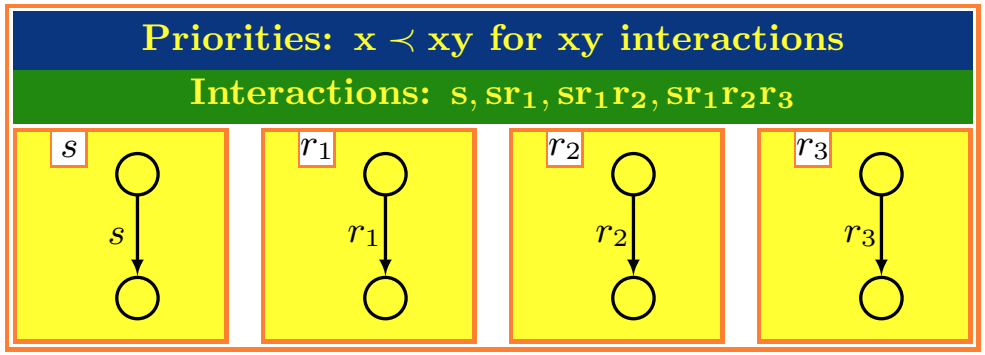
Atomic boadcast

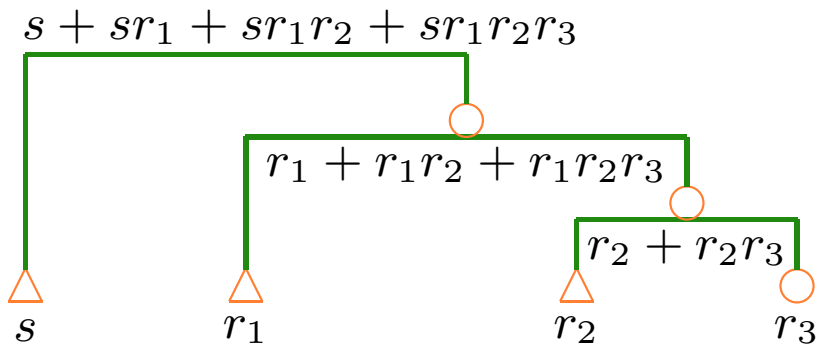




$$/s(1 + r_1r_2r_3)/$$

Causality chain





$$/s\Big(1 + r_1(1 + r_2(1 + r_3))\Big)/$$

Presentation outline

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

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

Syntax:

$$
\begin{array}{lcll}
s & ::= & [0] \mid [1] \mid [p] \mid [x] & (\textit{synchrons}) \\
t & ::= & [0]' \mid [1]' \mid [p]' \mid [x]' & (\textit{triggers}) \\
x & ::= & s \mid t \mid x \cdot x \mid x + x \mid (x) &
\end{array}
$$

Operators:

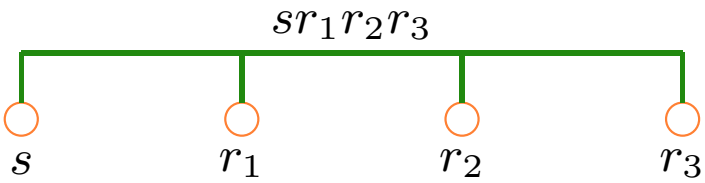
$+$	union	idempotent, associative, commutative, identity $[0]$
$\cdot$	fusion	idempotent, associative, commutative, identity $[1]$
		distributes over union ($[0]$ is **not** absorbing)
$[\cdot]$, $[\cdot]'$	typing	(often denoted $[\cdot]^\alpha$ for some trigger/synchron typing α)

