Resource tracking concurrent games

Aurore Alcolei, Pierre Clairambault, Olivier Laurent
ENS Lyon, LIP

Fossacs 2019, Prague – April, 9 2019

Time analysis

P =

```
newref r in
    wait(2)  ||  wait(1)
    !r       ||  r := true
             ||  wait(2)
```

Semantics.

$$M \Downarrow v \qquad\qquad [\![M]\!] = [\![v]\!]\ , \qquad\qquad v \in \{\mathsf{true}, \mathsf{false}\}$$

Time analysis

P =

```
newref r in
    wait(2) || wait(1)
    !r      || r := true
            || wait(2)
```

Semantics.

$$M \Downarrow^{t} v \qquad\qquad [\![M]\!] = [\![v]\!]^{t}, \qquad\qquad v \in \{\mathrm{true}, \mathrm{false}\}$$

Q. What is the minimal amount of time necessary to run P?

$\rightarrow$ to get true?

$\rightarrow$ to get false?

The $\mathbb{R}$-IPA language

- Types:

$$
\begin{array}{lcl}
\mathcal{B} & := & \textbf{com} \mid \textbf{bool} \mid \textbf{mem}_R \mid \textbf{mem}_W \\
A, B & := & \mathcal{B} \mid A \multimap B
\end{array}
$$

- Syntax:

$$
\begin{array}{lcll}
M, N & := & \mid x \mid \lambda x.t \mid MN & \\
 & & \mid \text{true} \mid \text{false} \mid \text{ifte } b\ M\ N & \\
 & & \mid \text{skip} \mid M; N \mid M \mid\mid N \mid \bot & \\
 & & \mid \text{wait}(\alpha) & \text{with } \alpha \in \mathbb{R} \\
 & & \mid \text{newref } r \text{ in } M \mid !M \mid M := \text{true} &
\end{array}
$$

Affine typing.

$$
\frac{\Gamma, r : \textbf{mem}_R, r : \textbf{mem}_W \vdash M : A}{\Gamma \vdash \text{newref } r \text{ in } M : A}
\qquad
\frac{\Gamma \vdash M : \textbf{bool} \quad \Delta \vdash N : \textbf{com}}{\Gamma, \Delta \vdash M \mid\mid N : \textbf{bool}}
$$

Examples

Coin

newref r in (r:= true ∥ !r)

Strictness testing.

$\lambda\ f^{\mathbf{com} \multimap \mathbf{com}}$. newref r in
(f (r:=true)) ; !r

Parallelism testing.

$\lambda\ f^{\mathbf{com} \multimap \mathbf{com} \multimap \mathbf{com}}$. newref x,y,$z_1$,$z_2$ in
f (if (!x) then (skip) else (z_1 := true ; y := true))
(if (!y) then (skip) else (z_2 := true ; x := true)) ;
(!z_1) and (!z_2)

Examples

Coin

newref r in (r:= true $\|$!r)

Strictness testing.

$\lambda\ f^{\mathbf{com} \multimap \mathbf{com}}$. newref r in
(f (r:=true)) ; !r

Parallelism testing.

$\lambda\ f^{\mathbf{com} \multimap \mathbf{com} \multimap \mathbf{com}}$. newref x,y,$z_1$,$z_2$ in
f (if (!x) then (skip) else (z_1 := true ; y := true))
(if (!y) then (skip) else (z_2 := true ; x := true)) ;
(!z_1) and (!z_2)

Examples

Coin

newref r in (r:= true $\|$!r)

Strictness testing.

$\lambda\ f^{\mathbf{com} \multimap \mathbf{com}}$. newref r in
(f (r:=true)) ; !r

Parallelism testing.

$\lambda\ f^{\mathbf{com} \multimap \mathbf{com} \multimap \mathbf{com}}$. newref x,y,$z_1$,$z_2$ in
f (if (!x) then (skip) else (z_1 := true ; y := true))
(if (!y) then (skip) else (z_2 := true ; x := true)) ;
(!z_1) and (!z_2)

The $\mathbb{R}$-IPA language (2)

- Interleaving-based small-step operational semantics:

$$\langle M, s, t\rangle, \quad \mathcal{L} \vdash M : B \text{ with } B \in \mathcal{B},\ t \in \mathbb{R},\ s : \mathcal{L} \to \{\text{true}, \text{false}\}$$

Reduction rules:

$$\langle (\lambda x.M)N, s, t\rangle \to \langle t[N/x], s, t\rangle \qquad \langle \text{wait}(\alpha), s, t\rangle \to \langle \text{skip}, s, t+\alpha\rangle$$

$$\langle \text{skip}; M, s, t\rangle \to \langle M, t\rangle \qquad \langle \text{skip} \mid\mid M, s, t\rangle \to \langle M, s, t\rangle$$

$$\langle r := \text{true}, s, t\rangle \to \langle \text{skip}, s[r \mapsto \text{true}], t\rangle \qquad \ldots$$

Contextual rules:

$$\frac{\langle M, s, t\rangle \to \langle M', s', t'\rangle}{\langle M || N, s, t\rangle \to \langle M' || N, s', t'\rangle} \qquad \ldots$$

Definition

For $v \in \{\text{true}, \text{false}, \text{skip}\}$, $\quad M \Downarrow^t v \quad$ iff $\quad \langle M, \emptyset, 0\rangle \to^* \langle v, s, t\rangle$.

The $\mathbb{R}$-IPA language (2)

- Interleaving-based small-step operational semantics:

$$\langle M, s, t \rangle, \quad \mathcal{L} \vdash M : B \text{ with } B \in \mathcal{B},\ t \in \mathbb{R},\ s : \mathcal{L} \to \{\text{true}, \text{false}\}$$

Reduction rules:

$$\langle (\lambda x.M)N, s, t\rangle \to \langle t[N/x], s, t\rangle \qquad \langle \text{wait}(\alpha), s, t\rangle \to \langle \text{skip}, s, t+\alpha\rangle$$

$$\langle \text{skip}; M, s, t\rangle \to \langle M, t\rangle \qquad \langle \text{skip} \mid\mid M, s, t\rangle \to \langle M, s, t\rangle$$

$$\langle r := \text{true}, s, t\rangle \to \langle \text{skip}, s[r \mapsto \text{true}], t\rangle \qquad \ldots$$

Contextual rules:

$$\frac{\langle M, s, t\rangle \to \langle M', s', t'\rangle}{\langle M || N, s, t\rangle \to \langle M' || N, s', t'\rangle} \qquad \ldots$$

Definition

For $v \in \{\text{true}, \text{false}, \text{skip}\}$, $\quad M \Downarrow^t v \quad$ iff $\quad \langle M, \emptyset, 0\rangle \to^* \langle v, s, t\rangle$.

