# Hyperstream Processing Systems

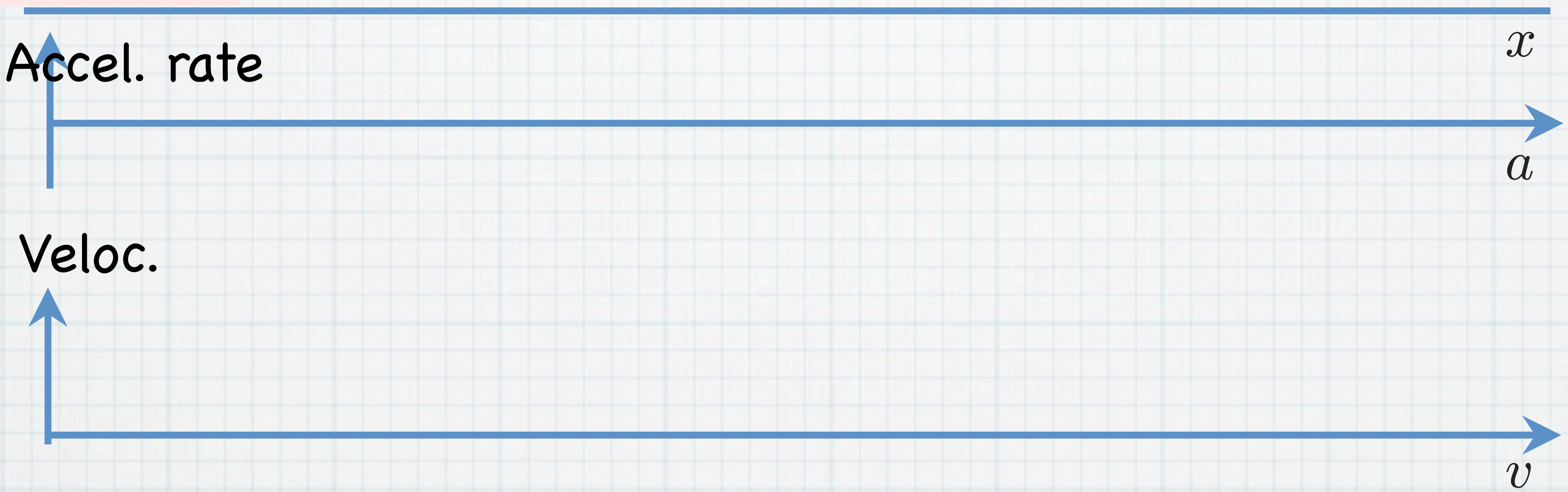
## Nonstandard Modeling of Continuous-Time Signals

Kohei Suenaga
Hakubi Project
Kyoto University (JP)

