The hierarchy of analytic semantics of weakly consistent parallelism

Jade Alglave (MSR-Cambridge, UCL, UK)
Patrick Cousot (NYU, Emer. ENS, PSL)

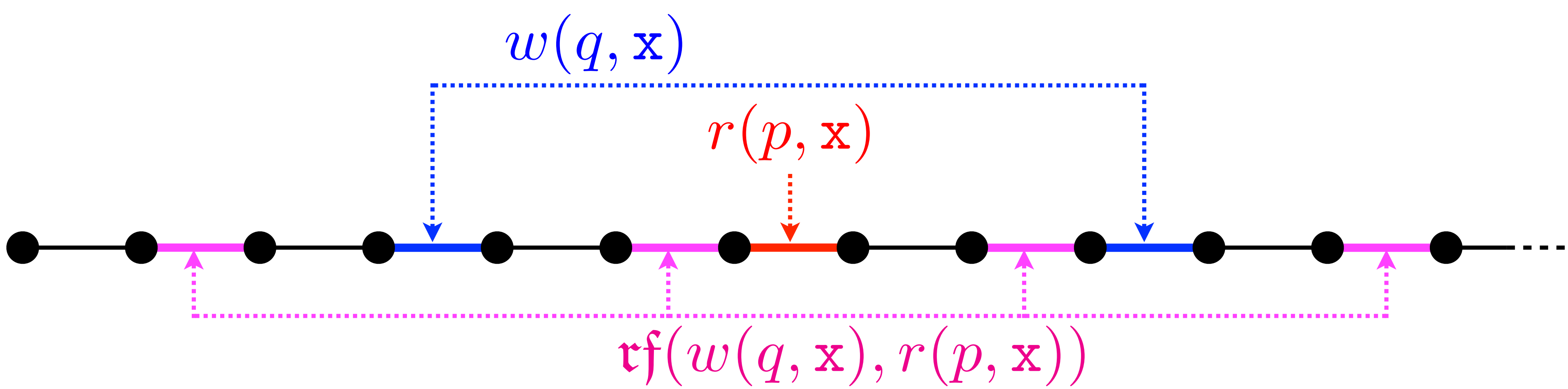
IMDEA seminar
Madrid
Tuesday, May 24th, 2016 — 11:00 AM

Analytic semantics

Weak consistency models (WCM)

- Sequential consistency:
  reads $r(p, \mathtt{x})$ are *implicitly coordinated* with writes $w(q, \mathtt{x})$

- WCM:
  No implicit coordination (depends on architecture, program dependencies, and explicit fences)



Analytic semantic specification

- Anarchic semantics:

  describes computations, no constraints on communications

- cat specification (Jade Alglave & Luc Maranget):

  imposes architecture-dependent communication constraints

- Hierarchy of anarchic semantics:

  many different styles to describe the same computations (e.g. stateless/stateful, interleaved versus true parallelism)

Example: load buffer (LB)

