

The hierarchy of analytic semantics of weakly consistent parallelism

Jade Alglave (MSR-Cambridge, UCL, UK)

Patrick Cousot (NYU, Emer. ENS, PSL)

REPS AT SIXTY (co-located with SAS 2016)

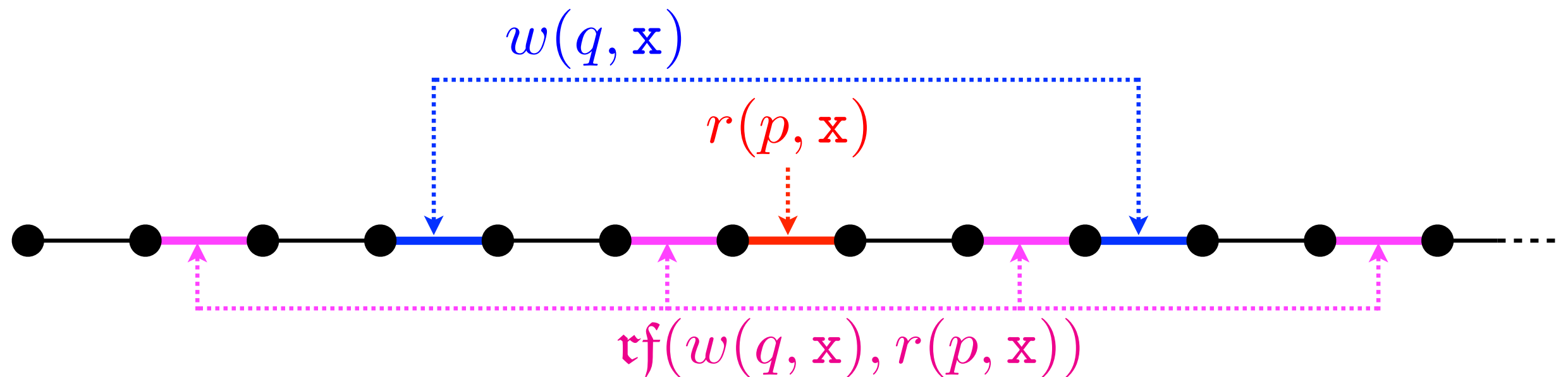
Edinburgh, Scotland

September 11, 2016

Analytic semantics

Weak consistency models (WCM)

- Sequential consistency:
reads $r(p, x)$ are *implicitly coordinated* with writes $w(q, x)$
- WCM:
No implicit coordination (depends on architecture, program dependencies, and explicit fences)



Analytic semantic specification

- *Analytic semantics = Anarchic semantics \cap Communication semantics*
- *Anarchic semantics $S^a[[P]]$:*
 - describes computations of program P, no constraints on communications
- *Communication semantics:*
 - imposes architecture-dependent communication constraints
 - e.g.: cat language (Jade Alglave & Luc Maranget)

Hierarchy of anarchic semantics

- Many different styles to describe the same computations e.g.
 - stateless/stateful
 - eager/lazy communications
 - interleaved versus true parallelism
 - ...
- They form a hierarchy of abstractions
- The communication semantics is the same for all semantics in the hierarchy

Example: load buffer (LB)

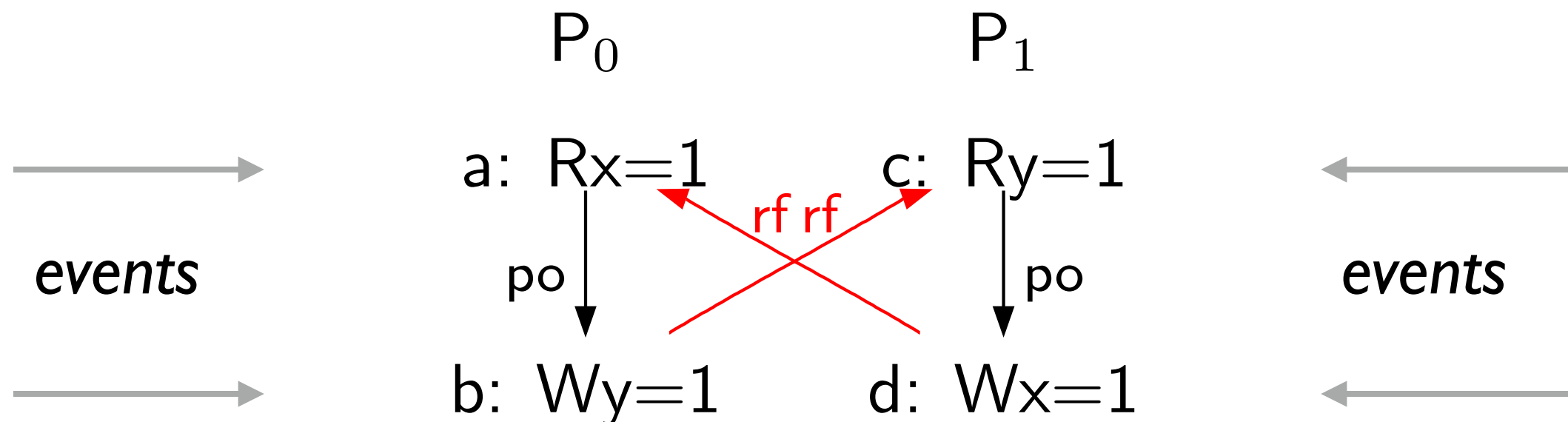
- Program:

{ x = 0; y = 0; }	
P0	P1
r[] r1 x	r[] r2 y
w[] y 1	w[] x 1
exists(0:r1=1 /\ 1:r2=1)	

- Example of execution trace $t \in S^\perp \llbracket P \rrbracket$:

$t = w(\text{start}, x, 0) \ w(\text{start}, y, 0) \ r(P0, x, 1) \ \text{rf}[w(P1, x, 1), r(P0, x, 1)] \ w(P0, y, 1) \ r(P1, y, 1) \ w(P1, x, 1) \ \text{rf}[w(P0, y, 1), r(P1, y, 1)]$

- Abstraction to cat *candidate execution* $\alpha_\Xi(t)$:




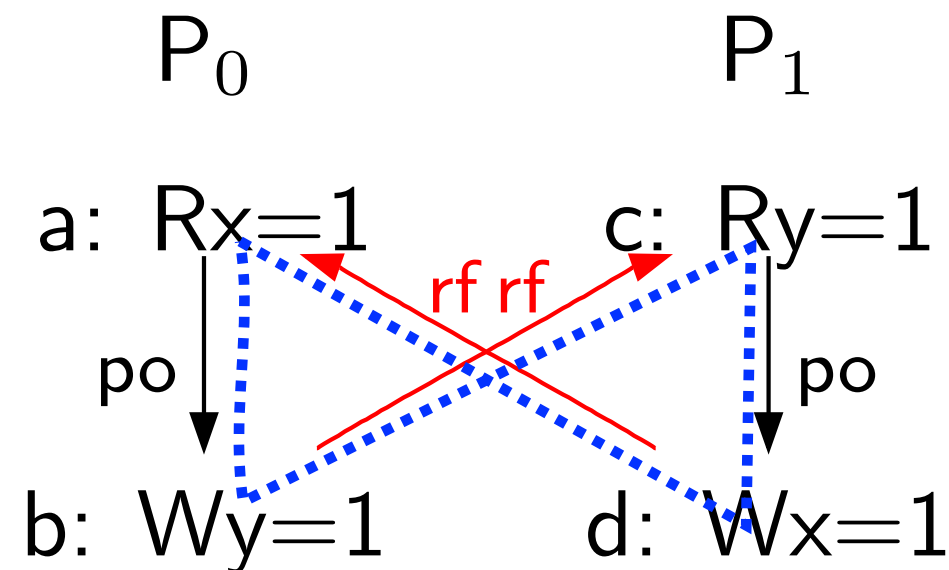
Example: load buffer (LB), cont'd

- cat specification:

$\text{acyclic (po} \mid \text{rf)}^+$

The cat semantics rejects this execution $\alpha_{\Xi}(t)$:

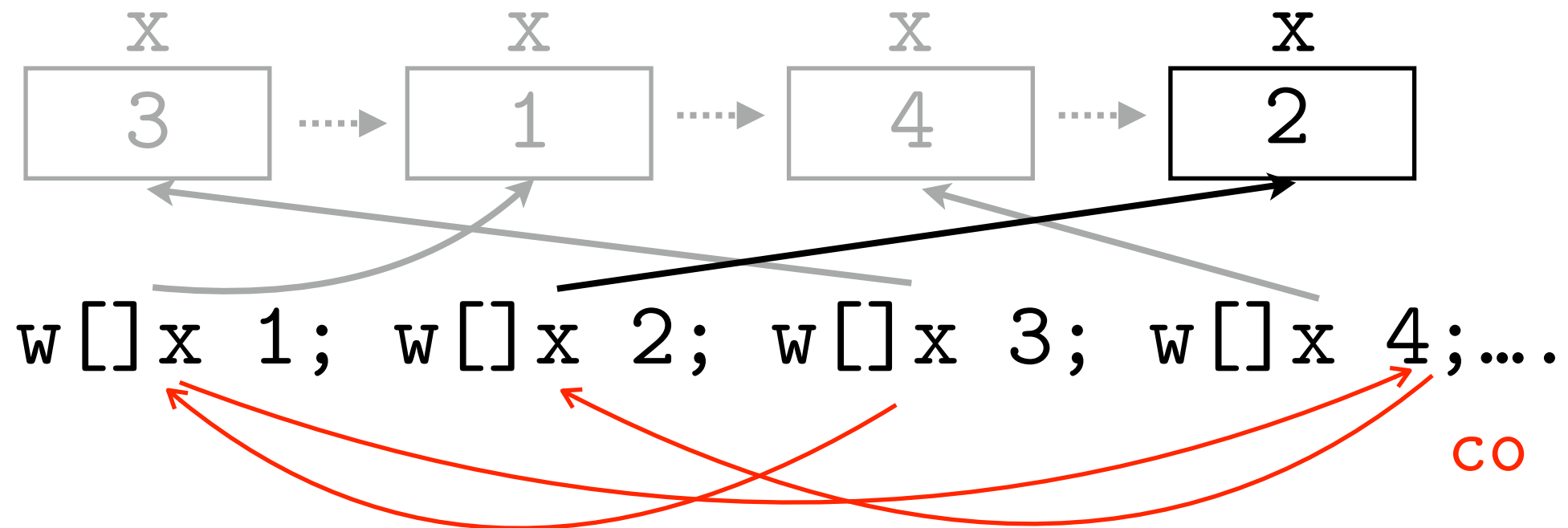
 $\llbracket \text{cat} \rrbracket (\alpha_{\Xi}(t)) = \text{forbidden}$



- The herd7 simulation tool: virginia.cs.ucl.ac.uk/herd/

Execution environment

- In general, the semantics of a parallel program depends on hypotheses on the **execution environment**
- e.g.: **coherence order**:



- the hypotheses on the execution environment (e.g. $co \subseteq po$) are part of the communication semantics

The WCM semantics

- Abstraction to a candidate execution:

$$\alpha_{\Xi}(t) \triangleq \langle \alpha_e(t), \alpha_{po}(t), \alpha_{rf}(t), \alpha_{iw}(t) \rangle$$

$$\alpha_{\Xi}(\mathcal{S}) \triangleq \{ \langle t, \alpha_{\Xi}(t) \rangle \mid t \in \mathcal{S} \}$$

- $\alpha_e(t)$ set of all events
- $\alpha_{po}(t)$ execution order of events on the same process
- $\alpha_{rf}(t)$ which read events read from which write events
- $\alpha_{iw}(t)$ initial write events (initialization before starting the parallel execution)

The WCM semantics

- The **cat** communication semantics

$$\text{cat} \llbracket \text{cat} \rrbracket (\Xi)$$

returns:

- Relations Γ on events representing hypotheses on the execution environment (e.g. co)
- allowed/forbidden depending on whether the candidate execution $\langle \Xi, \Gamma \rangle$ is consistent or not

$$\alpha_{\text{cat}} \llbracket \text{cat} \rrbracket (C) \triangleq \{t, \Gamma \mid \langle t, \Xi \rangle \in C \wedge \langle \text{allowed}, \Gamma \rangle \in \text{cat} \llbracket \text{cat} \rrbracket (\Xi)\}$$

The WCM semantics

- The **WCM** semantics:

$$S \llbracket P \rrbracket \triangleq \alpha_{\text{cat}} \llbracket \text{cat} \rrbracket \circ \alpha_{\Xi} (S^a \llbracket P \rrbracket)$$

where:

- abstraction to a candidate execution:

$$\alpha_{\Xi}(S) \triangleq \{ \langle t, \alpha_{\Xi}(t) \rangle \mid t \in S \}$$

- the cat communication semantics:

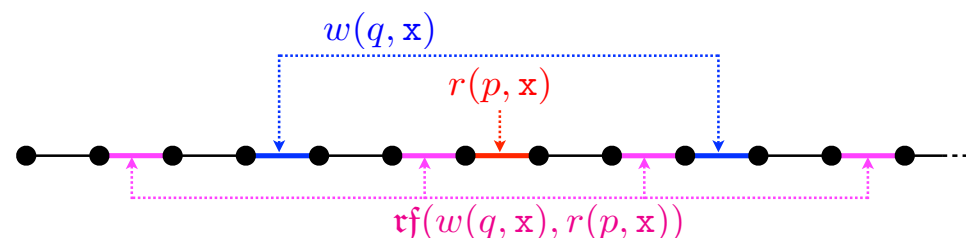
$$\alpha_{\text{cat}} \llbracket \text{cat} \rrbracket (\mathcal{C}) \triangleq \{ t, \Gamma \mid \langle t, \Xi \rangle \in \mathcal{C} \wedge \langle \text{allowed}, \Gamma \rangle \in \text{cat} \llbracket \text{cat} \rrbracket (\Xi) \}$$

- The composition of Galois connections.

Definition of the anarchic semantics

Axiomatic parameterized definition of the anarchic semantics

- The semantics $S^\perp \llbracket P \rrbracket$ is a finite/infinite **sequence of interleaved events of processes** satisfying well-formedness conditions.
- Events:
 - local computations and tests on registers, fences, rmw
 - start writing a shared variable $w(q, \mathbf{x})$
 - start reading of shared variable $r(p, \mathbf{x})$
 - communication event $\mathbf{rf}(w(q, \mathbf{x}), r(p, \mathbf{x}))$



Axiomatic parameterized definition of the anarchic semantics

- Examples of language independent well-formedness conditions of a semantics S :
 - uniqueness of events

$$\forall t \in S . \forall t_1, t_2 \in \mathcal{E}^*, t_3 \in \mathcal{E}^{*\infty} . \forall e, e' \in \mathcal{E} . (t = t_1 e t_2 e' t_3) \implies (e \neq e') . \quad (\text{Wf}_1(S))$$

- traces start with an initialization of the shared variables $(\text{Wf}_2(S))$

$$t = w(\text{start}, x, 0) w(\text{start}, y, 0) r(P0, x, 1) \text{rf}[w(P1, x, 1), r(P0, x, 1)] w(P0, y, 1) r(P1, y, 1) \\ w(P1, x, 1) \text{rf}[w(P0, y, 1), r(P1, y, 1)]$$

Axiomatic parameterized definition of the anarchic semantics

- Examples of language independent well-formedness conditions of a semantics S :
 - finite traces are **maximal**

$$\forall t \in S \cap \mathfrak{E}^+ . \nexists t' \in \mathfrak{E}^{+\infty} . t t' \in S . \quad (\text{Wf}_3(S))$$

Axiomatic parameterized definition of the anarchic semantics

- Examples of language independent well-formedness conditions of a semantics S :
 - read events must be satisfied by a unique communication event

$$\forall t \in S . \forall t_1 \in \mathfrak{E}^*, t_2 \in \mathfrak{E}^{*\infty} . (t = t_1 \text{ } r(p, \mathbf{x}) \text{ } t_2) \implies \quad (\text{Wf}_4(S))$$
$$(\exists t_3 \in \mathfrak{E}^*, t_4 \in \mathfrak{E}^{*\infty} . t = t_3 \text{ rf}[w(q, \mathbf{x}), r(p, \mathbf{x})] t_4) .$$

$$\forall t \in S . \forall t_1, t_2 \in \mathfrak{E}^*, t_3 \in \mathfrak{E}^{*\infty} . \quad (\text{Wf}_5(S))$$
$$(t \neq t_1 \text{ rf}[w(q, \mathbf{x}), r(p, \mathbf{x})] t_2 \text{ rf}[w'(q', \mathbf{x}), r(p, \mathbf{x})] t_3) .$$

Axiomatic parameterized definition of the anarchic semantics

- Examples of language independent well-formedness conditions of a semantics S :
 - communications cannot be **spontaneous** (must be originated by a read *and* a write)

$$\forall t \in S . \forall t_1 \in \mathfrak{E}^*, t_2 \in \mathfrak{E}^{*\infty} . (t = t_1 \text{ rf}[w(q, \mathbf{x}), r(p, \mathbf{x})] t_2) \implies (\exists t_3 \in \mathfrak{E}^*, t_4 \in \mathfrak{E}^{*\infty} . t = t_3 w(q, \mathbf{x}) t_4 \wedge \exists t_5 \in \mathfrak{E}^*, t_6 \in \mathfrak{E}^{*\infty} . t = t_5 r(p, \mathbf{x}) t_6) . \quad (\text{Wf}_6(S))$$

Axiomatic parameterized definition of the anarchic semantics

- The **language** :

- Programs : $\text{initialisation } \begin{matrix} \downarrow & \text{process} & \downarrow \\ [P_1 \parallel \dots \parallel P_n] \end{matrix}$

- Actions (labelled $\ell \in \mathbb{L}(p)$) :

$a ::= m$	imperative actions	marker
$r := e$		assignment
$r := x$		read of shared variable x
$x := e$		write of shared variable x
$b \mid \neg b$	conditional actions	test

- Next action : $\text{next}(p, \ell) \quad \text{nextt}(p, \ell) \quad \text{nextf}(p, \ell)$

$\uparrow \qquad \qquad \uparrow$
for tests

Axiomatic parameterized definition of the anarchic semantics

- Example of language-dependent well-formedness condition:
computation (markers: skip, fence, begin/end of rmw)