The $\mathbb{R}$-IPA language (2)

- Interleaving-based small-step operational semantics:

$$\langle M, s, t\rangle, \quad \mathcal{L} \vdash M : B \text{ with } B \in \mathcal{B},\ t \in \mathbb{R},\ s : \mathcal{L} \to \{\text{true}, \text{false}\}$$

Reduction rules:

$$\langle (\lambda x.M)N, s, t\rangle \to \langle t[N/x], s, t\rangle \qquad \langle \text{wait}(\alpha), s, t\rangle \to \langle \text{skip}, s, t+\alpha\rangle$$

$$\langle \text{skip}; M, s, t\rangle \to \langle M, t\rangle \qquad \langle \text{skip} \mid\mid M, s, t\rangle \to \langle M, s, t\rangle$$

$$\langle r := \text{true}, s, t\rangle \to \langle \text{skip}, s[r \mapsto \text{true}], t\rangle \qquad \ldots$$

Contextual rules:

$$\frac{\langle M, s, t\rangle \to \langle M', s', t'\rangle}{\langle M || N, s, t\rangle \to \langle M' || N, s', t'\rangle} \qquad \ldots$$

Definition

For $v \in \{\text{true}, \text{false}, \text{skip}\}$, $\quad M \Downarrow^t v \quad$ iff $\quad \langle M, \emptyset, 0\rangle \to^* \langle v, s, t\rangle$.

Example

P =

```
newref r in
    wait(2) || wait(1)
    !r      || r := true
            || wait(2)
```

0

Example

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

2

wait(2),

Example

P =

newref r in		
wait(2)	wait(1)	
!r	r := true	2
	wait(2)	

wait(2), !r,

Example

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

3

wait(2), !r, wait(1),

Example

P =

```
newref r in
    wait(2)  ||  wait(1)
    !r       ||  r := true        3
             ||  wait(2)
```

wait(2), !r, wait(1), r:=true,

Example

P =

```
newref r in
    wait(2)  ||  wait(1)
    !r       ||  r := true        5
             ||  wait(2)
```

wait(2), !r, wait(1), r:=true, wait(2) $P \Downarrow^{5}$ false

Example

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

1

wait(2), !r, wait(1), r:=true, wait(2) $P \Downarrow^5$ false

wait(1),

Example

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

3

wait(2), !r, wait(1), r:=true, wait(2) $P \Downarrow^{5}$ false
wait(1), wait(2),

Example

P =

```
newref r in
    wait(2)  ||  wait(1)
    !r       ||  r := true        3
             ||  wait(2)
```

wait(2), !r, wait(1), r:=true, wait(2) $P \Downarrow^5$ false
wait(1), wait(2), r:=true,

Example

P =

newref r in

wait(2)	wait(1)	
!r	r := true	3
	wait(2)	

wait(2), !r, wait(1), r:=true, wait(2) $P \Downarrow^5$ false

wait(1), wait(2), r:=true, !r,

Example

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

5

wait(2), !r, wait(1), r:=true, wait(2) $P \Downarrow^5$ false
wait(1), wait(2), r:=true, !r, wait(2) $P \Downarrow^5$ true
...

Slot games [Ghica05]

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

5

wait(2), !r, wait(1), r:=true, wait(2) $\quad P \Downarrow^5$ false
wait(1), wait(2), r:=true, !r, wait(2) $\quad P \Downarrow^5$ true
...

$[\![P]\!] = \{\text{run } ② ① ② \text{ ff}, \text{ run } ① ② ② \text{ tt}, \ldots\}$

Computational adequacy: $M \Downarrow^r v$ iff $\exists t \in [\![M]\!]$ st $|t| = r$

Slot games [Ghica05]

P =

newref r in

wait(2)	wait(1)
!r	r := true
	wait(2)

5

wait(2), !r, wait(1), r:=true, wait(2) $\quad P \Downarrow^{5}$ false
wait(1), wait(2), r:=true, !r, wait(2) $\quad P \Downarrow^{5}$ true
...

$[\![P]\!] = \{\text{run } ⑤ \text{ ff},\ \text{run } ⑤ \text{ tt}\}$

Computational adequacy: $\quad M \Downarrow^{r} v \quad$ iff $\quad \exists t \in [\![M]\!]$ st $|t| = r$

True concurrency?

Q: What about <u>multicore</u> systems?

P =

```
newref (r,false) in
    wait(2)  ||  wait(1)
    !r       ||  r := true
             ||  wait(2)
```

0

True concurrency?

Q: What about <u>multicore</u> systems?

P =

```
newref (r,false) in
    wait(2)  ||  wait(1)
    !r       ||  r := true        2
             ||  wait(2)
```

True concurrency?

Q: What about multicore systems?

P =

newref (r,false) in

wait(2)	wait(1)
!r	r := true
	wait(2)

2

True concurrency?

Q: What about multicore systems?

P =

```
newref (r,false) in
    wait(2)  ||  wait(1)
    !r       ||  r := true        2
             ||  wait(2)
```

True concurrency?

Q: What about multicore systems?

P =

newref (r,false) in

wait(2)	‖	wait(1)
!r	‖	r := true
	‖	wait(2)

4

$P \Downarrow^4$ false !

True concurrency?

Q: What about <u>multicore</u> systems?

P =

```
newref (r,false) in
    wait(2) || wait(1)
    !r      || r := true        1
            || wait(2)
```

True concurrency?

Q: What about multicore systems?

P =

```
newref (r,false) in
    wait(2)  ||  wait(1)
    !r       ||  r := true        1
             ||  wait(2)
```

True concurrency?

Q: What about multicore systems?

P =

```
newref (r,false) in
    wait(2)  ||  wait(1)
    !r       ||  r := true
             ||  wait(2)
```

3

True concurrency?

Q: What about <u>multicore</u> systems?

P =

```
newref (r,false) in
   wait(2) || wait(1)
   !r      || r := true        3
           || wait(2)
```

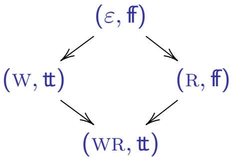
$P \Downarrow^3$ true !

True concurrency?

Q: What about multicore systems?