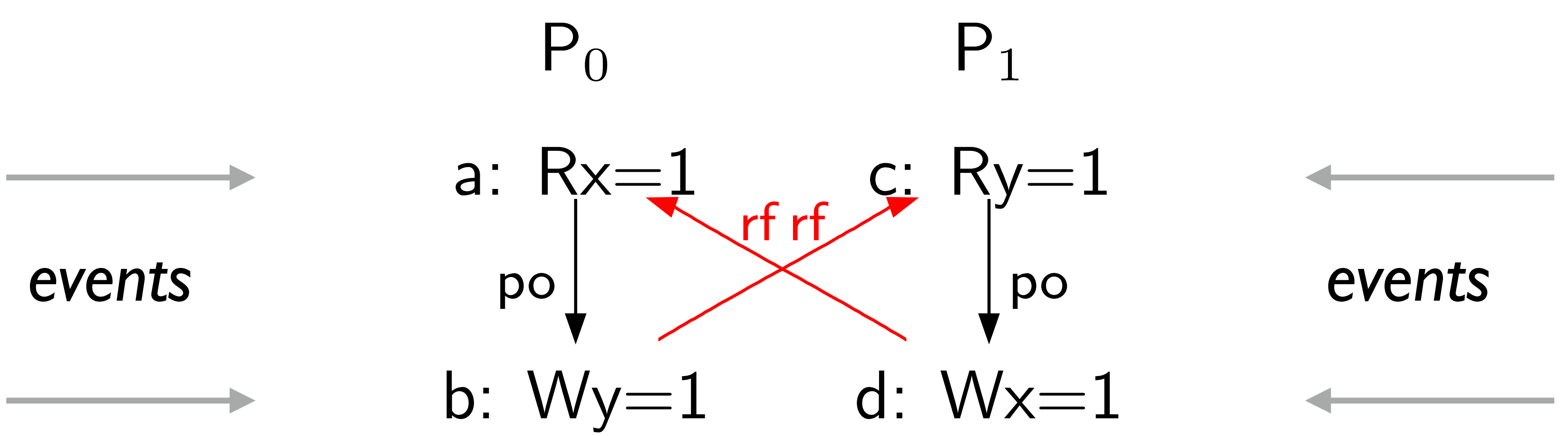
- Program:

```
{ 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 $t \in \mathcal{S}^{\perp}[\![\mathrm{P}]\!]$:

$$t \;=\; w(\mathsf{start}, \mathrm{x}, 0)\ w(\mathsf{start}, \mathrm{y}, 0)\ r(\mathtt{P0}, \mathrm{x}, 1)\ \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\mathtt{P0}, \mathrm{x}, 1)])\ w(\mathtt{P0}, \mathrm{y}, 1)\ r(\mathtt{P1}, \mathrm{y}, 1)\ w(\mathtt{P1}, \mathrm{x}, 1)\ \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\mathtt{P1}, \mathrm{y}, 1)]\ r(\mathsf{finish}, \mathrm{x})\ \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\mathsf{finish}, \mathrm{x}, 1)]\ r(\mathsf{finish}, \mathrm{y}, 1)\ \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\mathsf{finish}, \mathrm{y}, 1)]$$

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



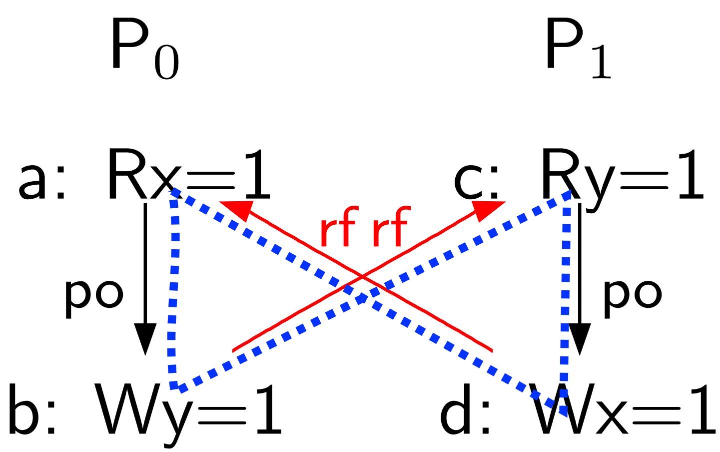
Example: load buffer (LB), cont'd

- cat specification:

```
acyclic (po | rf)+
```

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

$$[\![\texttt{cat}]\!]\,(\alpha_{\Xi}(t)) = \textsf{false}$$



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

The WCM semantics

- Abstraction to a candidate execution:

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

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

- The cat semantics:

$$\alpha_{\text{🐱}}[\![\mathtt{cat}]\!](\mathcal{S}) \triangleq \{t \mid \langle t,\ \Xi\rangle \in \mathcal{S} \wedge \text{🐱}[\![\mathtt{cat}]\!](\Xi)\}$$

- The WCM semantics:

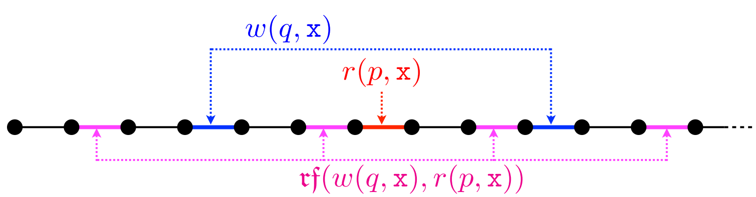
$$\alpha_{\text{🐱}}[\![\mathtt{cat}]\!] \circ \alpha_{\Xi}(\mathcal{S}[\![\mathrm{P}]\!])$$

GC: $\langle \wp(\mathfrak{E}^{+\infty}),\ \subseteq\rangle \underset{\alpha_{\Xi}}{\overset{\gamma_{\Xi}}{\leftrightarrows}} \langle \wp(\mathfrak{E}^{+\infty} \times \Xi),\ \subseteq\rangle \underset{\alpha_{\text{🐱}}[\![\mathtt{cat}]\!]}{\overset{\gamma_{\text{🐱}}[\![\mathtt{cat}]\!]}{\leftrightarrows}} \langle \wp(\mathfrak{E}^{+\infty}),\ \subseteq\rangle$

Definition of the anarchic semantics

Axiomatic parameterized definition of the anarchic semantics

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



Axiomatic parameterized definition of the anarchic semantics

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

$$\forall t \in S \,.\, \forall t_1, t_2 \in \mathfrak{E}^*, t_3 \in \mathfrak{E}^{*\infty} \,.\, \forall e, e' \in \mathfrak{E} \,.\, (t = t_1\, e\, t_2\, e'\, t_3) \Longrightarrow (e \neq e')\,. \qquad (\mathsf{Wf}_1(S))$$

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

$$t \;=\; w(\mathsf{start}, \mathrm{x}, 0)\; w(\mathsf{start}, \mathrm{y}, 0)\; r(\mathtt{P0}, \mathrm{x}, 1)\; \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\mathtt{P0}, \mathrm{x}, 1)])\; w(\mathtt{P0}, \mathrm{y}, 1)\; r(\mathtt{P1}, \mathrm{y}, 1)$$
