Programming in the n-Synchronous Model with Lucy-n

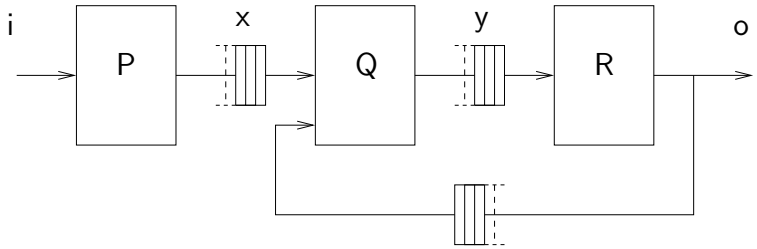
Louis Mandel Florence Plateau Marc Pouzet
{mandel,plateau,pouzet}@lri.fr

Laboratoire de Recherche en Informatique
Université Paris-Sud 11

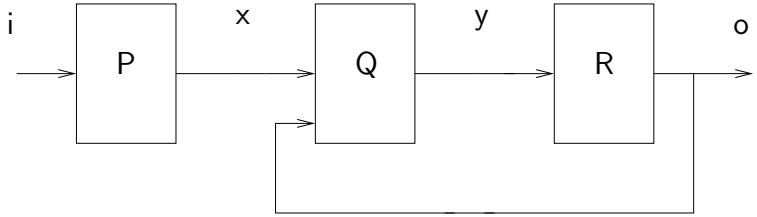
Partout 25 janvier 2010

Programming Kahn Networks

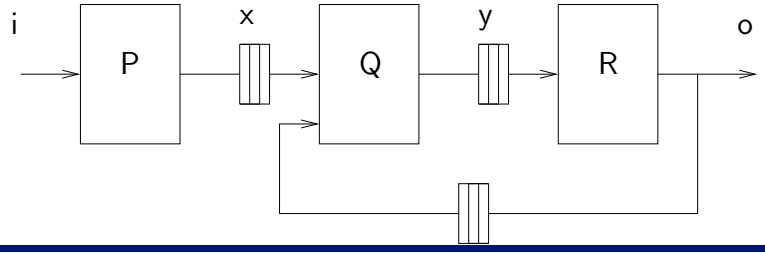
- Kahn networks (unbounded buffers):



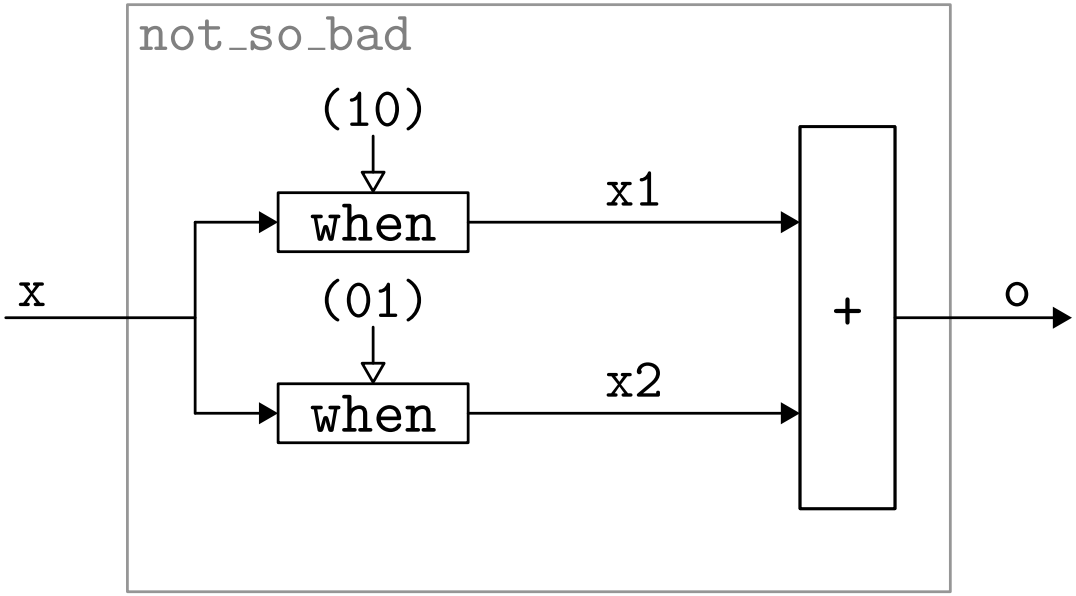
- Synchronous Kahn networks (no buffers):



- n-Synchronous Kahn networks (bounded buffers):