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

$$|p'qr| \stackrel{def}{=} p(1+q)(1+r)$$

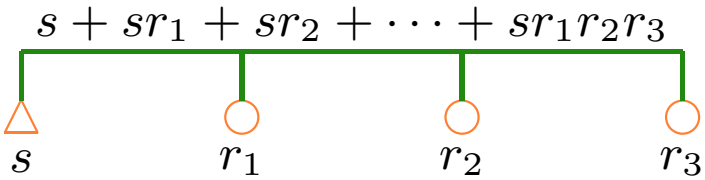
Examples

Rendezvous



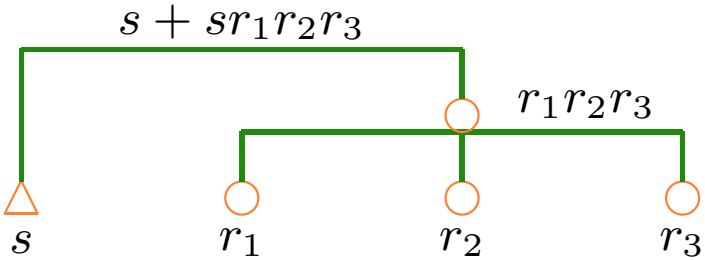
$sr_1r_2r_3$

Broadcast



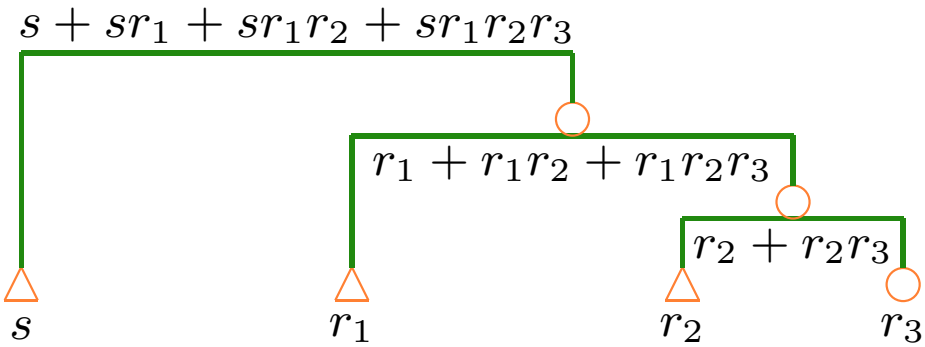
$s'r_1r_2r_3$

Atomic boadcast



$s'[r_1r_2r_3]$

Causality chain

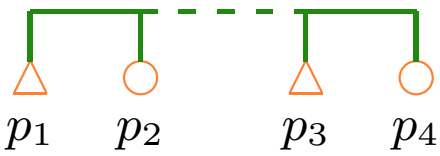


$s'\left[r_1'[r_2'r_3]\right]$

Examples: Fusion vs. Typing

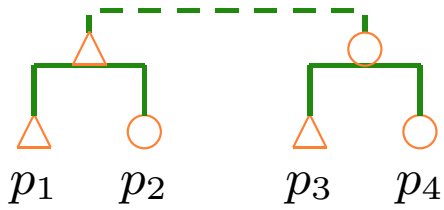
Consider two connectors: $x = p_1' \, p_2$ and $y = p_3' \, p_4$.

Fusion



$$xy = p_1' \, p_2 \, p_3' \, p_4$$

"Typed fusion"



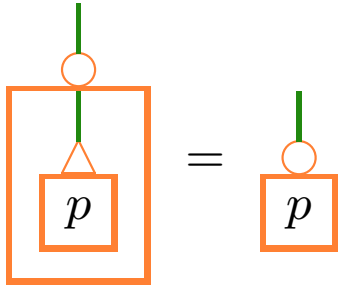
$$[x]'[y] = \left[p_1' \, p_2\right]' \left[p_3' \, p_4\right]$$

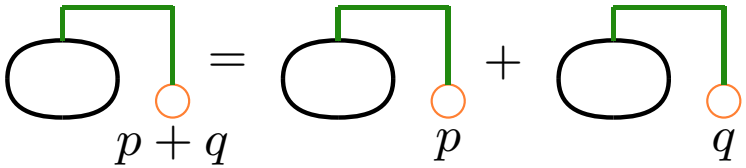
$$\begin{aligned}
xy \quad &\rightsquigarrow \quad p_1 + p_1 \, p_2 + p_1 \, p_3 + p_1 \, p_4 + p_1 \, p_2 \, p_3 + p_1 \, p_2 \, p_4 + p_1 \, p_3 \, p_4 \\
&\quad\; + p_3 + p_2 \, p_3 + p_3 \, p_4 + p_2 \, p_3 \, p_4 + p_1 \, p_2 \, p_3 \, p_4 \\
[x]'[y] \quad &\rightsquigarrow \quad p_1 + p_1 \, p_2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

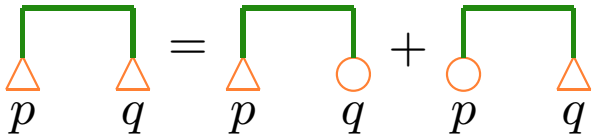
The axioms of $\mathcal{AC}(P)$: Typing

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. $[x]' \cdot [y]' = [x]' \cdot [y] + [x] \cdot [y]'$.

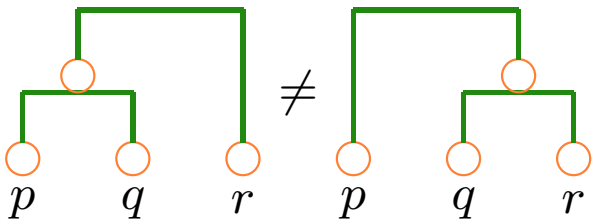






Fusion of typed connectors is not associative, e.g.