$$w(\mathtt{P1}, \mathrm{x}, 1)\; \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\mathtt{P1}, \mathrm{y}, 1)]\; r(\mathsf{finish}, \mathrm{x})\; \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\mathsf{finish}, \mathrm{x}, 1)]$$
$$r(\mathsf{finish}, \mathrm{y}, 1)\; \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\mathsf{finish}, \mathrm{y}, 1)]$$

Axiomatic parameterized definition of the anarchic semantics

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

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

  - the final value of shared variables in finite traces is known thanks to a final read $(\mathsf{Wf}_4(S))$

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

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

$$\forall t \in S \;.\; \forall t_1 \in \mathfrak{E}^*, t_2 \in \mathfrak{E}^{*\infty} \;.\; (t = t_1\, r(p, \mathrm{x})\, t_2) \Longrightarrow \qquad (\mathrm{Wf}_5(S))$$
$$(\exists t_3 \in \mathfrak{E}^*, t_4 \in \mathfrak{E}^{*\infty} \;.\; t = t_3\, \mathfrak{rf}[w(q, \mathrm{x}), r(p, \mathrm{x})]\, t_4) \;.$$

$$\forall t \in S \;.\; \forall t_1, t_2 \in \mathfrak{E}^*, t_3 \in \mathfrak{E}^{*\infty} \;. \qquad (\mathrm{Wf}_6(S))$$
$$(t \neq t_1\, \mathfrak{rf}[w(q, \mathrm{x}), r(p, \mathrm{x})]\, t_2\, \mathfrak{rf}[w'(q', \mathrm{x}), r(p, \mathrm{x})]\, t_3) \;.$$

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)

$$\forall t \in S \,.\, \forall t_1 \in \mathfrak{E}^*, t_2 \in \mathfrak{E}^{*\infty} \,.\, (t = t_1 \,\mathfrak{rf}[w(q,\mathtt{x}), r(p,\mathtt{x})]\, t_2) \Longrightarrow \qquad (\mathsf{Wf}_7(S))$$
$$(\exists t_3 \in \mathfrak{E}^*, t_4 \in \mathfrak{E}^{*\infty} \,.\, t = t_3 w(q,\mathtt{x}) t_4 \wedge \exists t_5 \in \mathfrak{E}^*, t_6 \in \mathfrak{E}^{*\infty} \,.\, t = t_5 r(p,\mathtt{x}) t_6) \,.$$

Axiomatic parameterized definition of the anarchic semantics

- The language :
  - Programs : *process* — *initialisation* $[\![\mathrm{P}_1 \| \ldots \| \mathrm{P}_n]\!]$ *finalisation*
  - Actions (labelled $\ell \in \mathbb{L}(p)$) :

$$
\begin{array}{rlll}