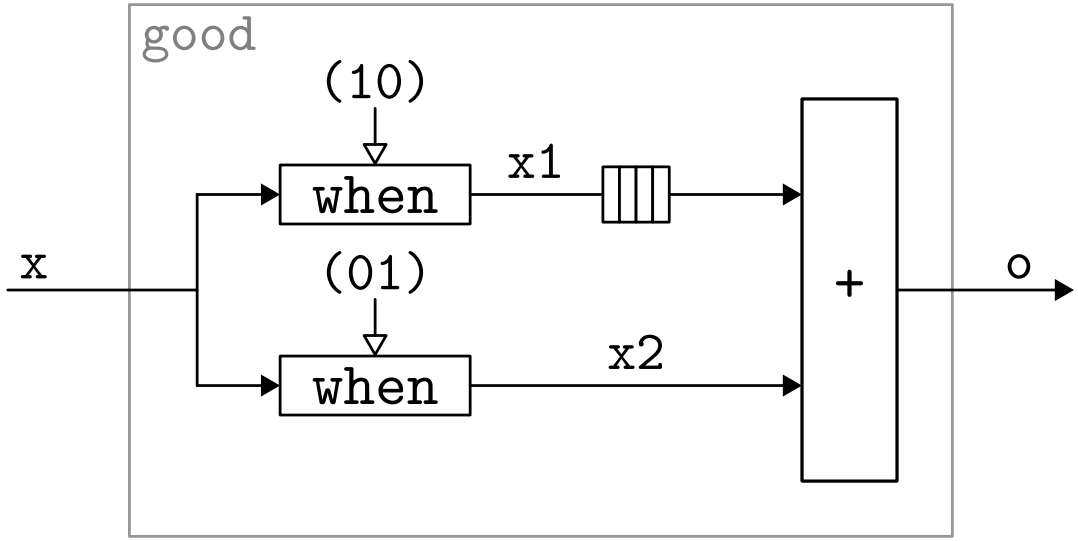


Synchronous Model



stream	values				clock
x1	5	3	2	...	(10)
x2	7	6	8	...	(01)

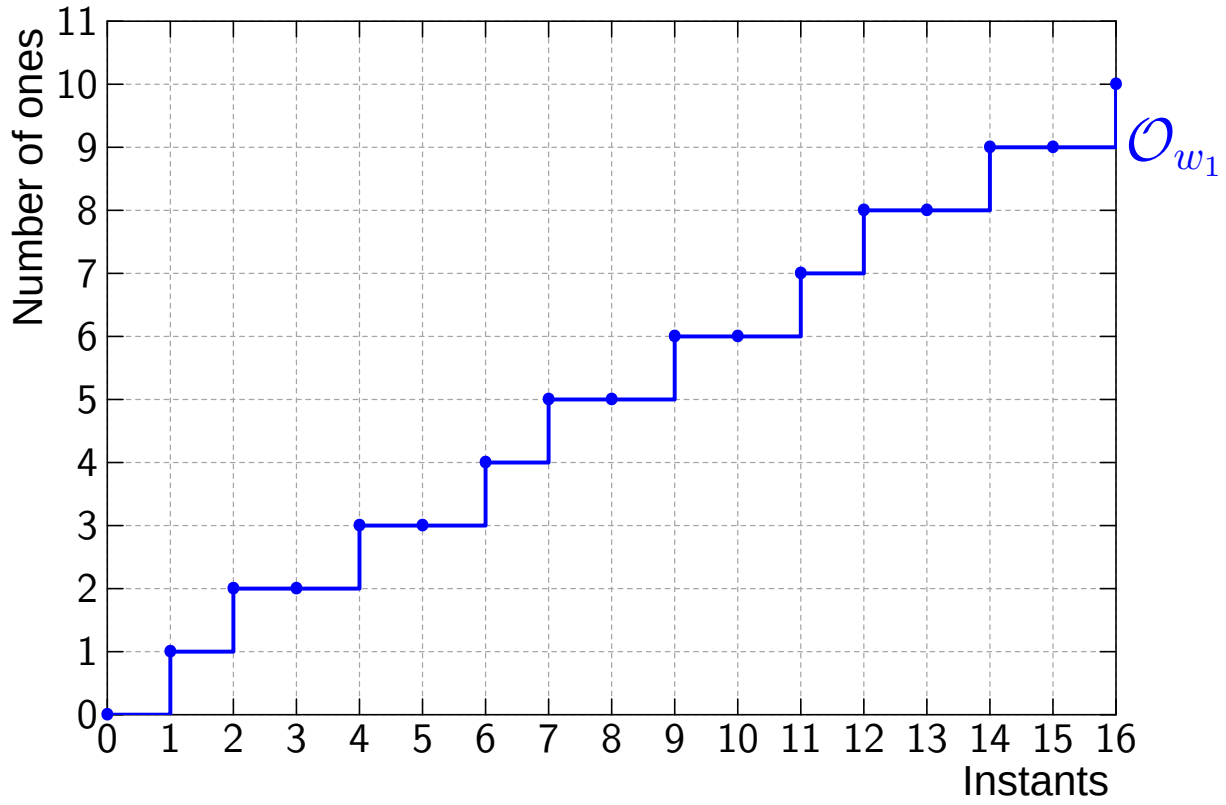
n-Synchronous Model



stream	values	clock
x1	5 3 2 ...	(10)
buffer(x1)	5 3 2 ...	(01)
x2	7 6 8 ...	(01)
buffer(x1) + x2	12 9 10 ...	(01)