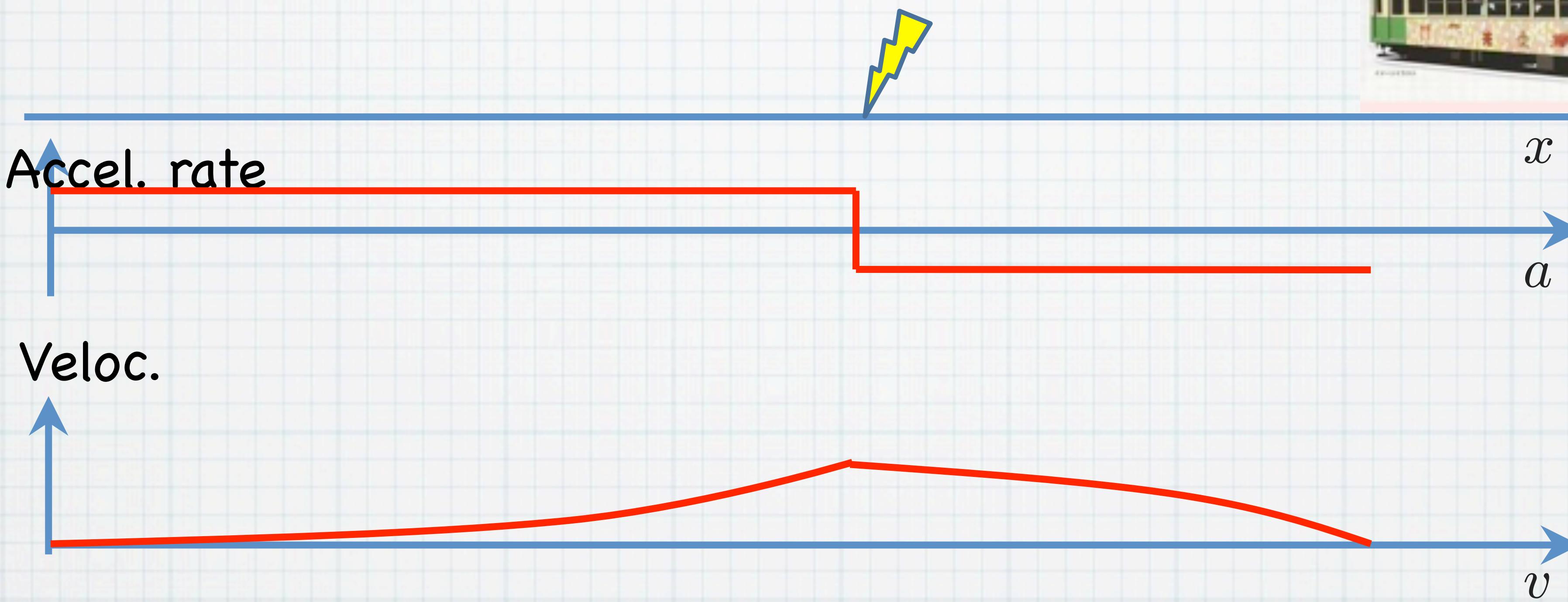
Hiroyoshi Sekine, Ichiro Hasuo
Dept. Computer Science
University of Tokyo (JP)