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



Equivalence of connectors

$x \simeq y \overset{def}{\Longleftrightarrow} |x| = |y|$, i.e. they represent the same sets of interactions.

- The axiomatisation of $\mathcal{AC}(P)$ is sematically sound, i.e. $x = y \Rightarrow x \simeq y$.
- Semantic equivalence is not a congruence (not preserved by fusion)

$$p + pq \simeq p'q, \quad \text{but} \quad pr + pqr \not\simeq p'qr \Big/ \simeq p + pq + pr + pqr \Big/ ,$$

$$p[qr] \simeq [pq]r, \quad \text{but} \quad s'p[qr] \not\simeq s'[pq]r\,.$$

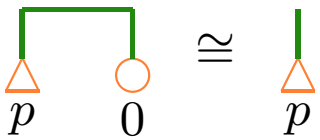
Congruence of connectors

$\cong$ 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

$$x \simeq y \Longrightarrow [x]^\alpha \cong [y]^\alpha \,.$$

- Zero is **not** absorbing: $[x]'[0] \cong [x]'$



- Congruent normal form: the union of all triggered, maximal, and potential interactions

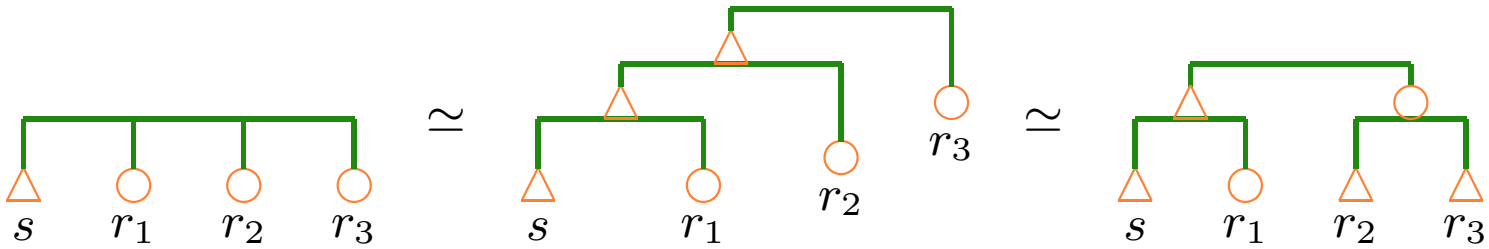
$$\begin{array}{lllll}
p'q & \cong & p' & +[pq]' & +0q & \simeq & p+pq \\
p'qr & \cong & p'r & +[pq]'r & +0qr & \simeq & p+pq+pr+pqr \\
p'qr' & \cong & p'r' & +[pq]'r' & +0qr' & \simeq & p+pq+pr+pqr+r+qr
\end{array}$$

Presentation outline

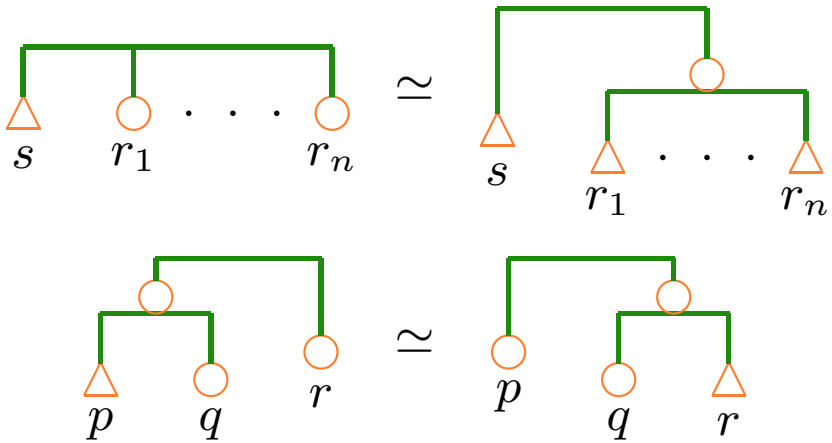
- Overview of BIP
- Interactions and Connectors
- The Algebra of Connectors
- Applications
- Conclusion