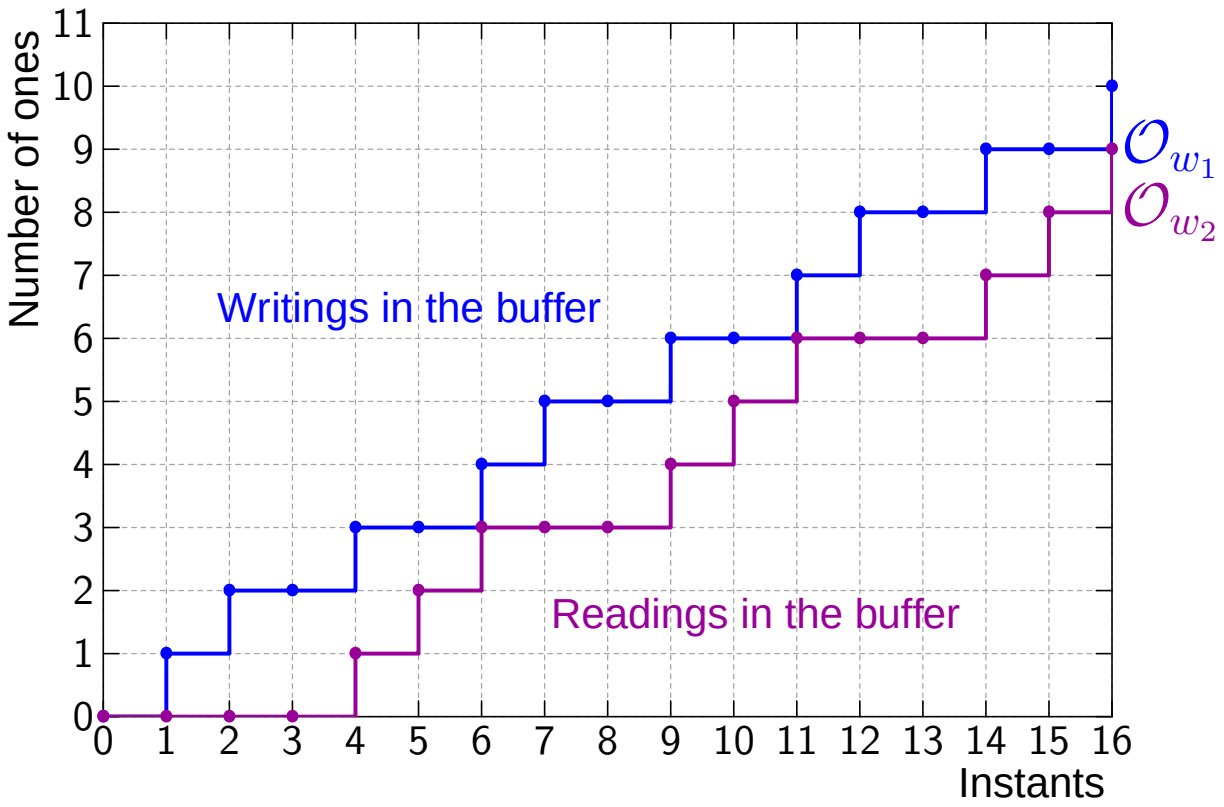
n-Synchronous Model

Clocks and Binary Words



$\mathcal{O}_w(i) =$ cumulative function of 1

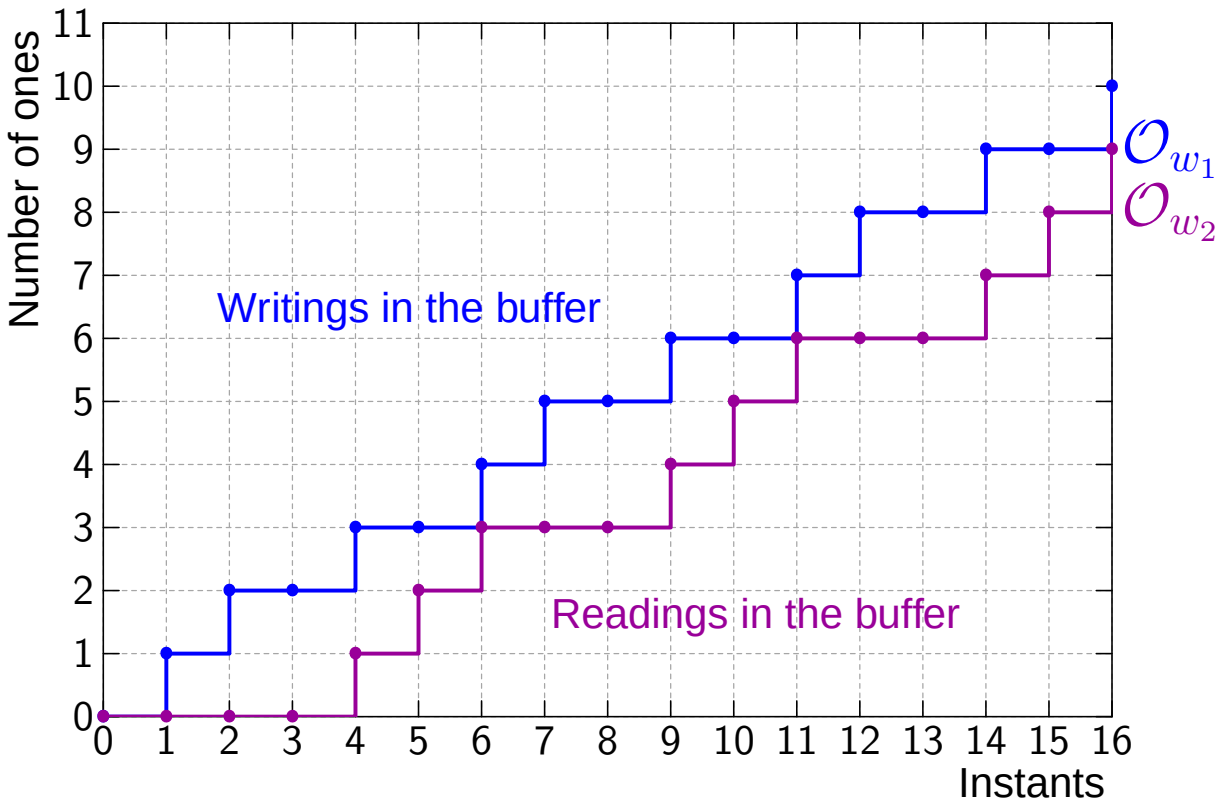
Clocks and Binary Words



buffer $size(w_1, w_2) = \max_{i \in \mathbb{N}}(\mathcal{O}_{w_1}(i) - \mathcal{O}_{w_2}(i))$

adaptability $w_1 <: w_2 \overset{def}{\Leftrightarrow} \exists n \in \mathbb{N}, \forall i,\ 0 \leq \mathcal{O}_{w_1}(i) - \mathcal{O}_{w_2}(i) \leq n$