Any process p Any point k in trace Any label ℓ of p

marker event by
process p in trace τ

$$\begin{aligned} & \forall p \in \mathbb{P}i . \forall k \in [1, 1 + |\tau|[. \forall \ell \in \mathbb{L}(p) . & (\text{Wf}_{21}(\tau)) \\ & (\exists \theta \in \mathfrak{P}(p) . \bar{\tau}_k = \mathfrak{m}(\langle p, \ell, m, \theta \rangle)) \\ & \implies (\ell \in N^p(\tau, k) \wedge \text{action}(p, \ell) = m) . \end{aligned}$$

(unique) event
stamp θ

control of process
 p is at label ℓ

action of process p
is at label ℓ is the
marker action m

Axiomatic parameterized definition of the anarchic semantics

- Example of language-dependent well-formedness condition: **computation** (local variable assignment)

*register assignment event
by process p in trace τ*

(unique) event stamp θ

$$\begin{aligned} & \forall p \in \mathbb{P}i . \forall k \in]1, 1 + |\tau|[. \forall \ell \in \mathbb{L}(p) . \forall v \in \mathcal{D} . & (\text{Wf}_{22}(\tau)) \\ & (\exists \theta \in \mathfrak{P}(p) . \bar{\tau}_k = \mathfrak{a}(\langle p, \ell, \mathbf{r} := e, \theta \rangle, v)) \\ & \implies (\ell \in N^p(\tau, k) \wedge \text{action}(p, \ell) = \mathbf{r} := e \wedge v = E^p[[e]](\tau, k - 1)) . \end{aligned}$$

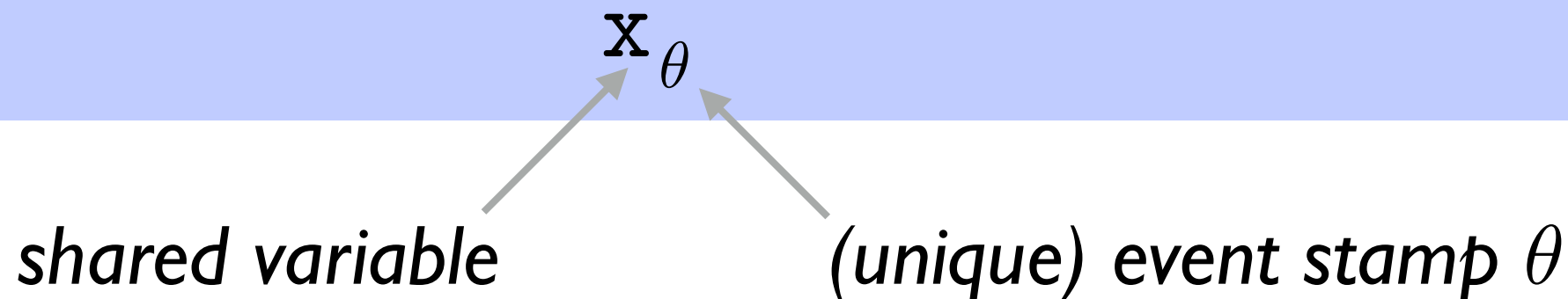
*control of process
 p is at label ℓ*

*action of process p
is at label ℓ is a
register assignment*

*value v of e is
evaluated by past-
travel*

Media variables

- With WCM there is **no notion of** “*the current value of shared variable x* ”
- At a given time each process may read a *different value* of the shared variable x (maybe guessed or unknown since a read may read from a future write)
- We use *pythia variables* (to record the values communicated between a write and read, whether the two accesses are on the same process or not)



Axiomatic parameterized definition of the anarchic semantics

- Example: communication

- a read event is initiated by a read action:

*read event by
process p in trace τ*

unique pythia variable

$$\begin{aligned} \forall p \in \mathbb{P}i . \forall k \in]1, 1 + |\tau|| . \forall \ell \in \mathbb{L}(p) . & \quad (Wf_{23}(\tau)) \\ (\exists \theta \in \mathfrak{P}(p) . (\bar{\tau}_k = \mathbf{r}(\langle p, \ell, \mathbf{r} := \mathbf{x}, \theta \rangle, \mathbf{x}_\theta))) & \\ \implies (\ell \in N^p(\tau, k) \wedge \text{action}(p, \ell) = \mathbf{r} := \mathbf{x}) . & \end{aligned}$$

- a read must read-from (rf) a write (weak fairness):

$$\begin{aligned} \forall p \in \mathbb{P}i . \forall i \in]1, 1 + |\tau|[. \forall r \in \mathfrak{Rf}(p) . & \quad (Wf_{26}(\tau)) \\ (\bar{\tau}_i = r) \implies (\exists j \in]1, 1 + |\tau|[. \exists w \in \mathfrak{W}i . \bar{\tau}_j = \mathbf{rf}[w, r]) . & \end{aligned}$$

communication (read-from) event

Axiomatic parameterized definition of the anarchic semantics

- Predictive evaluation of pythia variables:

$$V_{(32)}^p \llbracket \mathbf{x}_\theta \rrbracket (\tau, k) \triangleq v \text{ where } \exists ! i \in [1, 1 + |\tau|[\cdot (\bar{\tau}_i = \mathbf{r}(\langle p, \ell, \mathbf{r} := \mathbf{x}, \theta \rangle, \mathbf{x}_\theta)) \wedge \\ \exists ! j \in [1, 1 + |\tau|[\cdot (\bar{\tau}_j = \mathbf{rf}[\mathbf{w}(\langle p', \ell', \mathbf{x} := e', \theta' \rangle, v), \bar{\tau}_i])$$

- Local past-travel evaluation of an expression:

$$E_{(30)}^p \llbracket \mathbf{r} \rrbracket (\tau, k) \triangleq v \text{ if } k > 1 \wedge ((\bar{\tau}_k = \mathbf{a}(\langle p, \ell, \mathbf{r} := e, \theta \rangle, v)) \vee \\ (\bar{\tau}_k = \mathbf{r}(\langle p, \ell, \mathbf{r} := \mathbf{x}, \theta \rangle, \mathbf{x}_\theta) \wedge V^p \llbracket \mathbf{x}_\theta \rrbracket (\tau, k) = v))$$

$$E_{(30)}^p \llbracket \mathbf{r} \rrbracket (\tau, 1) \triangleq I \llbracket 0 \rrbracket \quad \text{i.e. } \bar{\tau}_1 = \epsilon_{\text{start}} \text{ by } \text{Wf}_{15}(\tau)$$

$$E_{(30)}^p \llbracket \mathbf{r} \rrbracket (\tau, k) \triangleq E_{(30)}^p \llbracket \mathbf{r} \rrbracket (\tau, k - 1) \quad \text{otherwise.}$$