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \parallel N, s, t\rangle \to^{*} \langle M' \parallel N', s' \bowtie s'', \max(t', t'')\rangle} \; s, s' \text{ non interfering}$$

$s : \mathcal{L} \to$



- s, s' are non interfering iff $\forall \ell \in \mathcal{L}$, $s(\ell) \leq s(\ell')$ or $s'(\ell) \leq s(\ell)$
- $s' \bowtie s''(\ell) = \bigvee(s'(\ell), s''(\ell))$

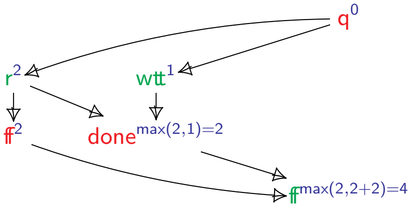
Annotated concurrent games:

a resource sensitive semantics

Annotated concurrent games

```
newref (r,false) in
    wait(2)  ||  wait(1)
    !r       ||  r := true
             ||  wait(2)
```

$P \Downarrow^4$ false

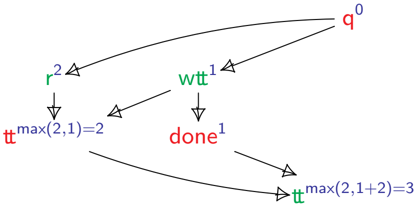


[CC16] + time annotations

Annotated concurrent games

```
newref (r,false) in
    wait(2) || wait(1)
    !r      || r := true
            || wait(2)
```

$P \Downarrow^3$ true



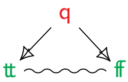
[CC16] + time annotations

Concurrent games

Based on event structures

- Types as games

$[\![\mathbf{com}]\!] =$ 

$[\![\mathbf{bool}]\!] =$ 

$\mathscr{C}(A)$ is the set of **configurations**: down-closed compatible subsets of A.

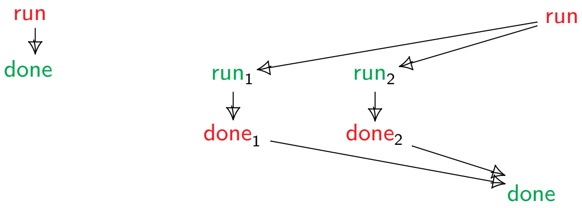
Constructions on games.

- If A is a game, $A^\perp$ has the same structure with polarity inverted.
- If A, B are games, $A \otimes B$ has events $|A| + |B|$, and components inherited.

Concurrent games

- Programs as strategies

$[\![\text{skip}]\!] : \ \textbf{com} \qquad [\![\ ||\ ]\!] : \ \textbf{com} \otimes \textbf{com} \ \to\!\!\!\!+ \ \textbf{com}$



Definition

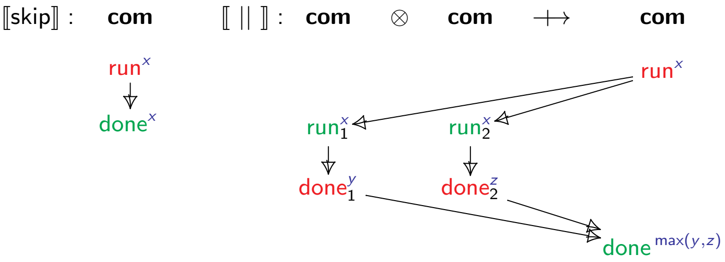
A play $(q, \leq_q) : A$ is a partial order s.t.:

- $*$ (rule respecting) $\mathscr{C}(q) \subseteq \mathscr{C}(A)$
- $*$ (courteous) $a \rightarrow b$

A strategy is a down-closed set of plays (with extra conditions).

Annotated concurrent games

- Programs as $\mathbb{R}$-strategies



Definition

A play $(q, \leq_q) : A$ is a partial order s.t.:

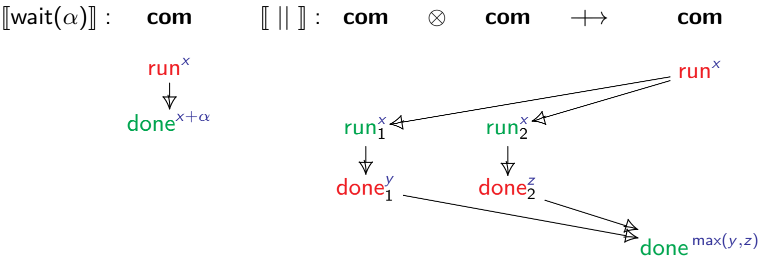
- $*$ (rule respecting) $\mathscr{C}(q) \subseteq \mathscr{C}(A)$
- $*$ (courteous) $a \rightarrow b$

A $\mathbb{R}$-annotation for q is a mapping $\lambda : (s \in |q|^P) \longrightarrow (\mathbb{R}^{[s]^O} \to \mathbb{R})$.

A $\mathbb{R}$-strategy is a down-closed set of $\mathbb{R}$-annotated plays (with extra conditions).

Annotated concurrent games

- Programs as $\mathbb{R}$-strategies



Definition

A play $(q, \leq_q) : A$ is a partial order s.t.:

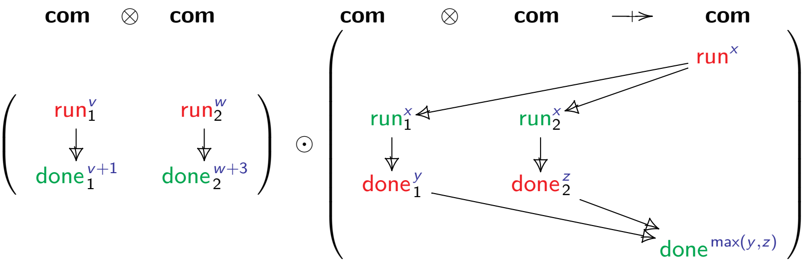
$*$ (rule respecting) $\mathscr{C}(q) \subseteq \mathscr{C}(A)$ $*$ (courteous) $a \rightarrowtriangle b$

A $\mathbb{R}$-annotation for q is a mapping $\lambda : (s \in |q|^P) \longrightarrow (\mathbb{R}^{[s]^O} \to \mathbb{R})$.

A $\mathbb{R}$-strategy is a down-closed set of $\mathbb{R}$-annotated plays (with extra conditions).