POPL 2013, Rome

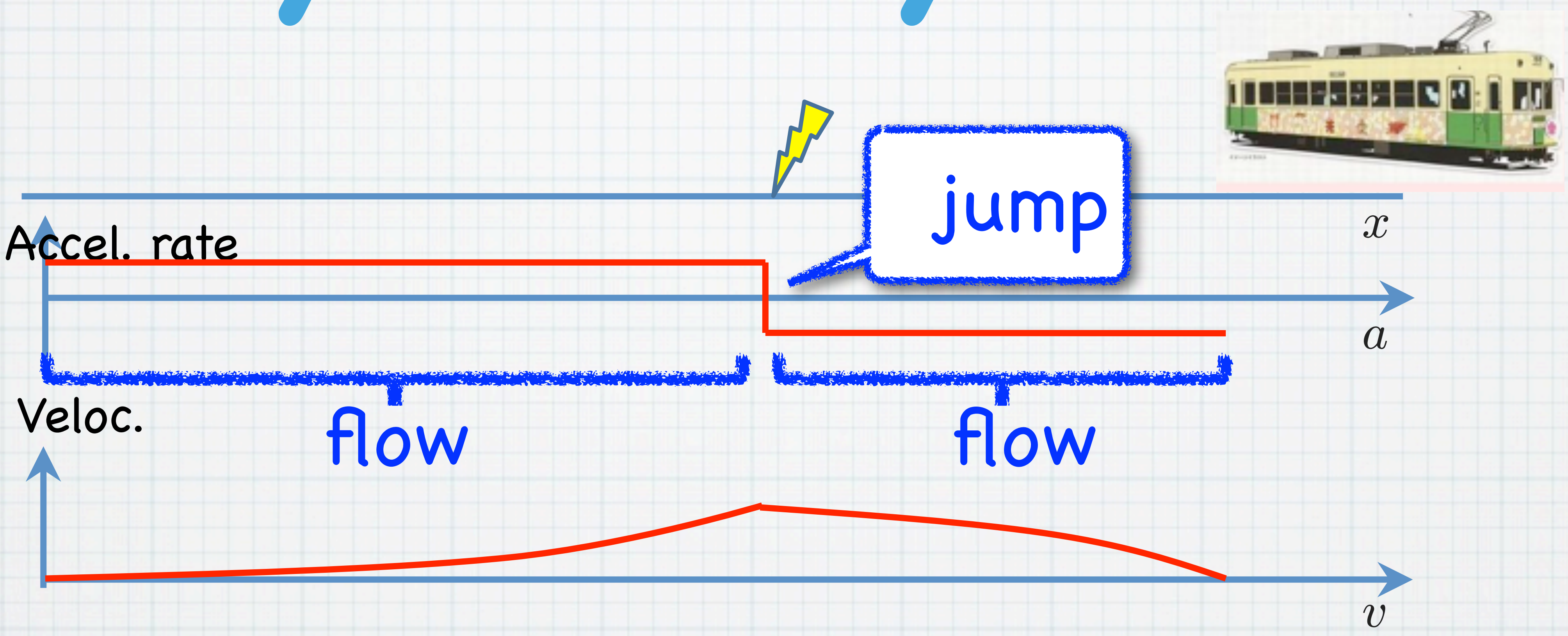
# Hybrid System



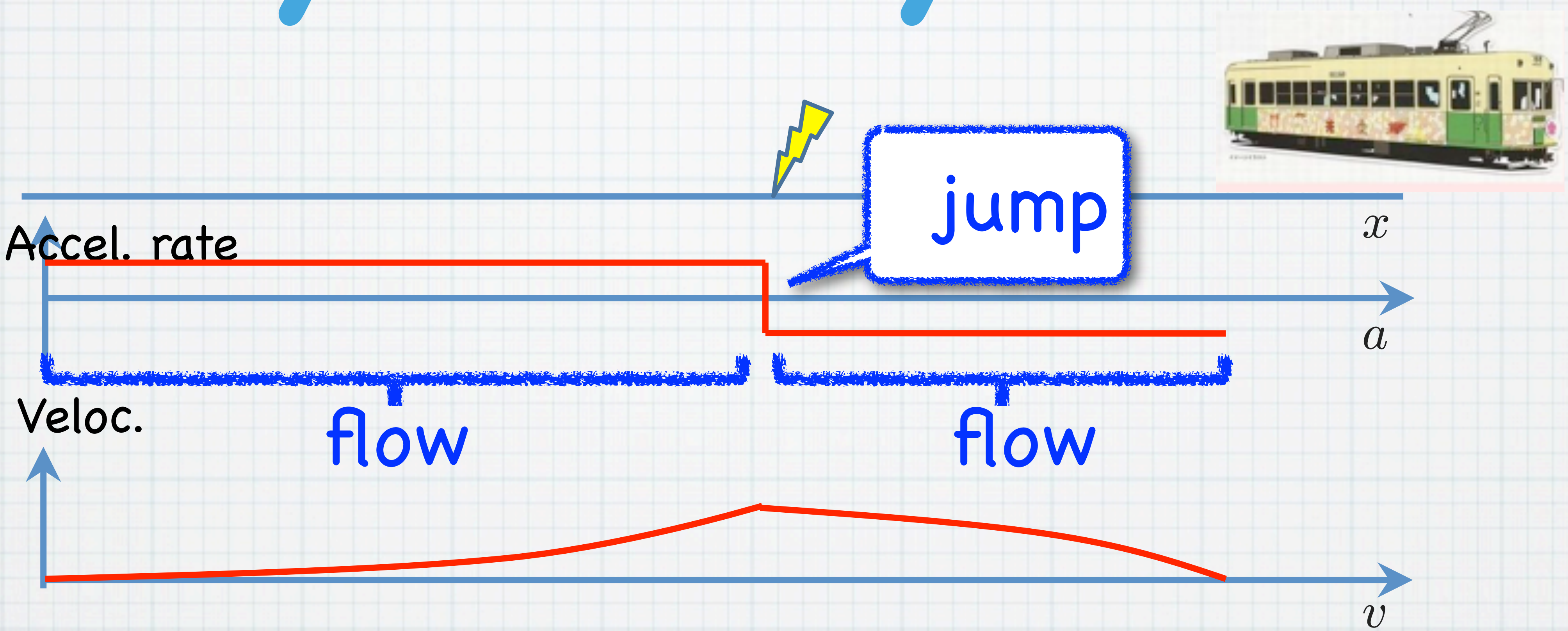
# Hybrid System



# Hybrid System



# Hybrid System




* **Flow & jump**
  * Digital control in a physical environment
  * Component of cyber-physical systems

# Hybrid System

Discrete
"**jump**"

and

Continuous
"**flow**"

# Hybrid System

Discrete
"**jump**"

and

Continuous
"**flow**"

# Hybrid System

**Formal verification**
(computer science)

Discrete
"jump"

and

Continuous
"flow"

**Control theory**
(applied analysis)

# Hybrid System

**Formal verification**
(computer science)

Hybrid!