Abstractions of the anarchic semantics

Abstractions

- Anarchic semantics:

$$S^{\perp}[[P]] \triangleq \lambda \langle \mathcal{B}, \text{sat}, \mathcal{D}, I, \mathcal{G}, V, E, N \rangle \bullet \{ \tau \in \mathcal{T}[[P]] |_{\cong} \mid \text{Wf}_1(\tau) \wedge \dots \wedge \text{Wf}_{29}(\tau) \}$$

parameters of the semantics

trace well-formedness conditions

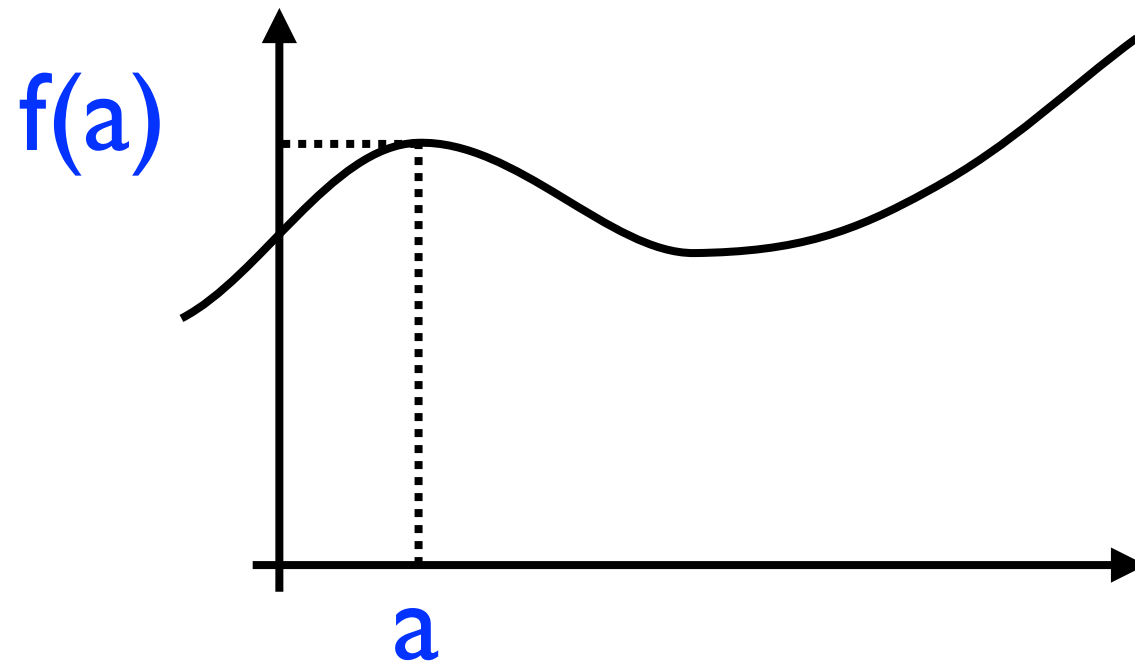
- Examples of **abstractions**:

- Choose data (e.g. ground values, uninterpreted symbolic expressions, interpreted symbolic expressions i.e. “symbolic guess”)
- Bind parameters (e.g. how expressions are evaluated)
- ...