Clocks and Binary Words

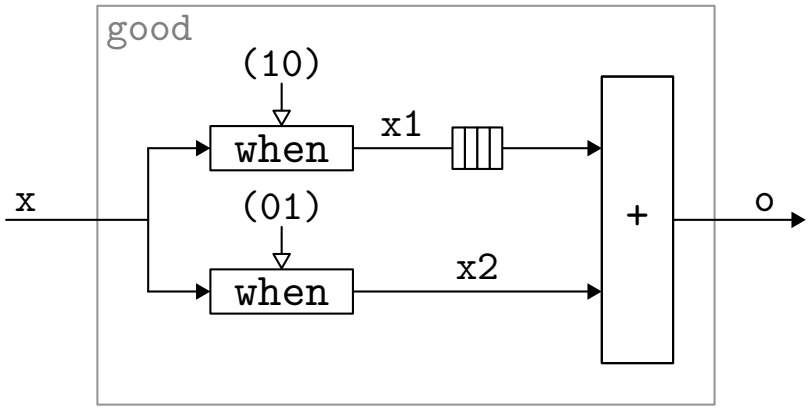


buffer	$size(w_1, w_2) = \max_{i \in \mathbb{N}}(\mathcal{O}_{w_1}(i) - \mathcal{O}_{w_2}(i))$
adaptability	$w_1 <: w_2 \overset{def}{\Leftrightarrow} \exists n \in \mathbb{N}, \forall i,\ 0 \leq \mathcal{O}_{w_1}(i) - \mathcal{O}_{w_2}(i) \leq n$
synchronizability	$w_1 \bowtie w_2 \overset{def}{\Leftrightarrow} \exists b_1, b_2 \in \mathbb{Z}, \forall i,\ b_1 \leq \mathcal{O}_{w_1}(i) - \mathcal{O}_{w_2}(i) \leq b_2$
precedence	$w_1 \preceq w_2 \overset{def}{\Leftrightarrow} \forall i,\ \mathcal{O}_{w_1}(i) \geq \mathcal{O}_{w_2}(i)$

LUCY-N: a n-Synchronous Language

The Lucy-n Language

```
1  let node good x = o where
2    rec x1 = x when (10)
3    and x2 = x when (01)
4    and o = buffer(x1) + x2
```




```
good :: forall 'a. 'a -> 'a on (01)
Buffer line 4, characters 11-21: size = 1
```

Clock Calculus

$$\frac{H \vdash e_1 : ck \mid C_1 \quad H \vdash e_2 : ck \mid C_2}{H \vdash e_1 + e_2 : ck \mid C_1 \cup C_2} \qquad \frac{H \vdash e : ck \mid C}{H \vdash \texttt{buffer}(e) : ck' \mid \{ck <: ck'\} \cup C}$$

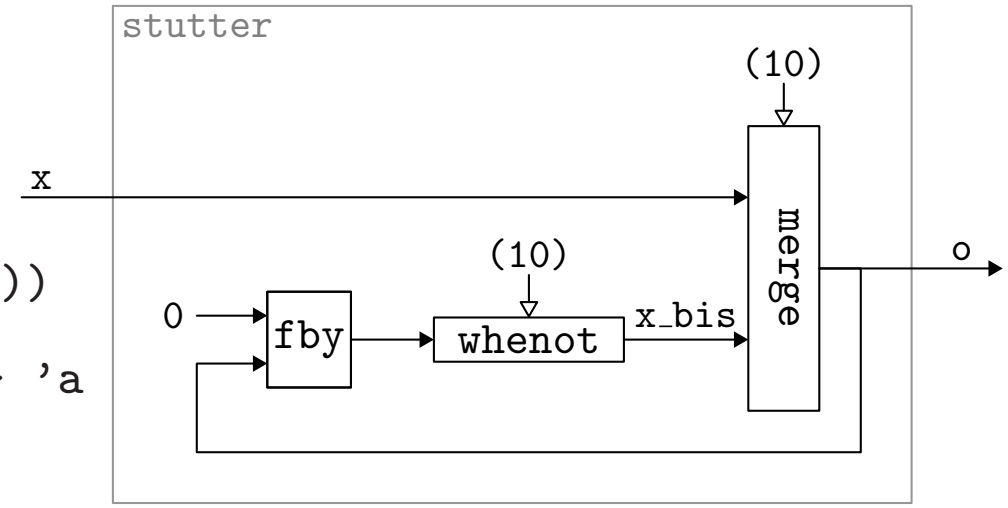
- Adaptability constraints are collected during the *clocking* of a node.
- They are solved at the end of node clocking.