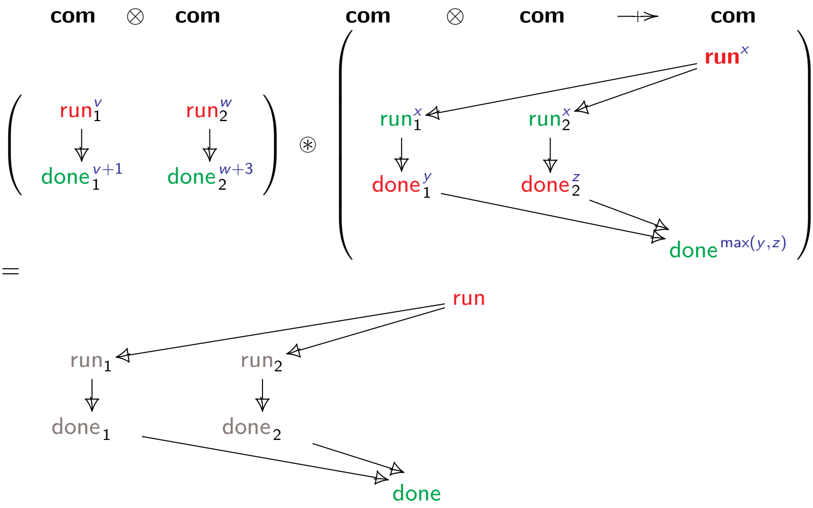
Composition example

$$[\![\mathsf{wait}(1) \mid\mid \mathsf{wait}(3)]\!] \quad = \quad [\![\ \mid\mid\ ]\!] \odot ([\![\mathsf{wait}(1)]\!] \otimes [\![\mathsf{wait}(3)]\!])$$



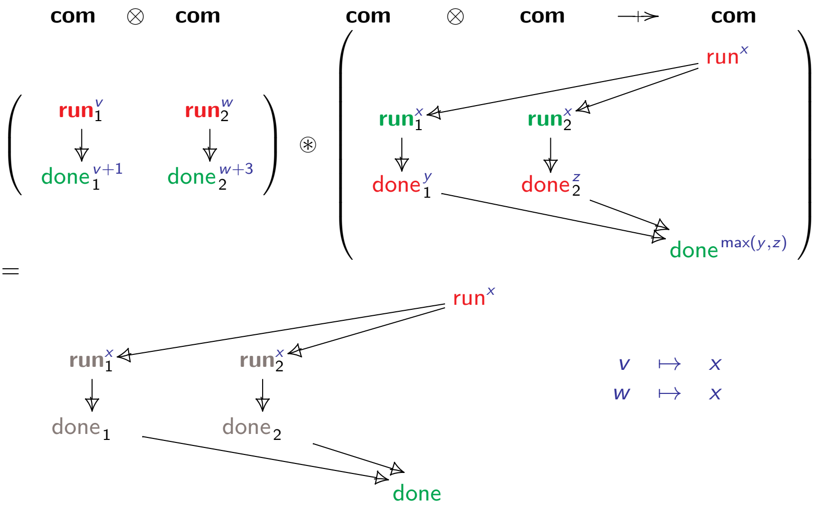
Composition example

$$[\![\mathsf{wait}(1) \parallel \mathsf{wait}(3)]\!] \quad = \quad [\![\ \parallel\ ]\!] \odot ([\![\mathsf{wait}(1)]\!] \otimes [\![\mathsf{wait}(3)]\!])$$



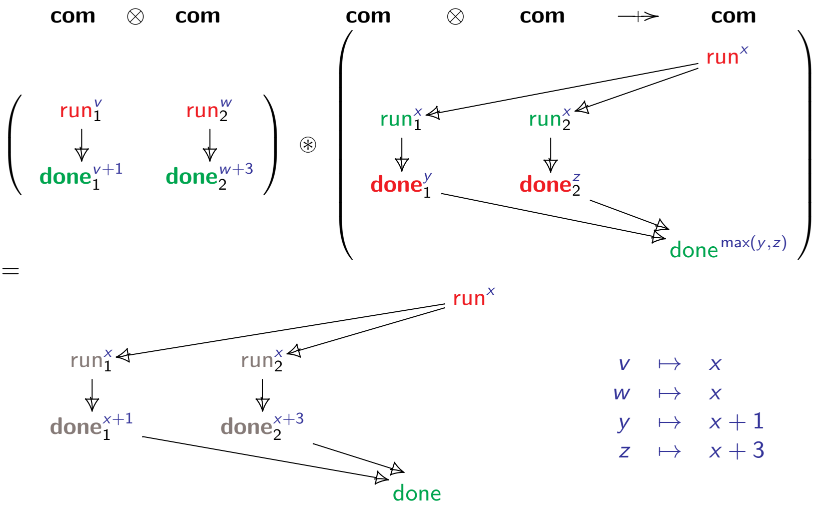
Composition example

$$[\![\text{wait}(1) \parallel \text{wait}(3)]\!] \quad = \quad [\![\ \parallel\ ]\!] \odot \left([\![\text{wait}(1)]\!] \otimes [\![\text{wait}(3)]\!]\right)$$



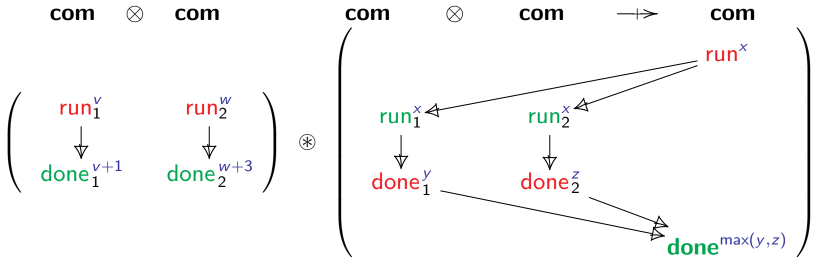
Composition example

$$[\![\text{wait}(1) \mid\mid \text{wait}(3)]\!] \quad = \quad [\![\ \mid\mid\ ]\!] \odot ([\![\text{wait}(1)]\!] \otimes [\![\text{wait}(3)]\!])$$

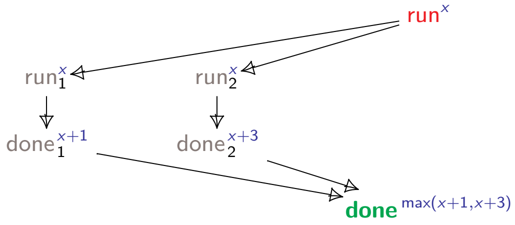


Composition example

$$[\![\mathsf{wait}(1) \mid\mid \mathsf{wait}(3)]\!] \quad = \quad [\![\ \mid\mid\ ]\!] \odot ([\![\mathsf{wait}(1)]\!] \otimes [\![\mathsf{wait}(3)]\!])$$