Binding a parameter of the semantics

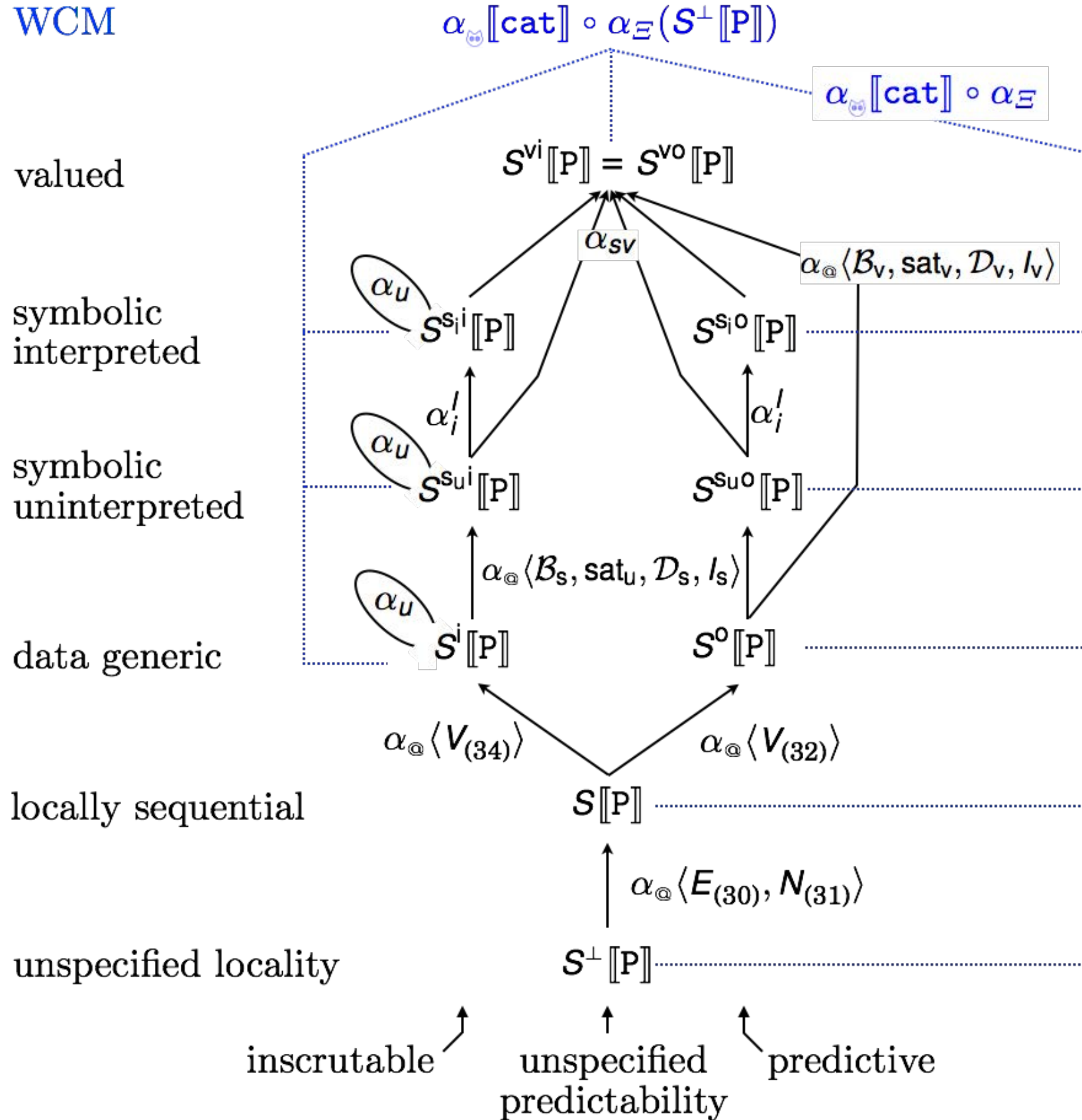
- The abstraction

$$\alpha_a(f) \stackrel{\text{def}}{=} f(a)$$



$$\langle \wp(A, B, \dots) \longrightarrow \wp(R), \dot{\subseteq} \rangle \begin{array}{c} \xleftarrow{\gamma_a} \\ \xrightarrow{\alpha_a} \end{array} \langle \wp(B, \dots) \longrightarrow \wp(R), \dot{\subseteq} \rangle$$

The hierarchy of interleaved semantics

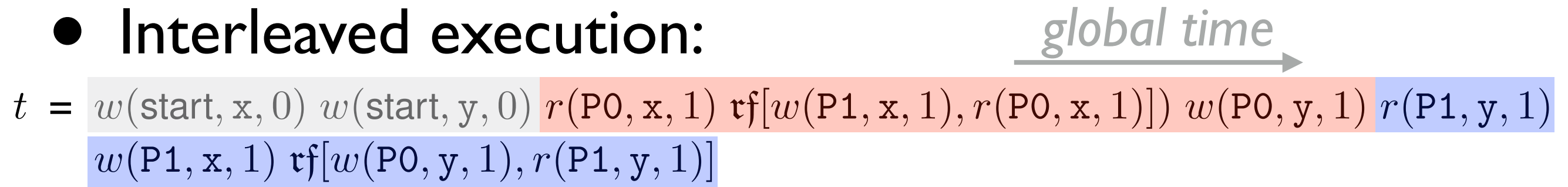


True parallelism with local communications

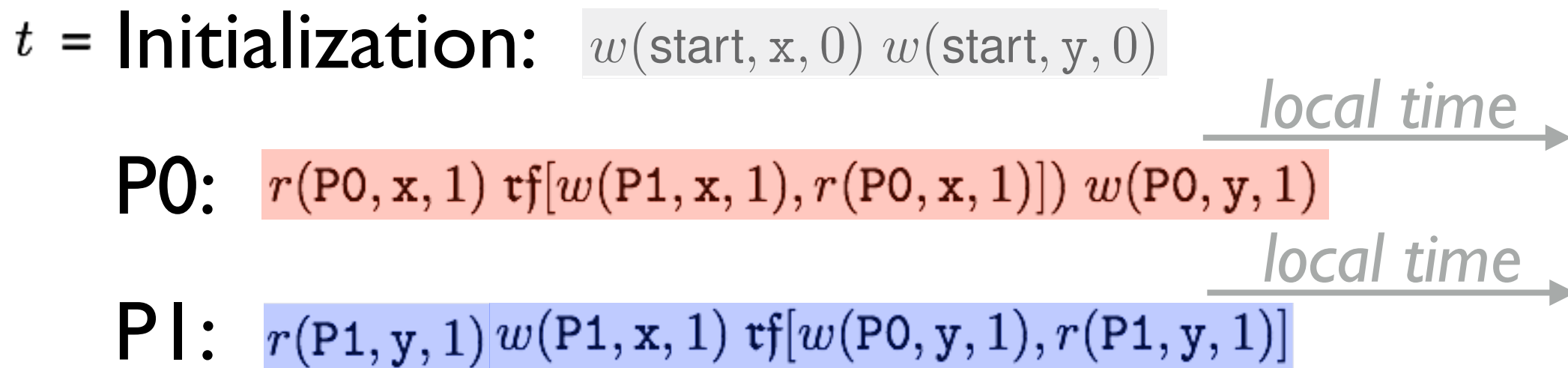
- Extract from interleaved executions:
 - The **subtrace of each process** keeping **communications** in the process that read
- ⇒ **no** more **global time** between processes
- ⇒ **local time** between local actions and communications (a read can still tell when it is satisfied by which write)

True parallelism with local communications

- Interleaved execution:



- Parallel executions with interleaved communications:



True parallelism of computations and communications

- Extract from interleaved executions:
 - The **subtrace of each process** (sequential execution of actions)
 - The **rf communication relation** (interactions between processes)
- ⇒ **no more global time** between processes
- ⇒ **no more global/local time** for communications

True parallelism with separate communications

- Parallel executions with interleaved communications:

Initialization: $w(\text{start}, x, 0) \ w(\text{start}, y, 0)$

P0: $r(P0, x, 1) \ w(P0, y, 1)$ *local time* →

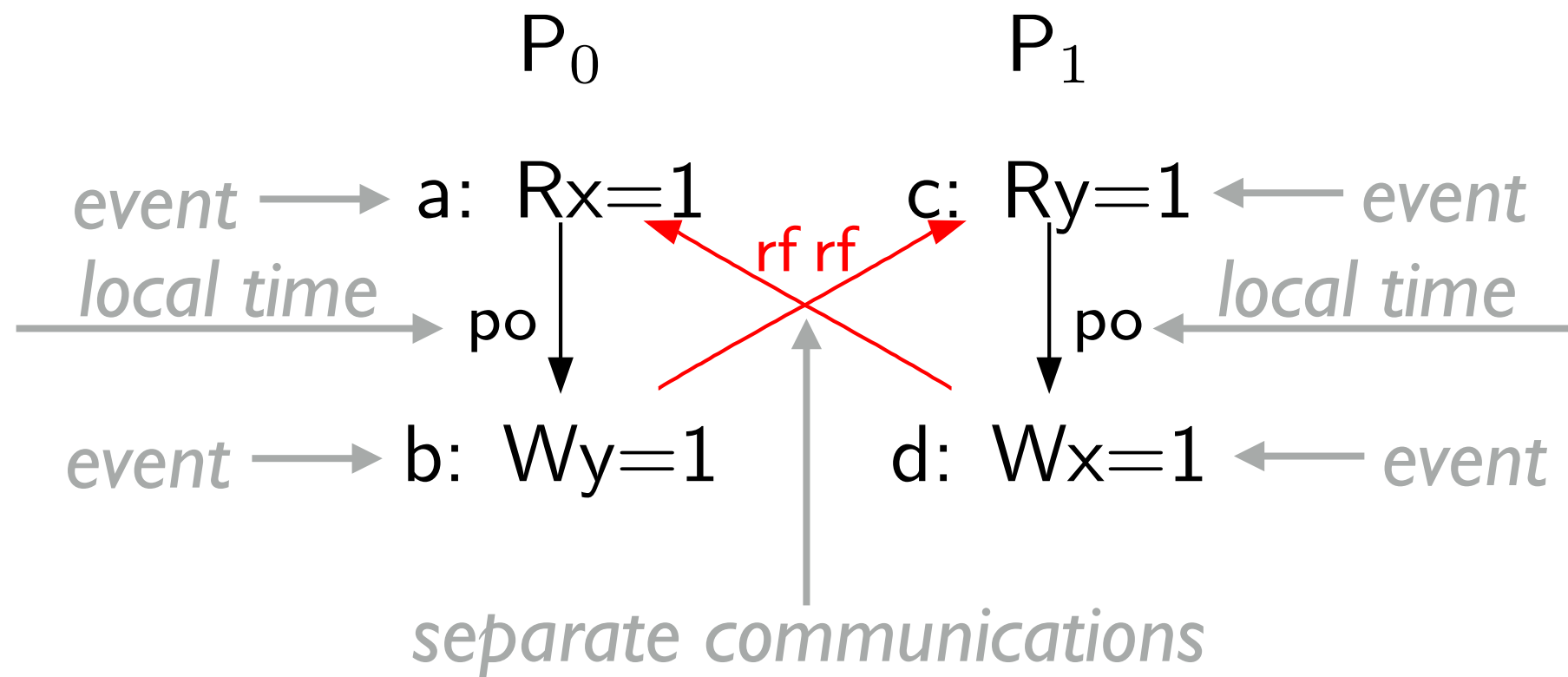
P1: $r(P1, y, 1) \ w(P1, x, 1)$ *local time* →

Communications:

$\{ \text{rf}[w(P1, x, 1), r(P0, x, 1)] , \text{rf}[w(P0, y, 1), r(P1, y, 1)] \}$

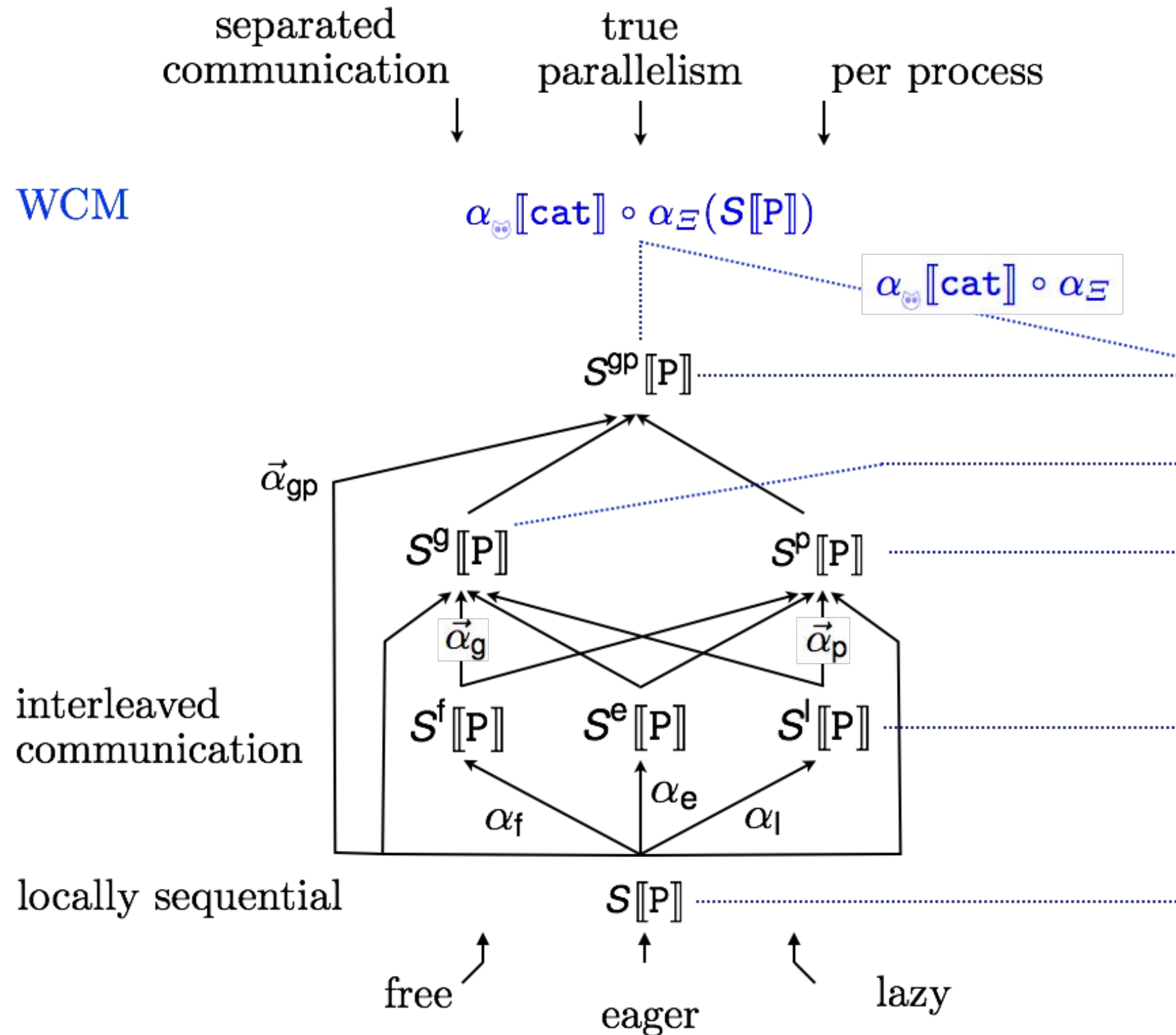
True parallelism with separate communications

- This is the semantics used by the herd7 tool:



+ interpreted symbolic expressions i.e. “symbolic guess”

The true parallelism hierarchy



States

- At each point in a trace, the state abstracts the past computation history up to that point
- Example: classical environment (assigning values to register at each point k of the trace):

$$\rho^p(\tau, k) \triangleq \lambda \mathbf{r} \in \mathbb{R}(p) \bullet E^p[\mathbf{r}](\tau, k)$$

$$\nu^p(\tau, k) \triangleq \lambda \mathbf{x}_\theta \bullet V_{(32)}^p[\mathbf{x}_\theta](\tau, k)$$

Prefixes, transitions, ...