Programming a stuttering node

- Synchronous version

```
let node stutter x = o where
  rec o = merge (10) x x_bis
  and x_bis = ((0 fby o) whenot (10))

stutter :: forall 'a. 'a on (10) -> 'a
```

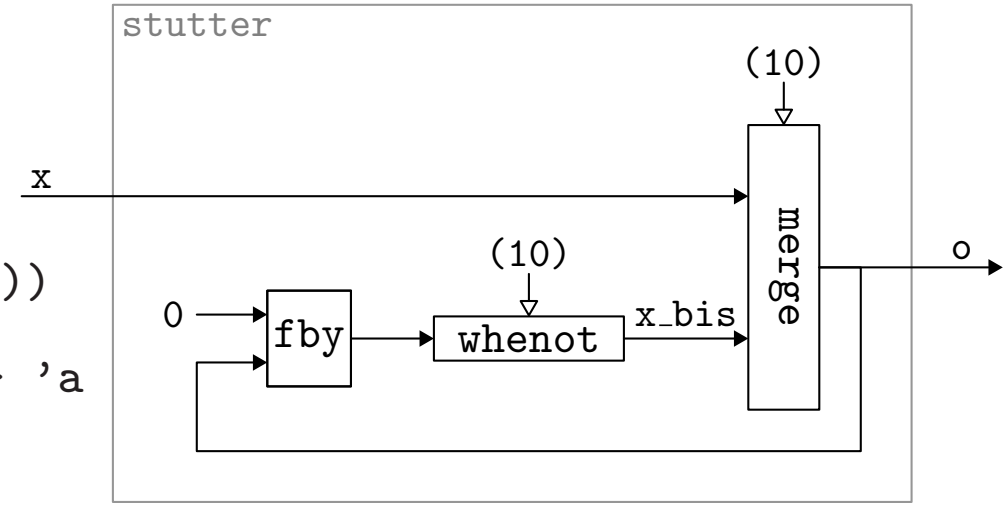


stream	values											clock
x	5		7		3		6		2		...	(10)
(10)	1	0	1	0	1	0	1	0	1	0	...	(1)
0 fby o	0	5	5	7	7	3	3	6	6	2	...	(1)
x_bis = (0 fby o) whenot (10)		5		7		3		6		2	...	*not* (10)
o = merge (10) x x_bis	5	5	7	7	3	3	6	6	2	2	...	(1)

Programming a stuttering node

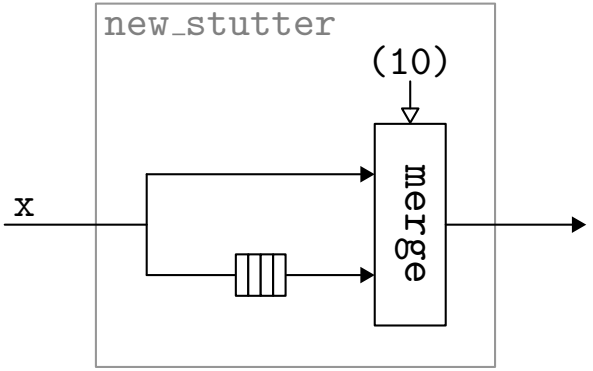
- Synchronous version

```
let node stutter x = o where
  rec o = merge (10) x x_bis
  and x_bis = ((0 fby o) whenot (10))
stutter :: forall 'a. 'a on (10) -> 'a
```



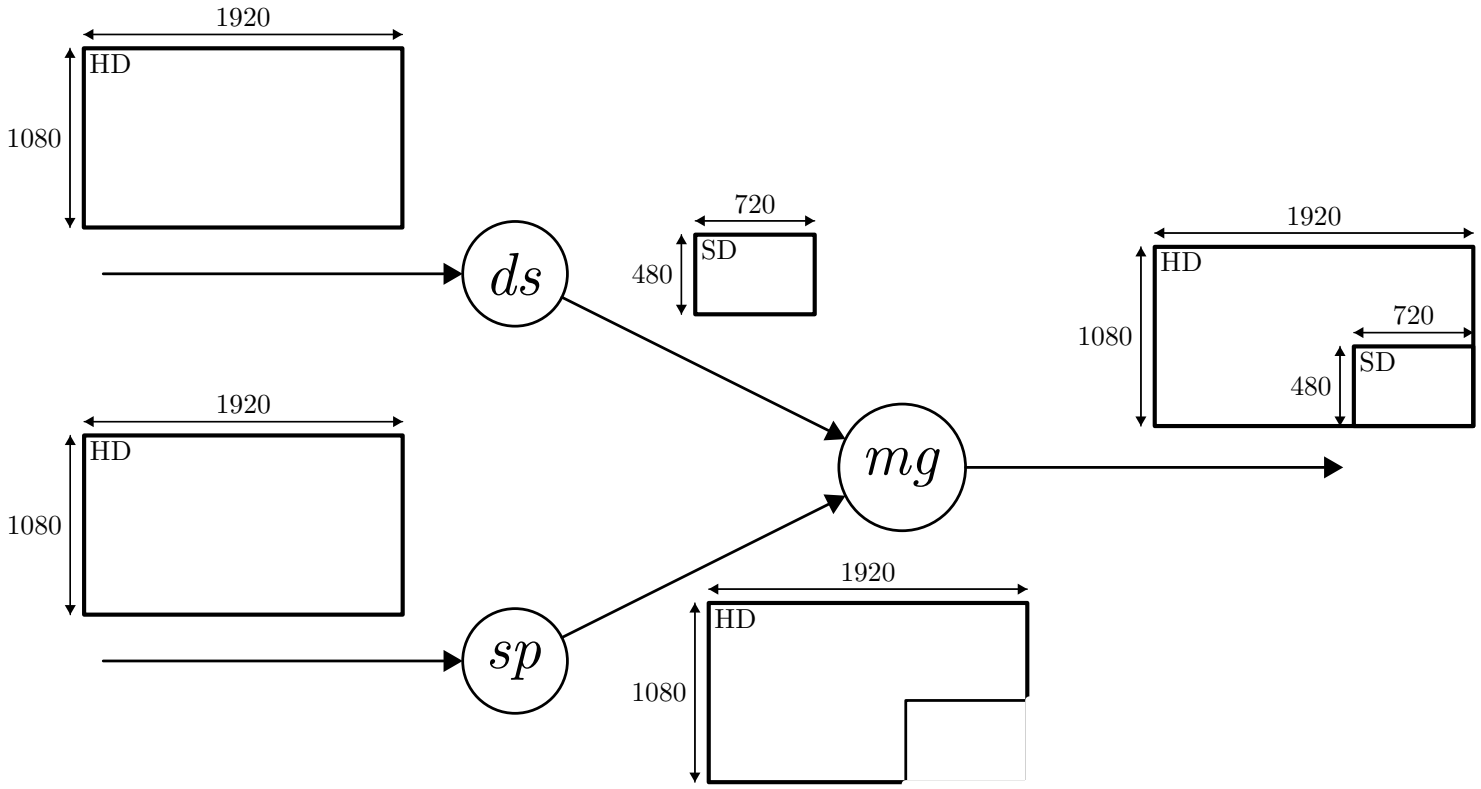
- n-Synchronous version

```
let node new_stutter x = o where
  rec o = merge (10) x (buffer(x))
new_stutter :: forall 'a. 'a on (10) -> 'a
Buffer line 2, characters 24-33: size = 1
```