Incrementality & transformation of connectors

Incremental construction

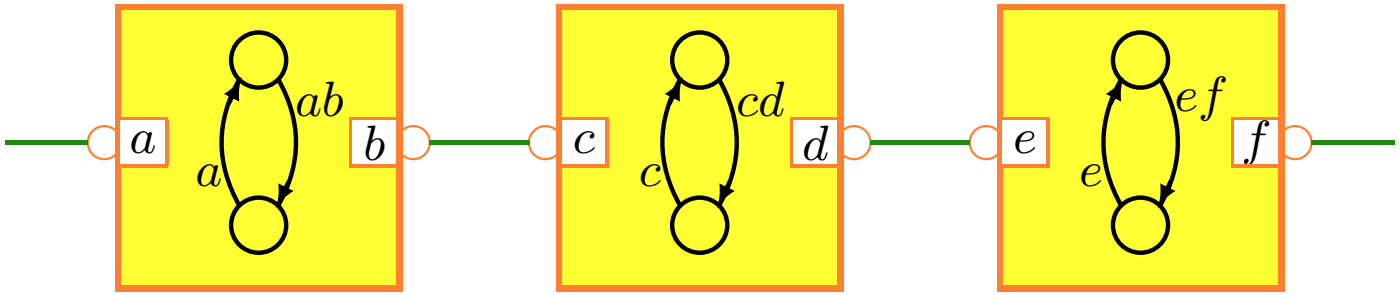


Transformation (separate one port in a connector)



Connector synthesis: Modulo-8 counter

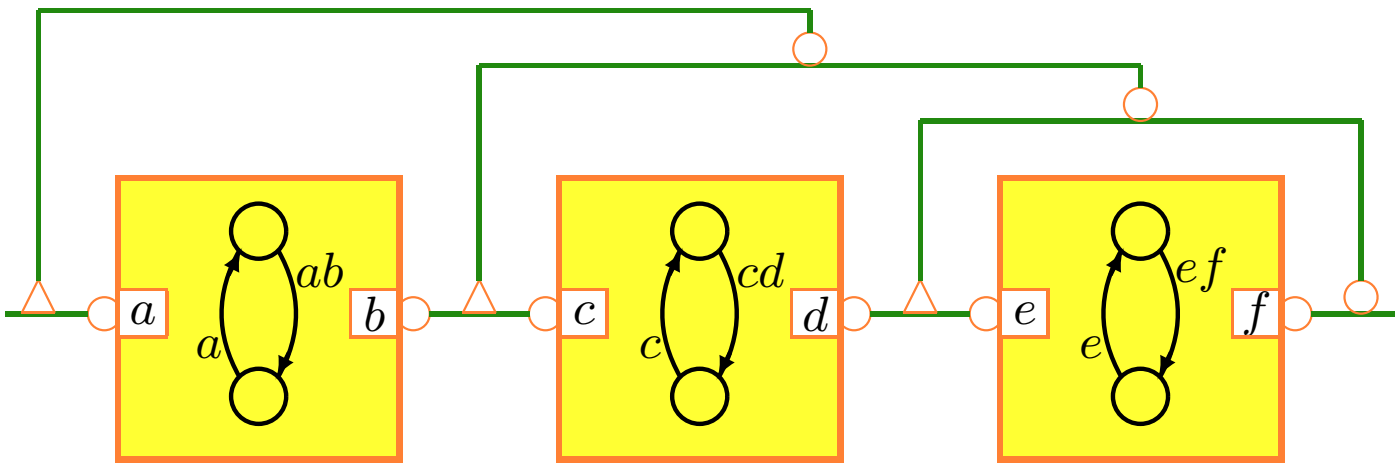
Multi-shot semantics: several connectors can be fired simultaneously