- Abstract traces by their prefixes:

$$\overleftarrow{\alpha}(S) \triangleq \bigcup \{ \overleftarrow{\alpha}(\tau) \mid \tau \in S \}$$

$$\overleftarrow{\alpha}(\tau) \triangleq \{ \tau \llbracket j \rrbracket \mid j \in [1, 1 + |\tau|] \}$$

$$\tau \llbracket j \rrbracket \triangleq \langle \xrightarrow{\overline{\tau}_i} \underline{\tau}_i \mid i \in [1, 1 + j[\rangle$$

- and transitions: extract transitions from traces

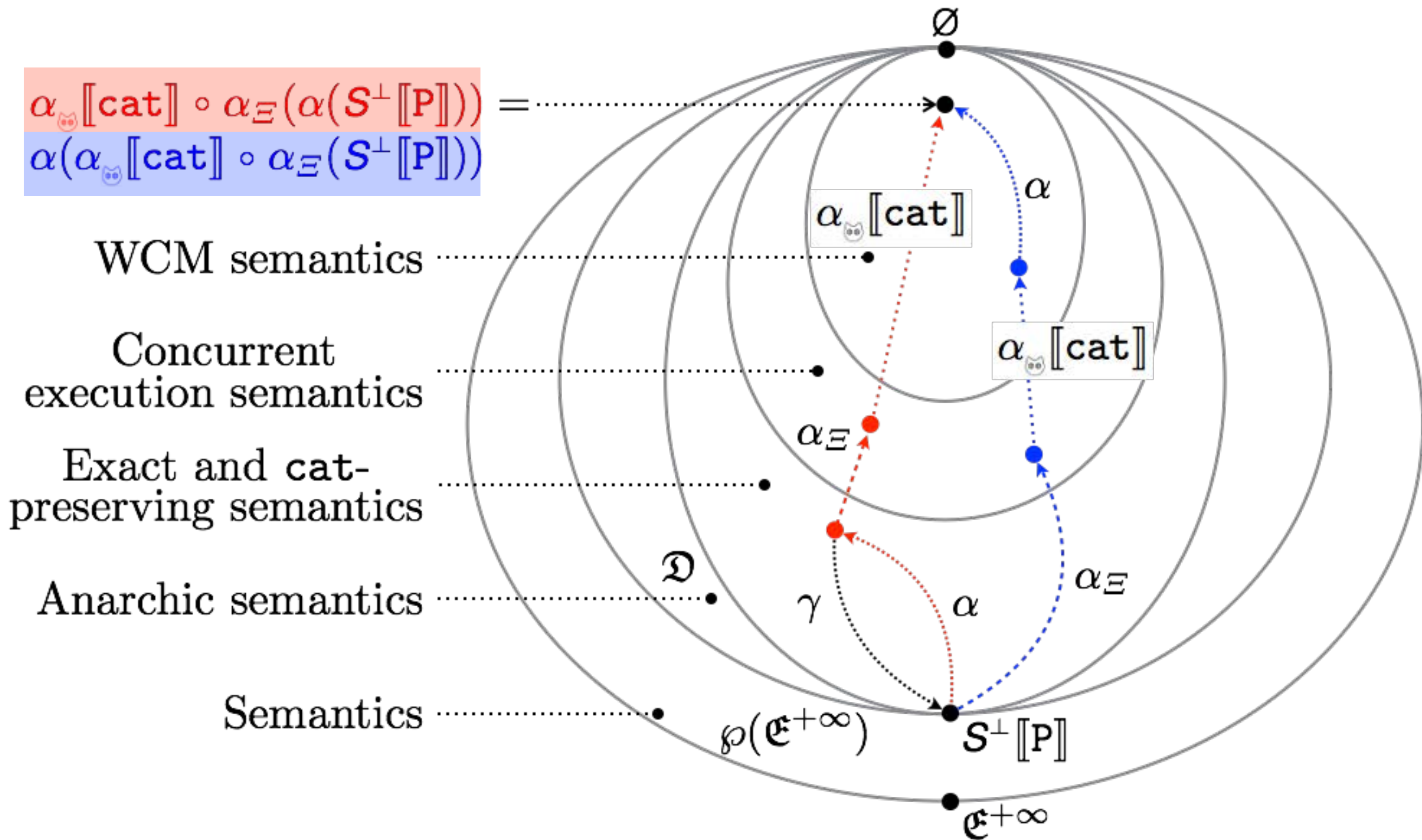
⇒ communication fairness is lost, inexact abstraction,

⇒ add fairness condition

⇒ impossible to implement with a scheduler (\neq process fairness)

Effect of the cat specification on the hierarchy

Exactness and cat preservation



The cat abstraction

- The same cat specification α_{cat} applies equally to any concurrent execution abstraction of any interleaved/truly parallel semantics in the hierarchy α_{Ξ}
- The appropriate level of abstraction to specify WCM:
 - No states, only marker (e.g. fence), r, w, rf(w,r) events
 - No values in events
 - No global time (only po order of events per process)
 - Time of communications forgotten (only rf of who communicates with whom)
 - Hypotheses on the execution environment independent of computed and communicated values

Conclusion

Conclusion

- **Analytic semantics**: a new style of semantics
- The hierarchy of **anarchic semantics** describes the same computations and potential communications in very different styles
- The **cat semantics** restricts communications to a machine/network architecture in the same way for all semantics in the hierarchy
- This idea of **parameterized semantics at various levels of abstraction** is useful for
 - **Verification**
 - **Static analysis**

The End