a ::= & m & \text{imperative actions} & \text{marker} \\
\mid & \mathtt{r} \texttt{ := } e & & \text{assignment} \\
\mid & \mathtt{r} \texttt{ := } \mathtt{x} & & \text{read of shared variable } \mathtt{x} \\
\mid & \mathtt{x} \texttt{ := } e & & \text{write of shared variable } \mathtt{x} \\
\mid & b \mid \neg b & \text{conditional actions} & \text{test}
\end{array}
$$

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

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

$$\forall p \in \mathbb{P}\mathbb{i} \;.\; \forall k \in [1, 1 + |\tau|[\;.\; \forall \ell \in \mathbb{L}(p) \;. \qquad (\mathsf{Wf}_{21}(\tau))$$
$$(\exists \theta \in \mathfrak{P}(p) \;.\; \overline{\tau}_k = \mathfrak{m}(\langle p,\ \ell,\ m,\ \theta\rangle))$$
$$\Longrightarrow (\ell \in \mathsf{N}^p(\tau, k) \wedge \mathsf{action}(p, \ell) = m) \;.$$

(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

$$\forall p \in \mathbb{P}\mathbb{i} \,.\, \forall k \in]1, 1 + |\tau|[\,.\, \forall \ell \in \mathbb{L}(p) \,.\, \forall v \in \mathcal{D} \,. \qquad (\mathsf{Wf}_{22}(\tau))$$
$$(\exists \theta \in \mathfrak{P}(p) \,.\, \overline{\tau}_k = \mathfrak{a}(\langle p,\ \ell,\ \mathtt{r} := e,\ \theta\rangle, v))$$
$$\Longrightarrow (\ell \in \mathsf{N}^p(\tau, k) \wedge \mathsf{action}(p, \ell) = \mathtt{r} := e \wedge v = \mathsf{E}^p[\![e]\!](\tau, k-1)) \,.$$

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 *media variables* (to record the values communicated between a write and read, whether the two accesses are on the same process or not)

$$\mathtt{x}_{\theta}$$

shared variable *(unique) event stamp θ*

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 media variable

$$\forall p \in \mathbb{P}\mathfrak{i} \,.\, \forall k \in \,]1, 1+|\tau|[\,.\, \forall \ell \in \mathbb{L}(p) \,. \qquad (\mathsf{Wf}_{23}(\tau))$$
$$(\exists \theta \in \mathfrak{P}(p) \,.\, (\overline{\tau}_k = \mathfrak{r}(\langle p,\ \ell,\ \mathtt{r} := \mathtt{x},\ \theta\rangle, \mathtt{x}_\theta)))$$