Discrete
"**jump**"

and

Continuous
"**flow**"

Hybrid!

**Control theory**
(applied analysis)

# Hybrid System

**Formal verification**
(computer science)

Hybrid!

- Flow?
- With minimal cost?

Discrete
"**jump**"

and

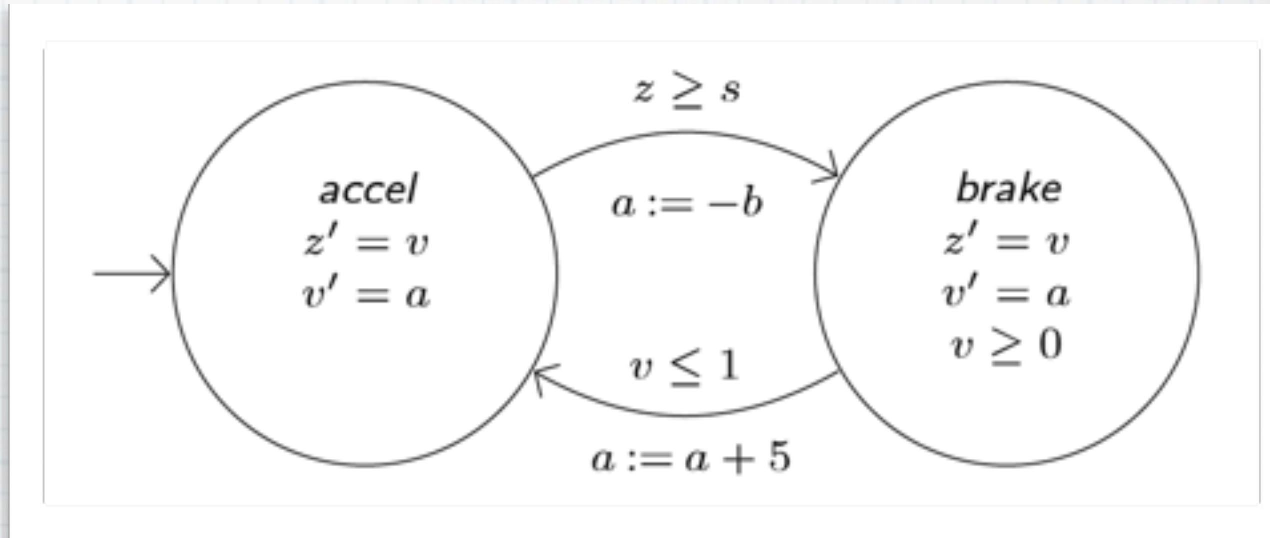
Continuous
"**flow**"

Hybrid!

**Control theory**
(applied analysis)

# Formal Verification Approaches

* Hybrid automata
  [Alur, Henzinger, …; '90s-]




* Differential dynamic logic
  [Platzer & others, '07-]