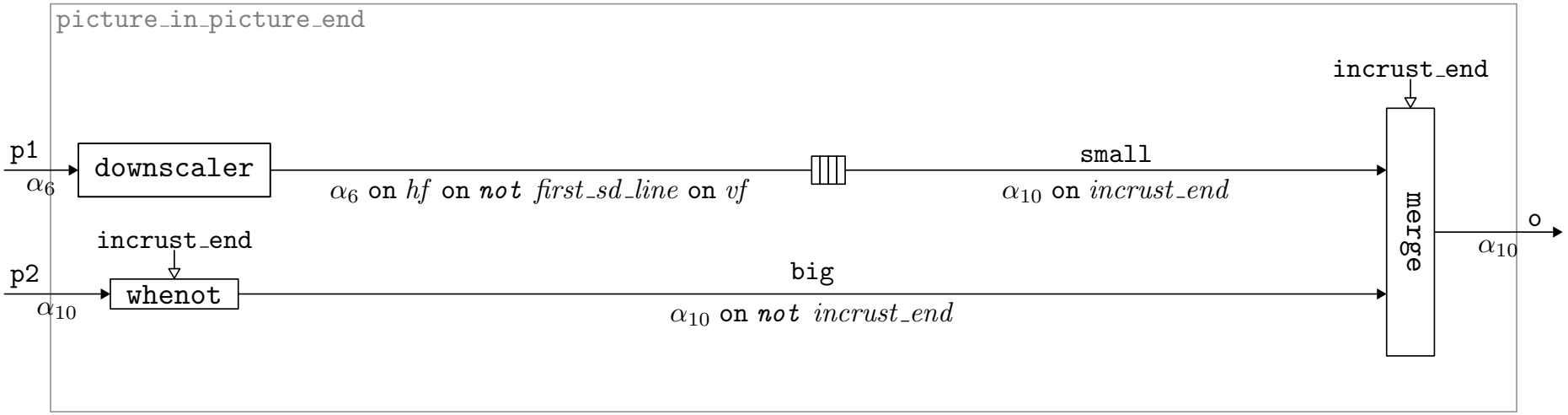


stream	values													clock
x	5		7		3		6		2		8		...	(10)
(10)	1	0	1	0	1	0	1	0	1	0	1	0	...	(1)
buffer(x)		5		7		3		6		2		8	...	*not* (10)
merge (10) x (buffer(x))	5	5	7	7	3	3	6	6	2	2	8	8	...	(1)