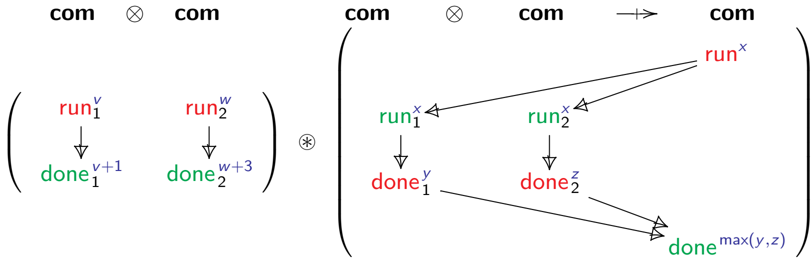
=



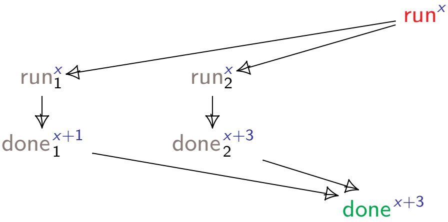
$$v \mapsto x$$
$$w \mapsto x$$
$$y \mapsto x+1$$
$$z \mapsto x+3$$

Composition example

$$[\![\mathsf{wait}(1) \parallel \mathsf{wait}(3)]\!] \quad = \quad [\![\ \parallel\ ]\!] \odot ([\![\mathsf{wait}(1)]\!] \otimes [\![\mathsf{wait}(3)]\!])$$



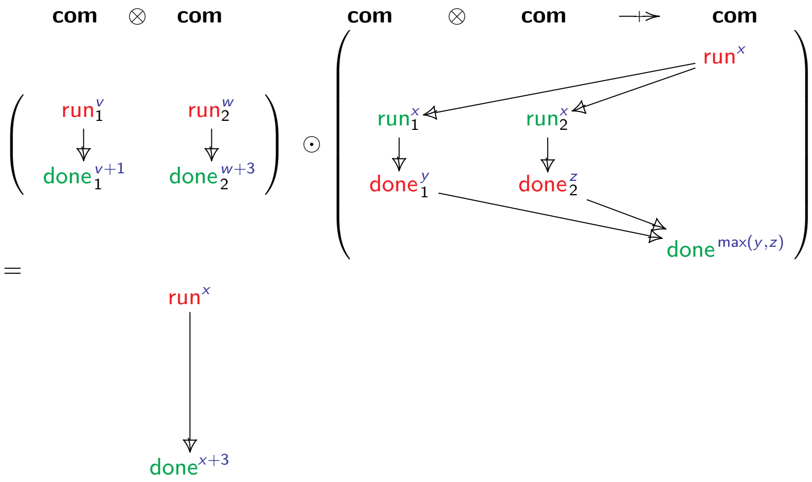
=



$$v \mapsto x$$
$$w \mapsto x$$
$$y \mapsto x+1$$
$$z \mapsto x+3$$

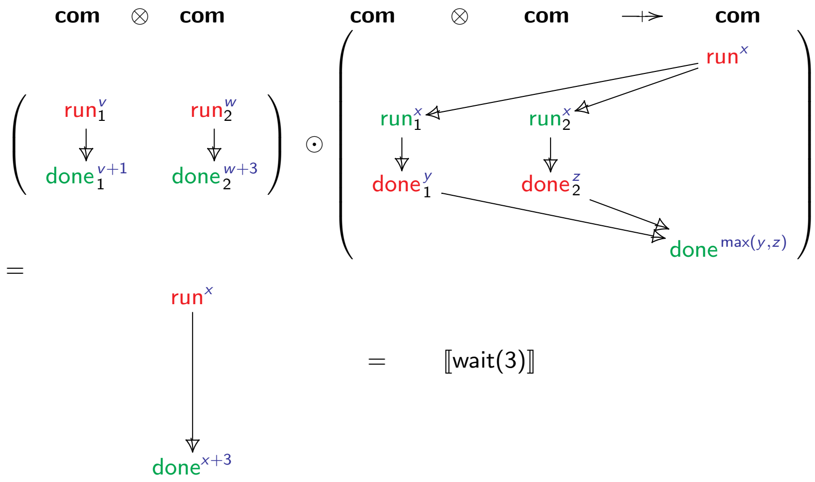
Composition example

$$\llbracket \mathrm{wait}(1) \mid\mid \mathrm{wait}(3) \rrbracket \quad = \quad \llbracket \;\mid\mid\; \rrbracket \odot (\llbracket \mathrm{wait}(1) \rrbracket \otimes \llbracket \mathrm{wait}(3) \rrbracket)$$



Composition example

$$[\![\mathsf{wait}(1) \mid\mid \mathsf{wait}(3)]\!] \quad = \quad [\![\ \mid\mid\ ]\!] \odot ([\![\mathsf{wait}(1)]\!] \otimes [\![\mathsf{wait}(3)]\!])$$



$\mathbb{R}$-CG$_{_}$

Theorem

Games and $\mathbb{R}$-strategies form a symmetric monoidal closed category (smcc)

In fact, $\mathbb{R}$-CG$_{_}$ also has products.

$$[\![\Gamma \vdash M : A]\!] : [\![\Gamma]\!] \twoheadrightarrow [\![A]\!]$$

$$\begin{array}{lcll} M, N & := & \mid x \mid \lambda x.t \mid MN & \checkmark \\ & & \mid \text{true} \mid \text{false} \mid \text{ifte } b\ M\ N & \checkmark \\ & & \mid \text{skip} \mid M; N \mid M \mid\mid N \mid \bot & \checkmark \\ & & \mid \text{wait}(\alpha) & \checkmark \\ & & \mid \text{newref } r \text{ in } M \mid !M \mid M := \text{true} & \end{array}$$

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] : \mathbf{mem}$$

Interpretation: Shared memory

$[\![\text{newref } r \text{ in } M]\!]$

$= \quad [\![\text{cell}]\!] \odot [\![M]\!]$

$[\![\textbf{mem}]\!] = \quad [\![\textbf{mem}_R]\!] \quad \otimes \quad [\![\textbf{mem}_W]\!]$

$[\![\text{cell}]\!] \quad : \quad \textbf{mem}$

r

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$$

Interpretation: Shared memory

$[\![\text{newref } r \text{ in } M]\!]$

$= \quad [\![\text{cell}]\!] \odot [\![M]\!]$

$[\![\mathbf{mem}]\!] = \quad [\![\mathbf{mem}_R]\!] \quad \otimes \quad [\![\mathbf{mem}_W]\!]$

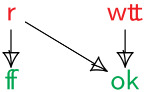
$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] : \mathbf{mem}$$