Interactions: $a + abc + abcde + abcdef$

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

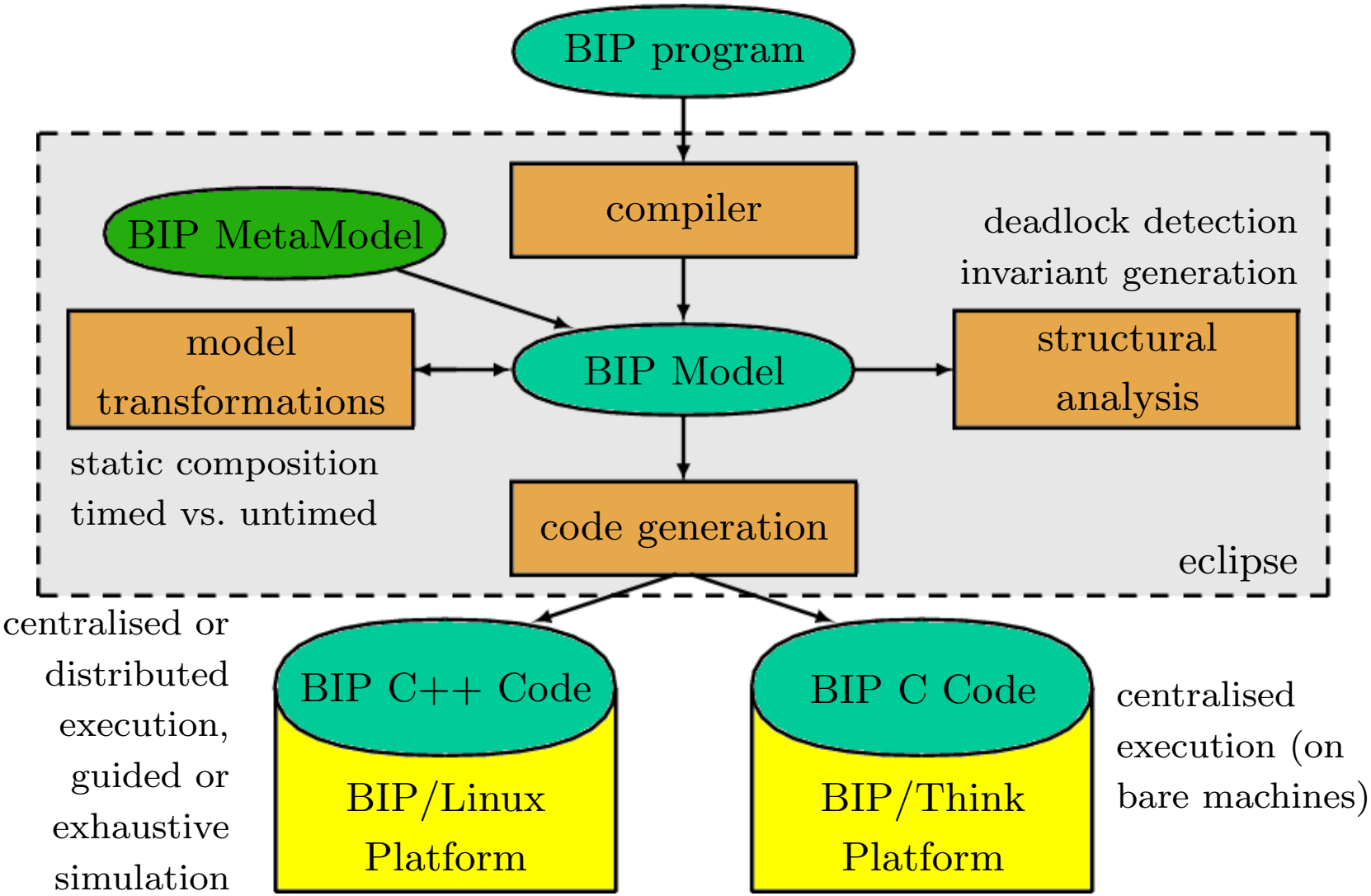
Connector synthesis: $\left[a+ab\right]'\left[c+cd\right]'\left[e+ef\right]' \cap a'[bc]'[de]'f' \simeq a'\left[[bc]'\left[[de]'f\right]\right]$.



Presentation outline

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

The BIP framework: Implementation



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.
- **Not a process algebra:** these do not study interactions as such, but only as 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.

`http://www-verimag.imag.fr/~async/BIP/bip.html`

Perspectives

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

$$
\begin{aligned}
\beta(0) &= false\,, \\
\beta(p_{i_1} \ldots p_{i_k}) &= \bigwedge_{j=1}^{k} p_{i_j} \;\wedge\; \bigwedge_{i \notin \{i_j\}} \overline{p_i}\,, \quad \text{for any } p_{i_1}, \ldots p_{i_k} \in P\,, \\
\beta(x+y) &= \beta(x) \;\vee\; \beta(y)\,, \quad \text{for any } x, y \in \mathcal{AI}(P)\,, \\
\beta(1) &= \bigwedge_{p \in P} \overline{p}\,,
\end{aligned}
$$

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

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

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

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

Annex: Characterisation of $\cong$

Th 1 *For $x, y \in \mathcal{AC}(P)$, we have*

$$x \cong y \Longleftrightarrow \begin{cases} x \simeq y \\ x \cdot 1' \simeq y \cdot 1' \\ \#x > 0 \Leftrightarrow \#y > 0\,. \end{cases}$$

$\#x$ is the number of top-level triggers in x.

Annex: Full axiomatisation of $\cong$

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

Basic:

1. $[0]' = [0]$,
2. $\Big[[x]^{\alpha}\Big]^{\beta} = [x]^{\beta}$,
3. $[x+y]^{\alpha} = [y]^{\alpha} + [x]^{\alpha}$,

Additional:

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