$$\Longrightarrow (\ell \in N^p(\tau, k) \wedge \mathsf{action}(p, \ell) = \mathtt{r} := \mathtt{x}) \,.$$

  - a read must read-from ($\mathfrak{rf}$) a write (weak fairness):

$$\forall p \in \mathbb{P}\mathfrak{i} \,.\, \forall i \in \,]1, 1+|\tau|[\,.\, \forall r \in \mathfrak{Rf}(p) \,. \qquad (\mathsf{Wf}_{26}(\tau))$$
$$(\overline{\tau}_i = r) \Longrightarrow (\exists j \in \,]1, 1+|\tau|[\,.\, \exists w \in \mathfrak{Wi} \,.\, \overline{\tau}_j = \mathfrak{rf}[w, r]) \,.$$

communication (read-from) event

Axiomatic parameterized definition of the anarchic semantics

- Predictive evaluation of media variables:

$$V^{p}_{(32)}[\![\mathtt{x}_\theta]\!](\tau, k) \triangleq v \;\; \textbf{where} \;\; \exists! i \in [1, 1+|\tau|[\,.\,(\overline{\tau}_i = \mathfrak{r}(\langle p,\ \ell,\ \mathtt{r}\ \mathtt{:=}\ \mathtt{x},\ \theta\rangle, \mathtt{x}_\theta)) \wedge \exists! j \in [1, 1+|\tau|[\,.\,(\overline{\tau}_j = \mathfrak{rf}[\mathfrak{w}(\langle p',\ \ell',\ \mathtt{x}\ \mathtt{:=}\ e',\ \theta'\rangle, v), \overline{\tau}_i])$$

- Local past-travel evaluation of an expression:

$$E^{p}_{(30)}[\![\mathtt{r}]\!](\tau, k) \triangleq v \quad \text{if } k > 1 \wedge \big((\overline{\tau}_k = \mathfrak{a}(\langle p,\ \ell,\ \mathtt{r}\ \mathtt{:=}\ e,\ \theta\rangle, v)) \vee (\overline{\tau}_k = \mathfrak{r}(\langle p,\ \ell,\ \mathtt{r}\ \mathtt{:=}\ \mathtt{x},\ \theta\rangle, \mathtt{x}_\theta) \wedge V^{p}[\![\mathtt{x}_\theta]\!](\tau, k) = v)\big)$$

$$E^{p}_{(30)}[\![\mathtt{r}]\!](\tau, 1) \triangleq I[\![0]\!] \qquad i.e.\ \overline{\tau}_1 = \epsilon_{\mathsf{start}} \text{ by } \mathsf{Wf}_{15}(\tau)$$

$$E^{p}_{(30)}[\![\mathtt{r}]\!](\tau, k) \triangleq E^{p}_{(30)}[\![\mathtt{r}]\!](\tau, k-1) \qquad \text{otherwise.}$$

Abstractions of the anarchic semantics

Abstractions

- Anarchic semantics:

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

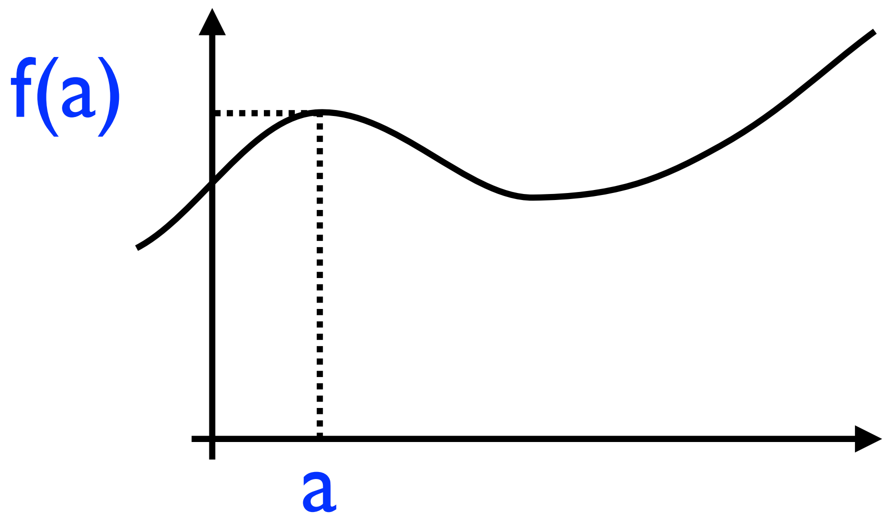