Video Application



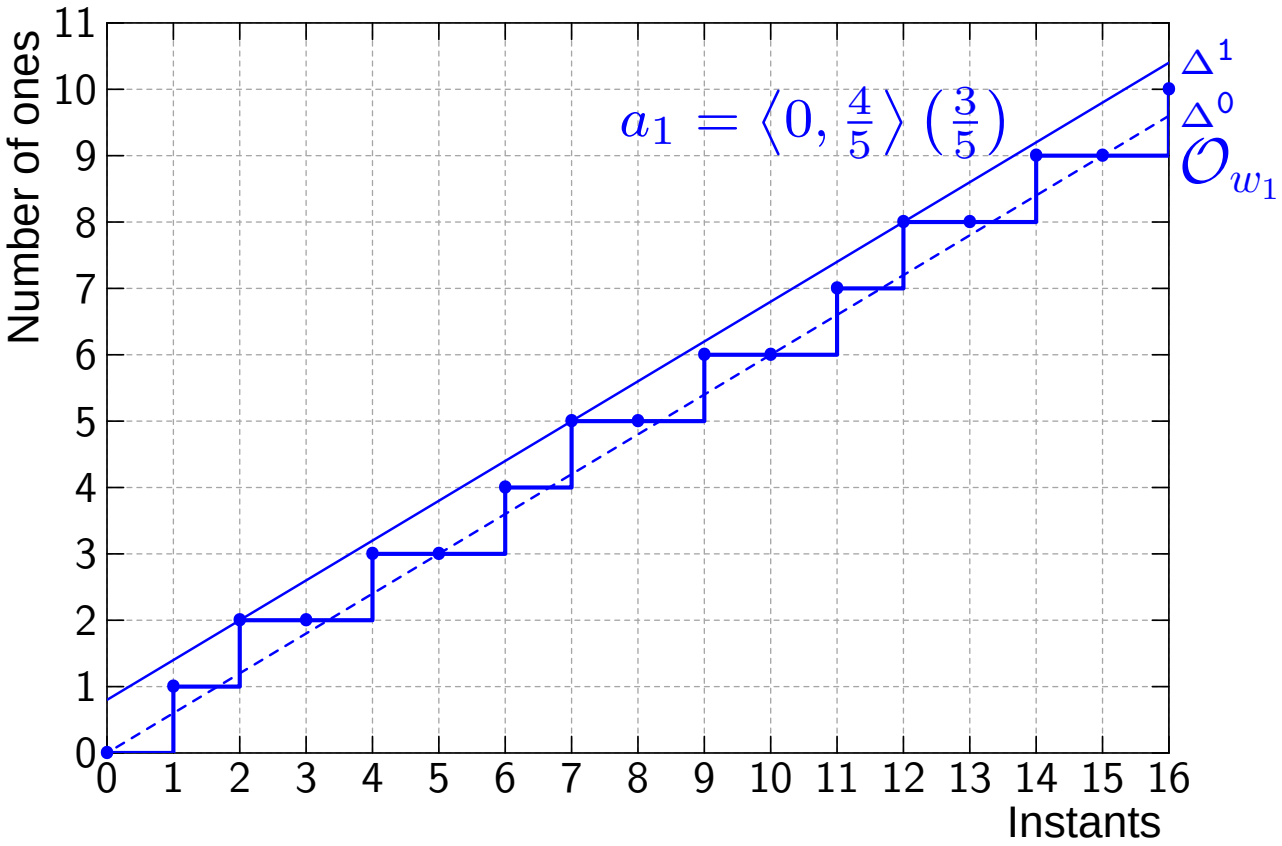
Video Application




```
51  (* picture in picture *)
52  let clock incrust_end =
53    (0^(1920 * (1080 - 480)) {0^1200 1^720}^480)
54
55  let node picture_in_picture_end (p1, p2) = o where
56    rec small = buffer(downscaler p1)
57    and big = (p2 whenot incrust_end)
58    and o = merge incrust_end small big
```