Interpretation: Shared memory

$[\![\text{newref } r \text{ in } M]\!]$

$= \quad [\![\text{cell}]\!] \odot [\![M]\!]$

$[\![\textbf{mem}]\!] = \quad [\![\textbf{mem}_R]\!] \quad \otimes \quad [\![\textbf{mem}_W]\!]$

$[\![\text{cell}]\!] \quad : \quad \textbf{mem}$

wtt

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] : \mathbf{mem}$$

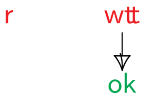
Interpretation: Shared memory

$[\![\text{newref } r \text{ in } M]\!]$

$= \quad [\![\text{cell}]\!] \odot [\![M]\!]$

$[\![\mathbf{mem}]\!] = \quad [\![\mathbf{mem}_R]\!] \quad \otimes \quad [\![\mathbf{mem}_W]\!]$

$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$

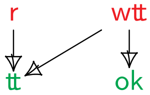


Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] : \mathbf{mem}$$



Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$$

r wtt

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

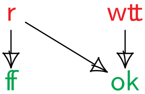
$$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$$

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$$



Interpretation: Shared memory

$[\![\text{newref } r \text{ in } M]\!]$

$= \quad [\![\text{cell}]\!] \odot [\![M]\!]$

$$[\![\mathbf{mem}]\!] = \quad [\![\mathbf{mem}_R]\!] \quad \otimes \quad [\![\mathbf{mem}_W]\!]$$

$[\![\text{cell}]\!] \quad : \quad \mathbf{mem}$

r wtt

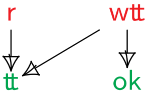
Interpretation: Shared memory

$[\![\text{newref } r \text{ in } M]\!]$

$= \quad [\![\text{cell}]\!] \odot [\![M]\!]$

$[\![\textbf{mem}]\!] = \quad [\![\textbf{mem}_R]\!] \quad \otimes \quad [\![\textbf{mem}_W]\!]$

$[\![\text{cell}]\!] \quad : \quad \textbf{mem}$



Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

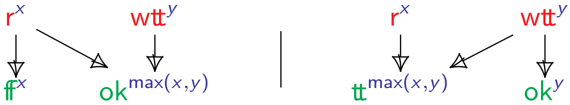
$$[\![\text{cell}]\!] : \mathbf{mem}$$

Interpretation: Shared memory

$$[\![\text{newref } r \text{ in } M]\!] = [\![\text{cell}]\!] \odot [\![M]\!]$$

$$[\![\mathbf{mem}]\!] = [\![\mathbf{mem}_R]\!] \otimes [\![\mathbf{mem}_W]\!]$$

$$[\![\text{cell}]\!] : \mathbf{mem}$$

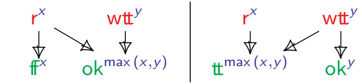


Soundness

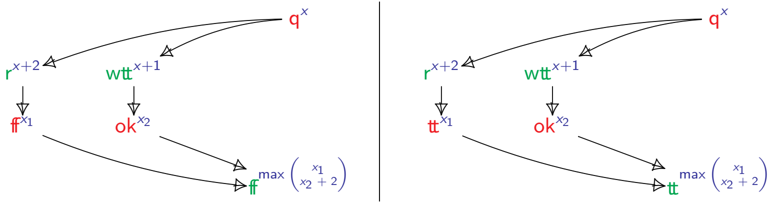
P =
newref r in

wait(2) !r	wait(1) r := true wait(2)

$[\![\text{cell}]\!]$: **mem**



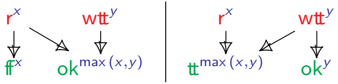
$[\![P']\!]$: **mem** ⇸ **bool**



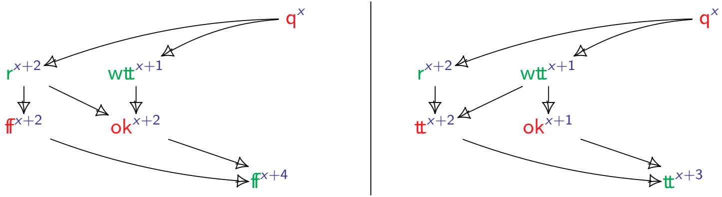
Soundness

```
P =
    newref  r in
        wait(2)  ||  wait(1)
        !r       ||  r := true
                 ||  wait(2)
```

$[\![\mathsf{cell}]\!]$: **mem**



$[\![P' \circledast \mathsf{cell}]\!]$: **mem** $\longrightarrow$ **bool**

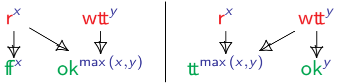


Soundness

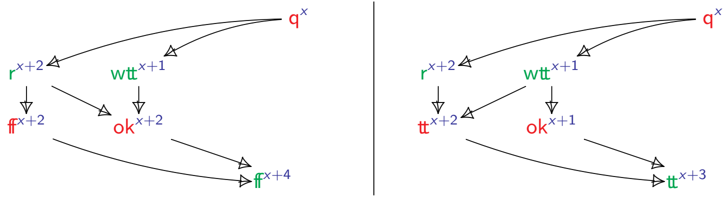
P =

```
    newref  r in
        wait(2)  ||  wait(1)
        !r       ||  r := true
                 ||  wait(2)
```

$[\![\text{cell}]\!]$: **mem**



$[\![P' \circledast \text{cell}]\!]$: **mem** $\multimap$ **bool**



Theorem

If $M \Downarrow^t v$ *then* $q^x \multimap v^{x+t'} \in [\![M]\!]$ *with* $t' \leq t$.

What resources?

Theorem

If $M \Downarrow^r v$ *then* $q^x \multimap v^{x+r'} \in [\![M]\!]$ *with* $r' \leq r$.

$$\mathsf{wait}(\alpha),\ \ \alpha \in \mathbb{R} \qquad \rightsquigarrow \qquad \mathsf{consume}(\alpha),\ \ \alpha \in \mathcal{R}$$

Def. Resource bimonoid: $(\mathcal{R}, 0, ;, \|, \leq)$

- $(\mathcal{R}, 0, ;, \leq)$ ordered monoid
- $(\mathcal{R}, 0, \|, \leq)$ commutative ordered monoid
- $\|$ is idempotent, i.e. $r \parallel r = r$

	$\mathcal{R}$	;	$\|$
time	$\mathbb{R}$	$+$	max

What resources?