parameters of the semantics   trace well-formedness conditions

- 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

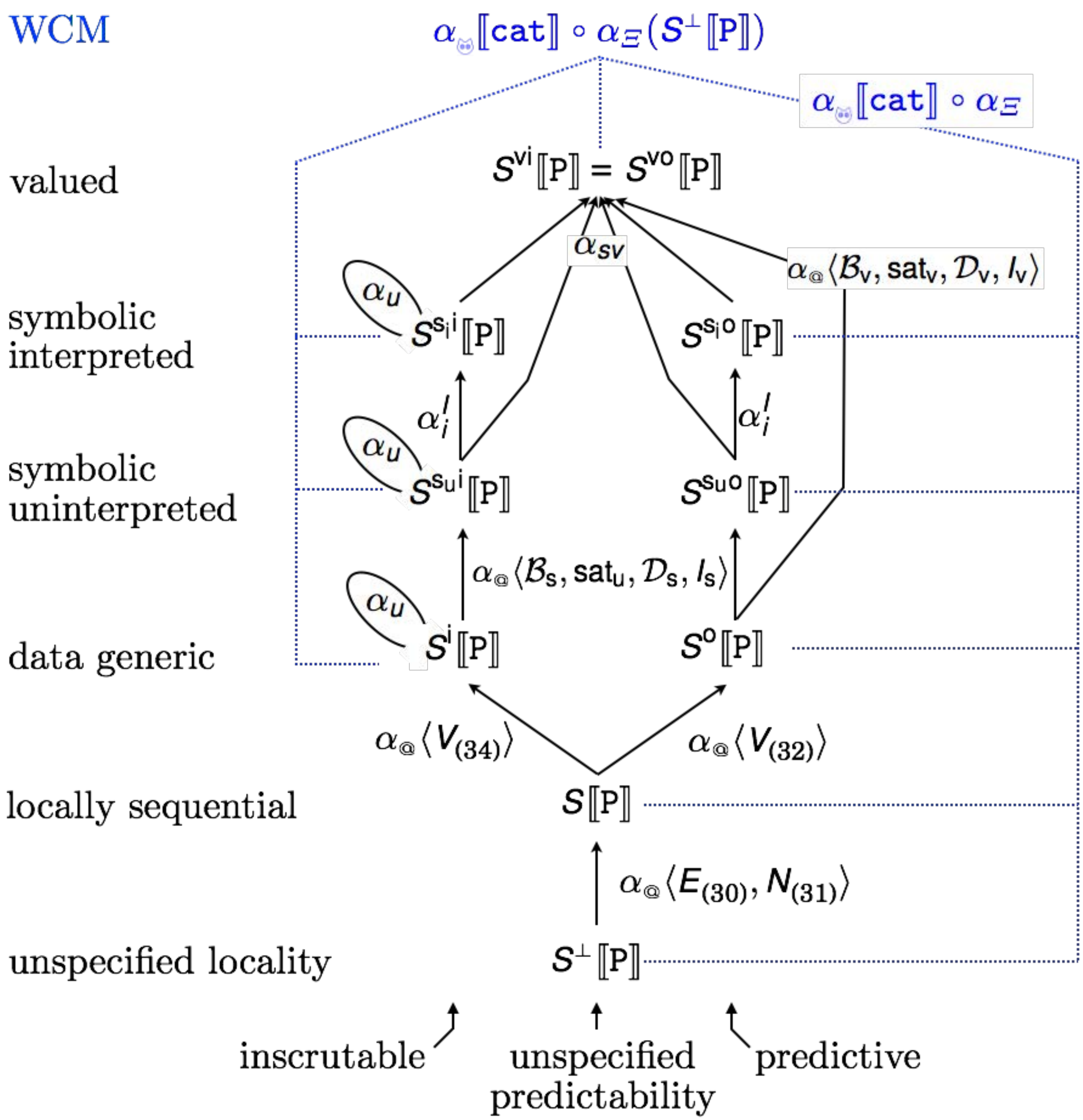
- The abstraction

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



$$\langle \wp(A,B,\ldots) \longrightarrow \wp(R), \dot{\subseteq} \rangle \underset{\alpha_a}{\overset{\gamma_a}{\leftrightarrows}} \langle \wp(B,\ldots) \longrightarrow \wp(R), \dot{\subseteq} \rangle$$

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

$\Longrightarrow$ no more global time between processes

$\Longrightarrow$ 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:

global time →

$t = w(\mathsf{start}, \mathrm{x}, 0)\ w(\mathsf{start}, \mathrm{y}, 0)\ r(\mathtt{P0}, \mathtt{x}, 1)\ \mathfrak{rf}[w(\mathtt{P1}, \mathtt{x}, 1), r(\mathtt{P0}, \mathtt{x}, 1)])\ w(\mathtt{P0}, \mathtt{y}, 1)\ r(\mathtt{P1}, \mathtt{y}, 1)$
$w(\mathtt{P1}, \mathtt{x}, 1)\ \mathfrak{rf}[w(\mathtt{P0}, \mathtt{y}, 1), r(\mathtt{P1}, \mathtt{y}, 1)]\ r(\mathsf{finish}, \mathrm{x})\ \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\mathsf{finish}, \mathrm{x}, 1)]$
$r(\mathsf{finish}, \mathrm{y}, 1)\ \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\mathsf{finish}, \mathrm{y}, 1)]$

- Parallel executions with interleaved communications:

t = Initialization: $w(\mathsf{start}, \mathrm{x}, 0)\ w(\mathsf{start}, \mathrm{y}, 0)$

local time →

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

local time →

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

Finalization: $r(\mathsf{finish}, \mathrm{x})\ \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\mathsf{finish}, \mathrm{x}, 1)]$
$r(\mathsf{finish}, \mathrm{y}, 1)\ \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\mathsf{finish}, \mathrm{y}, 1)]$

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)

$\Longrightarrow$ no more global time between processes

$\Longrightarrow$ no more global/local time for communications

True parallelism with separate communications