Abstraction

Abstract Clocks: $abs\,(w) = \langle b^0, b^1 \rangle\,(r)$

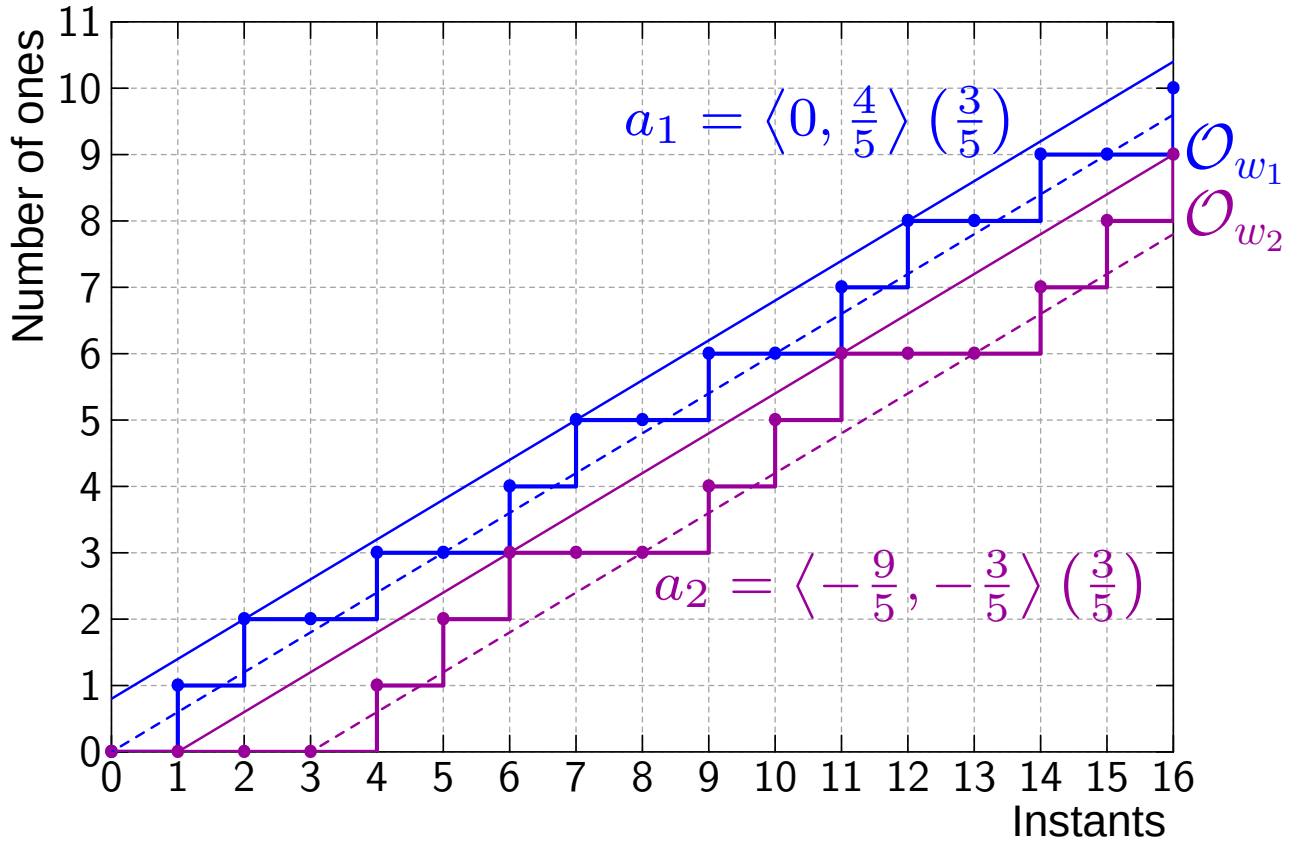


$$\Delta^1 : r \times i + b^1$$

$$\Delta^0 : r \times i + b^0$$

$$concr\left(\langle b^0, b^1 \rangle\,(r)\right) = \left\{ w \mid \begin{array}{lcl} w[i] = 1 & \Rightarrow & \mathcal{O}_w(i) \leq \Delta^1(i) \\ w[i] = 0 & \Rightarrow & \mathcal{O}_w(i) \geq \Delta^0(i) \end{array} \right\}$$

Abstract Relations



buffer	$size(a_1, a_2) = \lfloor {b^1}_1 - {b^0}_2 \rfloor$	
adaptability	$a_1 <:^{\sim} a_2$	$\overset{def}{\Leftrightarrow} \; a_1 \bowtie^{\sim} a_2 \wedge a_1 \preceq^{\sim} a_2$
synchronizability	$\langle {b^0}_1, {b^1}_1 \rangle (r_1) \bowtie^{\sim} \langle {b^0}_2, {b^1}_2 \rangle (r_2)$	$\overset{def}{\Leftrightarrow} \; r_1 = r_2$
precedence	$\langle {b^0}_1, {b^1}_1 \rangle (r) \preceq^{\sim} \langle {b^0}_2, {b^1}_2 \rangle (r)$	$\overset{def}{\Leftrightarrow} \; {b^1}_2 - {b^0}_1 < 1$

Abstraction of Clocks

Abstraction of clock expressions:

$$\begin{aligned} abs(\mathit{not}\ w) &= \mathit{not}^{\sim}\ abs(w) \\ abs(ce_1\ \mathit{on}\ ce_2) &= abs(ce_1)\ \mathit{on}^{\sim}\ abs(ce_2) \end{aligned}$$

Correctness property of operators:

$$\begin{aligned} \mathit{not}\ w &\in concr(\mathit{not}^{\sim}\ abs(w)) \\ ce_1\ \mathit{on}\ ce_2 &\in concr(abs(ce_1)\ \mathit{on}^{\sim}\ abs(ce_2)) \end{aligned}$$

Definition of abstract operators:

$$\mathit{not}^{\sim}\ \langle b^0, b^1 \rangle\,(r) \stackrel{def}{=} \langle -b^1, -b^0 \rangle\,(\ 1 - r)$$

$$\langle {b^0}_1, {b^1}_1 \rangle\,(r_1)\ \mathit{on}^{\sim}\ \langle {b^0}_2, {b^1}_2 \rangle\,(r_2) \stackrel{def}{=}$$

$$\langle {b^0}_1 \times r_2 + {b^0}_2, {b^1}_1 \times r_2 + {b^1}_2 \rangle\,(r_1 \times r_2) \text{ with } {b^0}_1 \leq 0 \text{ and } {b^0}_2 \leq 0$$

Constraints Solving

System to solve:

$$\{\alpha_6 \text{ on } hf \text{ on not } \mathit{first\_sd\_line} \text{ on } vf <: \alpha_{10} \text{ on } \mathit{incrust\_end}\}$$

1. Adaptability constraints

   $\alpha_6 \leftarrow \alpha \text{ on } c_6, \quad \alpha_{10} \leftarrow \alpha \text{ on } c_{10}$

   $\{c_6 \ \mathit{on} \ hf \ \mathit{on} \ \mathit{not} \ \mathit{first\_sd\_line} \ \mathit{on} \ vf <: c_{10} \ \mathit{on} \ \mathit{incrust\_end}\}$

2. Abstraction of constraints

   $\{abs\,(c_6) \ \mathit{on}^{\sim} \ abs\,(hf) \ \mathit{on}^{\sim} \ \mathit{not}^{\sim} \ abs\,(\mathit{first\_sd\_line}) \ \mathit{on}^{\sim} \ abs\,(vf)$
   $<:^{\sim} abs\,(c_{10}) \ \mathit{on}^{\sim} \ abs\,(\mathit{incrust\_end})\}$

   i.e. $\{abs\,(c_6) \ \mathit{on}^{\sim} \ \langle -720, \frac{481}{3} \rangle \left(\frac{1}{6}\right) <:^{\sim} abs\,(c_{10}) \ \mathit{on}^{\sim} \ \langle -192200, 0 \rangle \left(\frac{1}{6}\right)\}$

3. System of linear inequations

4. Solution: $c_6 = 0^{4315}(1), \quad c_{10} = (1)$

   $$\mathit{picture\_in\_picture\_end} :: \forall\alpha.\,(\alpha \times \alpha \text{ on } 0^{4315}(1)) \rightarrow \alpha \text{ on } 0^{4315}(1)$$

Video Application

	delay	buffer size
minimal result	$1\,920$ ($\approx$ 1 HD line)	$191\,970$ (≈ 266.6 SD lines)
abstract result	$4\,315$ ($\approx$ 2 HD lines)	$193\,079$ (≈ 268.1 SD lines)

Conclusion

http://www.lri.fr/~plateau/these