Theorem

If $M \Downarrow^r v$ *then* $q^x \multimap v^{x+r'} \in [\![M]\!]$ *with* $r' \leq r$.

$$\mathsf{wait}(\alpha),\ \ \alpha \in \mathbb{R} \qquad \rightsquigarrow \qquad \mathsf{consume}(\alpha),\ \ \alpha \in \mathcal{R}$$

Def. Resource bimonoid: $(\mathcal{R}, 0, ;, \|, \leq)$

- $(\mathcal{R}, 0, ;, \leq)$ ordered monoid
- $(\mathcal{R}, 0, \|, \leq)$ commutative ordered monoid
- $\|$ is idempotent, i.e. $r \parallel r = r$

	$\mathcal{R}$	;	$\|$
time	$\mathbb{R}$	$+$	max
parametric time	$\mathbb{R} \to \mathbb{R}$	$+$	max
permission	$\mathcal{P}(P)$	$\cup$	$\cup$
~~energy~~	~~$\mathbb{R}$~~	~~max~~	~~$+$~~

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

newref r in

wait(2)	wait(1)	
!r	r := true	0
wait(1)	wait(2)	

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \max(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

newref r in

wait(2)	wait(1)
!r	r := true
wait(1)	wait(2)

2

$$\frac{\langle M, s, t\rangle \to^* \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^* \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^* \langle M' \mid\mid N', s' \bowtie s'', \max(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

```
newref r in
    wait(2) || wait(1)
    !r      || r := true        2
    wait(1) || wait(2)
```

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \mathsf{max}(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

newref r in

wait(2)	‖	wait(1)	
!r	‖	r := true	2
wait(1)	‖	wait(2)	

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \mathsf{max}(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

newref r in		
wait(2)	‖	wait(1)
!r	‖	r := true
wait(1)	‖	wait(2)

4

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \max(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

```
newref r in
    wait(2) || wait(1)
    !r      || r := true        1
    wait(1) || wait(2)
```

$$\frac{\langle M, s, t\rangle \rightarrow^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \rightarrow^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \rightarrow^{*} \langle M' \mid\mid N', s' \bowtie s'', \mathsf{max}(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

```
newref r in
    wait(2) || wait(1)
    !r      || r := true        1
    wait(1) || wait(2)
```

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \mathsf{max}(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

newref r in			
wait(2)	‖	wait(1)	
!r	‖	r := true	3
wait(1)	‖	wait(2)	

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \mathsf{max}(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

newref r in

wait(2)	wait(1)
!r	r := true
wait(1)	wait(2)

3

$$\frac{\langle M, s, t\rangle \to^* \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^* \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^* \langle M' \mid\mid N', s' \bowtie s'', \max(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

```
newref r in
   wait(2)  ||  wait(1)
   !r       ||  r := true
   wait(1)  ||  wait(2)
```

4

$$\frac{\langle M, s, t\rangle \to^{*} \langle M', s', t'\rangle \quad \langle N, s, t\rangle \to^{*} \langle N', s'', t''\rangle}{\langle M \mid\mid N, s, t\rangle \to^{*} \langle M' \mid\mid N', s' \bowtie s'', \max(t', t'')\rangle}$$

Adequacy?

Q: What is the minimal amount of time necessary to get true?

P =

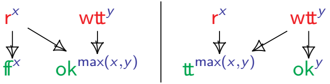
```
newref r in
    wait(2)  ||  wait(1)
    !r       ||  r := true        4
    wait(1)  ||  wait(2)
```

$P \Downarrow^4$ true

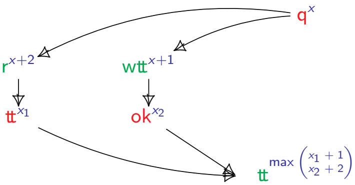
Adequacy?

```
newref r in
    wait(2)  ||  wait(1)
    !r       ||  r := true
    wait(1)  ||  wait(2)
```

$[\![\text{cell}]\!]$: **mem**



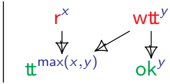
$[\![P']\!]$: **mem** $\rightarrow$ **bool**



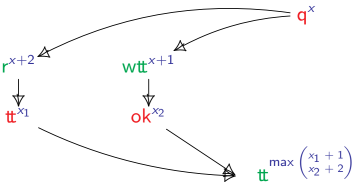
Adequacy?

```
newref r in
    wait(2) || wait(1)
    !r      || r := true
    wait(1) || wait(2)
```

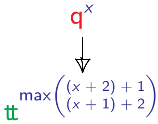
$[\![\mathrm{cell}]\!]$: **mem**



$[\![P']\!]$: **mem** $\to$ **bool**



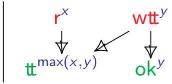
$[\![P]\!]$: **bool**



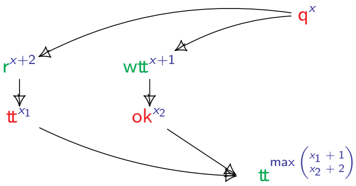
Adequacy?

```
newref r in
    wait(2) || wait(1)
    !r      || r := true
    wait(1) || wait(2)
```

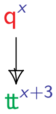
$[\![\text{cell}]\!]$: **mem**



$[\![P']\!]$: **mem** $\rightarrow$ **bool**



$[\![P]\!]$: **bool**



Is $[\![\ ]\!]$ degenerated?

Adequacy (Time)

An efficient small-step semantics :

$$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t\rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1\rangle \qquad \ldots$$

```
newref r in
     wait(2) || wait(1)
    !r       || r := true          0
     wait(1) || wait(2)
```

Theorem

If $q^x \rightharpoonup v^{x+t} \in [\![M]\!]$ then $M \Downarrow^t v$.

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t \rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1 \rangle$ . . .

```
newref r in
    wait(2)  ||  wait(1)
   !r        ||  r := true        0
    wait(1)  ||  wait(2)
```

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ *then* $M \Downarrow^t v$.

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t\rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1\rangle$ . . .

newref r in

wait(1)	wait(0)
!r	r := true
wait(1)	wait(2)

1

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ then $M \Downarrow^t v$.

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t\rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1\rangle$ . . .

```
newref r in
    wait(1) || wait(0)
   !r       || r := true        1
    wait(1) || wait(2)
```

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ *then* $M \Downarrow^t v$*.*

Adequacy (Time)

An efficient small-step semantics :

$$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t \rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1 \rangle \qquad \ldots$$

```
newref r in
   wait(1)  ||  wait(0)
 !r         ||  r := true
   wait(1)  ||  wait(2)
```

1

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ *then* $M \Downarrow^t v$.

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t\rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1\rangle$ . . .

newref r in

wait(0)	wait(0)
!r	r := true
wait(1)	wait(1)

2

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ then $M \Downarrow^t v$.