- Parallel executions with interleaved communications:

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

P0: $r(\mathtt{P0}, \mathtt{x}, 1)\ w(\mathtt{P0}, \mathtt{y}, 1)$ *local time*

P1: $r(\mathtt{P1}, \mathtt{y}, 1)\ w(\mathtt{P1}, \mathtt{x}, 1)$ *local time*

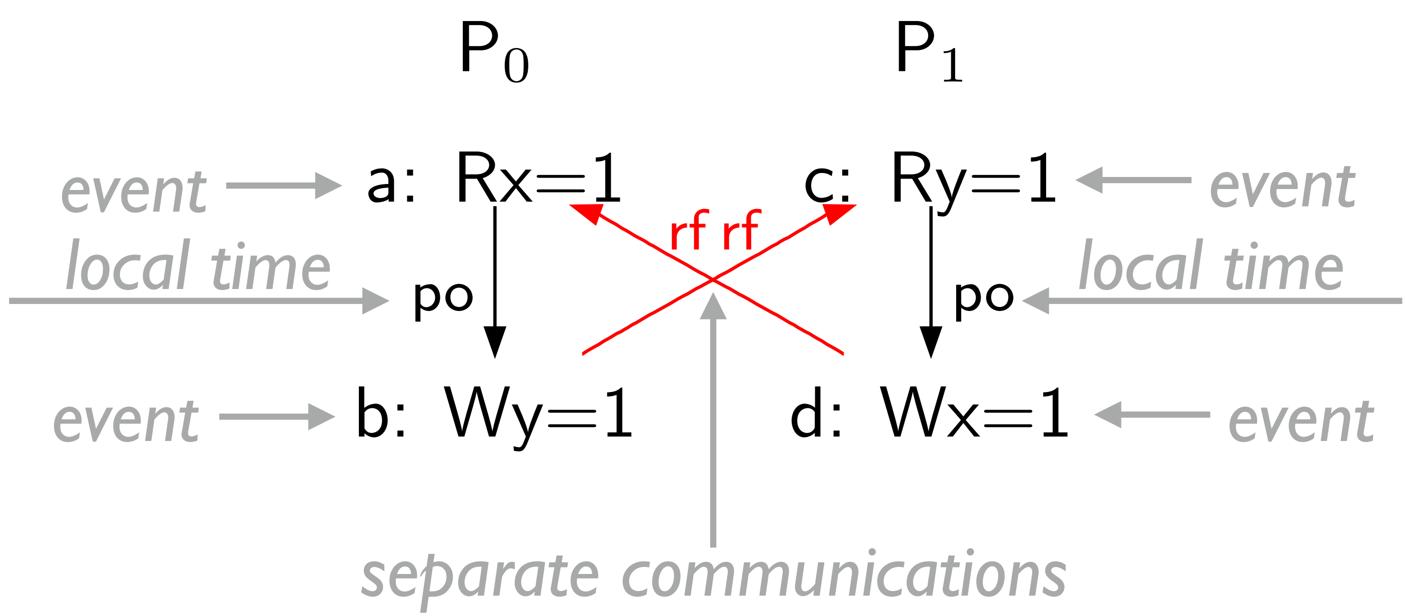
Finalization: $r(\text{finish}, \mathrm{x})\ \mathfrak{rf}[w(\mathtt{P1}, \mathrm{x}, 1), r(\text{finish}, \mathrm{x}, 1)]$
$r(\text{finish}, \mathrm{y}, 1)\ \mathfrak{rf}[w(\mathtt{P0}, \mathrm{y}, 1), r(\text{finish}, \mathrm{y}, 1)]$

Communications:

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

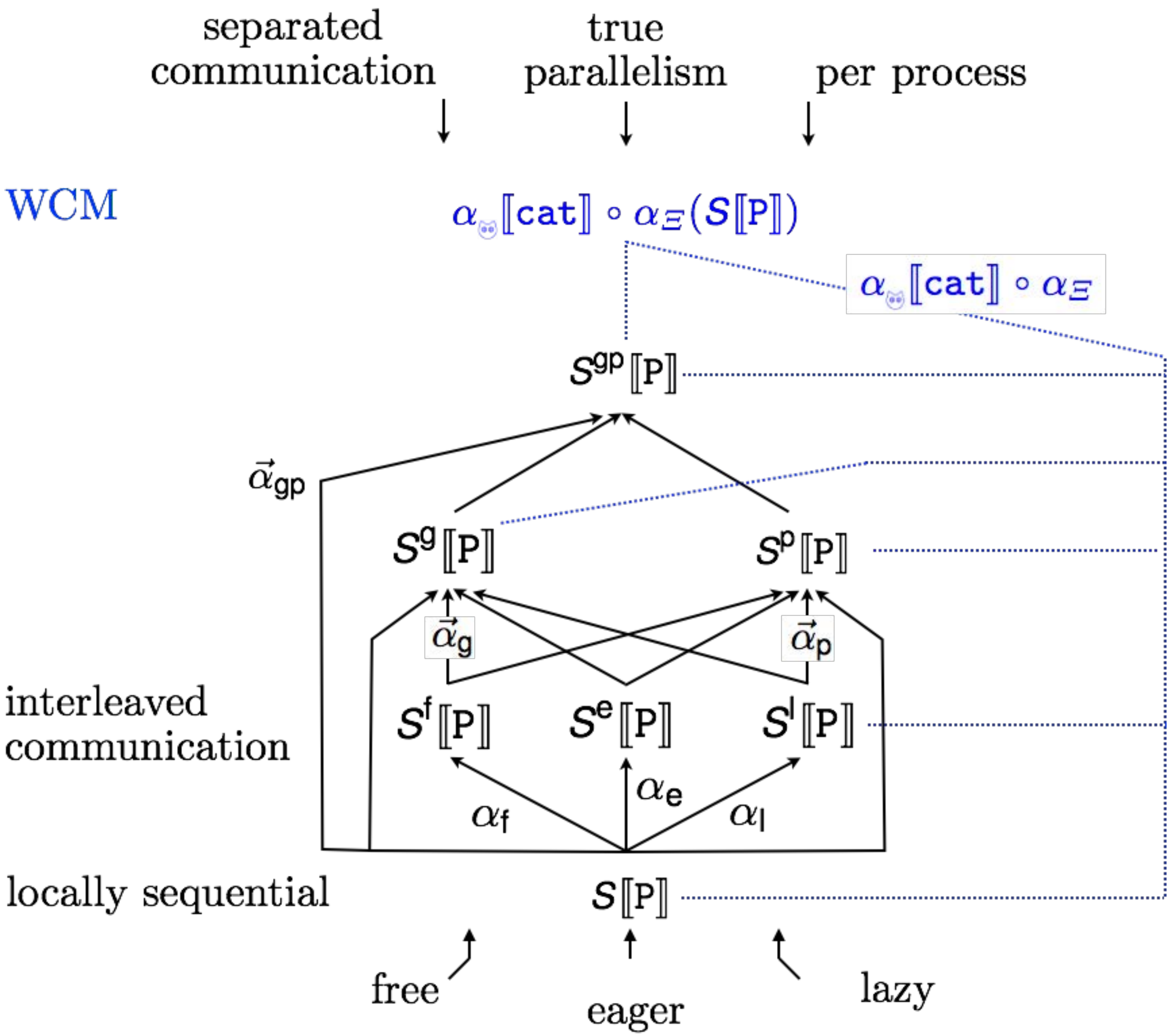
True parallelism with separate communications

- 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):

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

$$\nu^p(\tau, k) \triangleq \boldsymbol{\lambda}\, \mathtt{x}_\theta \bullet \mathsf{V}^p_{(32)}[\![\mathtt{x}_\theta]\!](\tau, k)$$

Prefixes, transitions, ...

- Abstract traces by their prefixes:

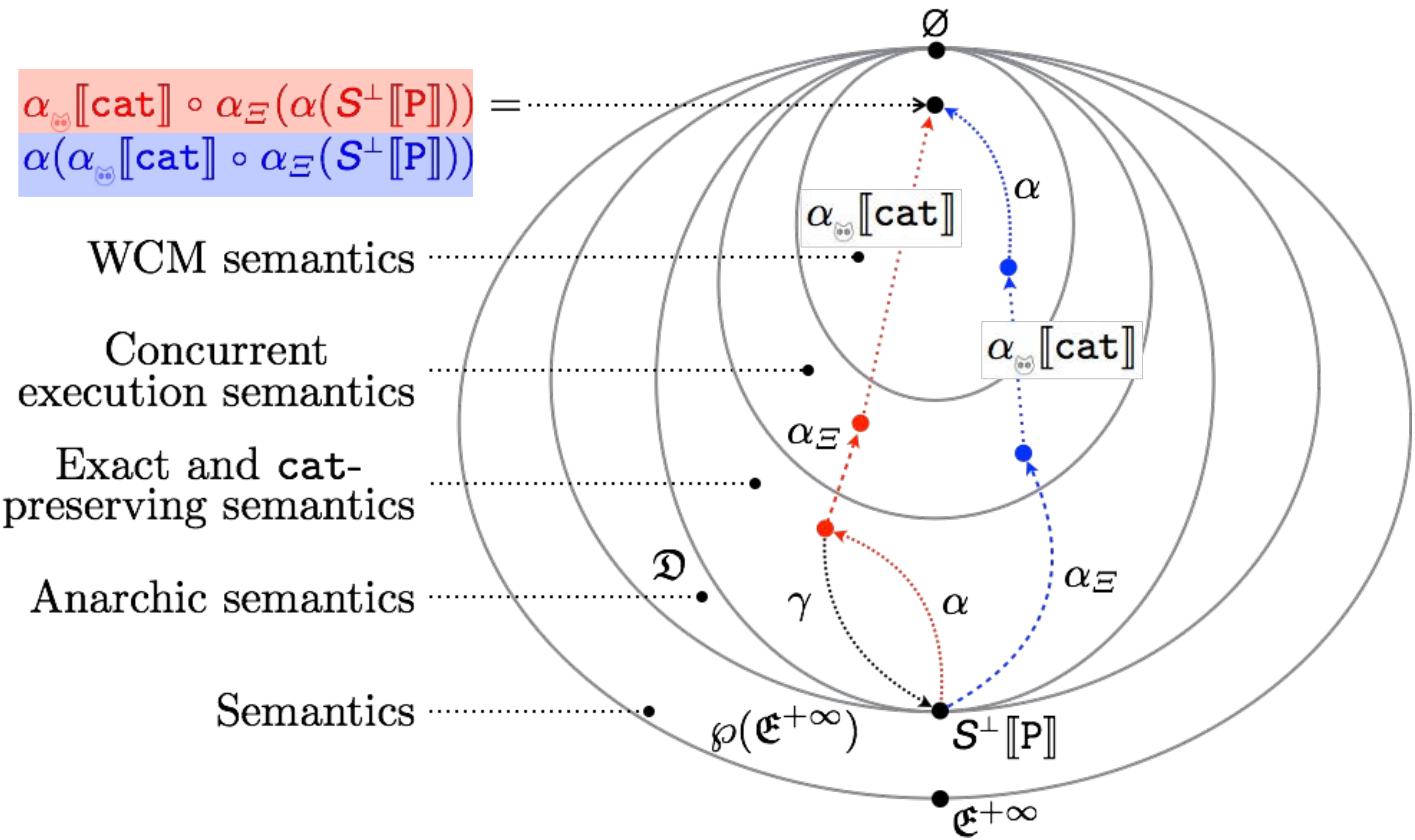
$$
\begin{aligned}
\overleftarrow{\alpha}(\mathcal{S}) &\triangleq \bigcup\{\overleftarrow{\alpha}(\tau) \mid \tau \in \mathcal{S}\} \\
\overleftarrow{\alpha}(\tau) &\triangleq \{\tau\langle\!|j\langle\!| \mid j \in [1, 1 + |\tau|[\} \\
\tau\langle\!|j\langle\!| &\triangleq \langle \xrightarrow{\overline{\tau}_i} \underline{\tau}_i \mid i \in [1, 1 + j[\rangle
\end{aligned}
$$

- and transitions: extract transitions from traces

$\Longrightarrow$ communication fairness is lost, inexact abstraction,
$\Longrightarrow$ add fairness condition
$\Longrightarrow$ 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 $\alpha \llbracket \texttt{cat} \rrbracket$ applies equally to any concurrent execution abstraction α_{Ξ} of any interleaved/truly parallel semantics in the hierarchy
- 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)

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