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t\rangle \to \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1\rangle$ . . .

newref r in

wait(0)	wait(0)
!r	r := true
wait(1)	wait(1)

2

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ *then* $M \Downarrow^t v$*.*

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t\rangle \rightarrow \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1\rangle$ . . .

```
newref r in
   wait(0) || wait(0)
   !r      || r := true
   wait(1) || wait(1)
```

2

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ then $M \Downarrow^t v$.

Adequacy (Time)

An efficient small-step semantics :

$\langle \mathsf{wait}(\alpha_1 + \alpha_2), s, t \rangle \rightarrow \langle \mathsf{wait}(\alpha_2), s, t + \alpha_1 \rangle$ . . .

newref r in

wait(0)	wait(0)
!r	r := true
wait(0)	wait(0)

3

Theorem

If $q^x \multimap v^{x+t} \in [\![M]\!]$ then $M \Downarrow^t v$.

Conclusion

- Concurrent games with annotations:
  - $\rightarrow$ A sound model for $\mathcal{R}$-IPA,
  - $\rightarrow$ Adequate for $\mathbb{R}$-IPA.

- Future work:
  - Replication;
  - Non-idempotent resources?
  - Resources affecting the control-flow?

Conclusion

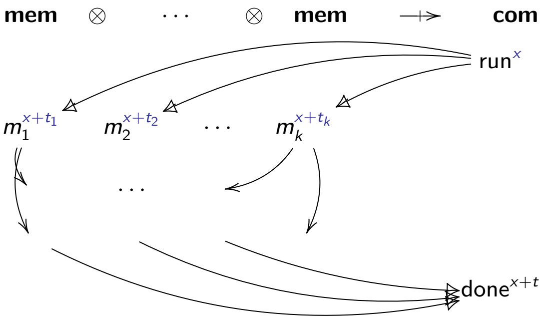
- Concurrent games with annotations:
  - → A sound model for $\mathcal{R}$-IPA,
  - → Adequate for $\mathbb{R}$-IPA.

- Future work:
  - Replication;
  - Non-idempotent resources?
  - Resources affecting the control-flow?

Thank you!

Proof sketch

A witness for $q^x \multimap \mathrm{done}^{x+t} \in [\![M]\!]$

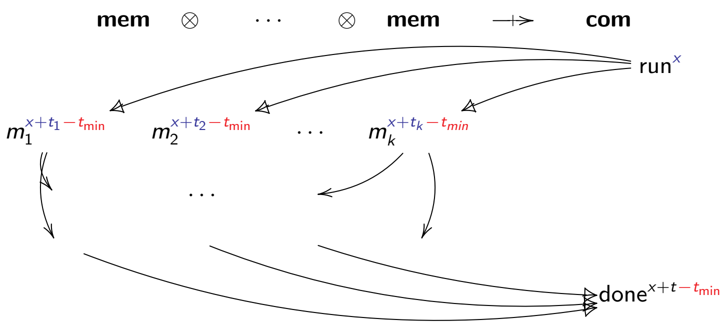


can be read as

1. If $t_i = 0$ then M can perform m_i at no cost or other memory operations.
2. Else M can reduce in high parallelism to case 1.

Proof sketch

A witness for $q^x \multimap \text{done}^{x+t} \in [\![M]\!]$

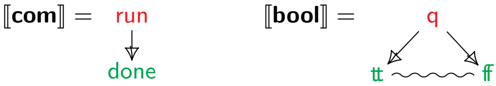


can be read as

1. If $t_i = 0$ then M can perform m_i at no cost or other memory operations.
2. Else M can reduce in high parallelism to case 1.

Concurrent games [CC16]

- Types as games



Definitions

An arena $(|A|, \leq_A, \#_A, \mathrm{pol}_A)$ is an event structure with polarity:

- $(|A|, \leq_A)$ a causal relation (**partial order**, with finite histories $[a]$),
- $\#_A$ a binary conflict relation (up-closed),
- $\mathrm{pol}_A : A \to \{-, +\}$,

that is

- **negative**: $\mathrm{pol}(\min(A)) = \{-\}$
- **well-threaded**: for all $a \in A$, $\min([a])$ is unic.

$\mathscr{C}(A)$ is the set of **configurations**: down-closed compatible subsets of A.

Concurrent arenas [CC16]

Constructions on arenas.

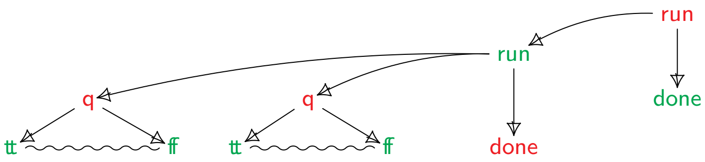
- If A is a arena, $A^\perp$ has the same structure with polarity inverted.
- If A, B are arenas, $A \otimes B$ has events $|A| + |B|$, and components inherited.

Linear map.

- If A, B are arena and B is rooted, $A \multimap B$ is $A^\perp \leftarrow B$,

  i.e. $A^\perp \otimes B$ with extra relation $\min(B) \leq a$ for all $a \in |A|$.

$$[\![\ (\mathbf{bool} \multimap \mathbf{bool} \multimap \mathbf{com}) \multimap \mathbf{com} \ ]\!]$$



Annotated concurrent games

Interaction (plays) For $q_A : A$, $q_{A^\perp} : A^\perp$

$$q_{A^\perp} \circledast q_A = (q_A, \leq_A) \bigwedge (q_{A^\perp, \leq_{A^\perp}})$$

$$\lambda_{q_{A^\perp} \circledast q_A}(a) = \lambda_{q_A}(a) \langle \lambda_{q_{A^\perp} \circledast q_A}(e) \rangle_{e \in [e]^-_{q_A}}$$

strategies

$$\tau \circledast \sigma = \{ q_\tau \circledast q_\sigma \mid q_\tau \in A \otimes \tau, q_\sigma \in \sigma \otimes C \}$$

$$\tau \odot \sigma = \tau \circledast \sigma \downarrow V$$

What programs?

Concurrency (interleavings)

Replication

Parametrisation

(shared) Memory

Typed λ-calculus
(Higher order functions)

[Ghica05]
Slot games

[LMMP13]
Weighted relational model

What programs?

Concurrency (interleavings)

Replication

Parametrisation

(shared) Memory

Typed λ-calculus
(Higher order functions)

[Ghica05]
Slot games

[LMMP13]
Weighted relational model

What programs?

Concurrency (multi-core)

Replication

Parametrisation

(shared) Memory

Typed λ-calculus
(Higher order functions)

[Ghica05]
Slot games

[LMMP13]
Weighted relational model