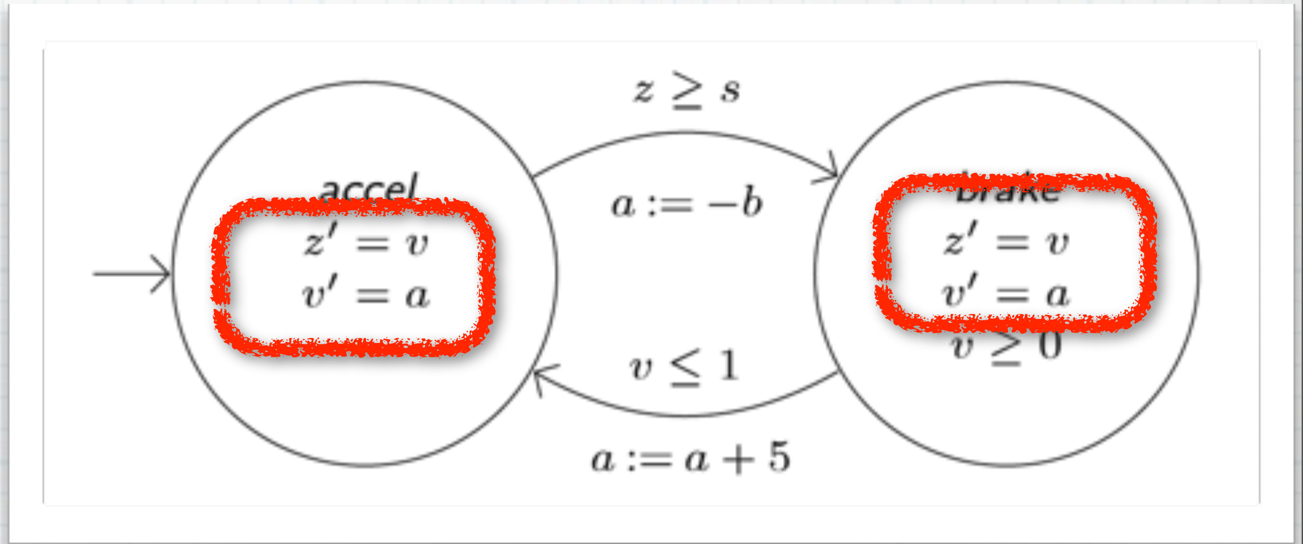
$$[\dot{x} = 1 \text{ while } x \leq 3]\varphi$$

* Differential equations, explicitly
  ➜ distinction jump vs. flow

# Formal Verification Approaches

* Hybrid automata

  [Alur, Henzinger, ...; '90s-]




* Differential dynamic logic

  [Platzer & others, '07-]

$$[\dot{x} = 1 \text{ while } x \leq 3]\varphi$$

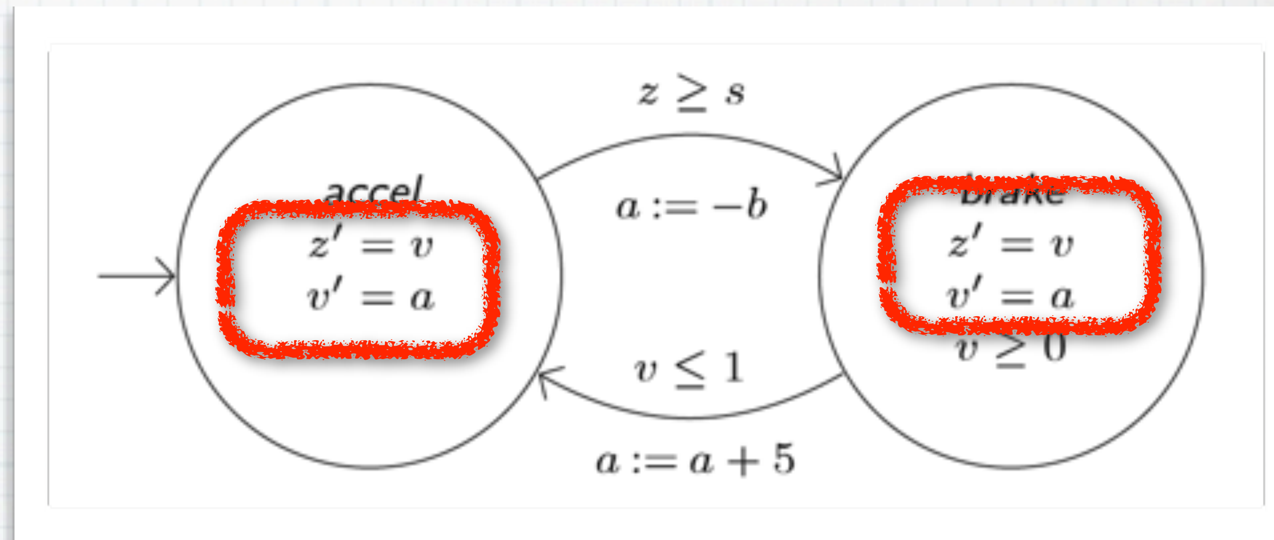
* Differential equations, explicitly

  ➜ distinction jump vs. flow

# Formal Verification Approaches

* Hybrid automata

  [Alur, Henzinger, ...; '90s-]




* Differential dynamic logic

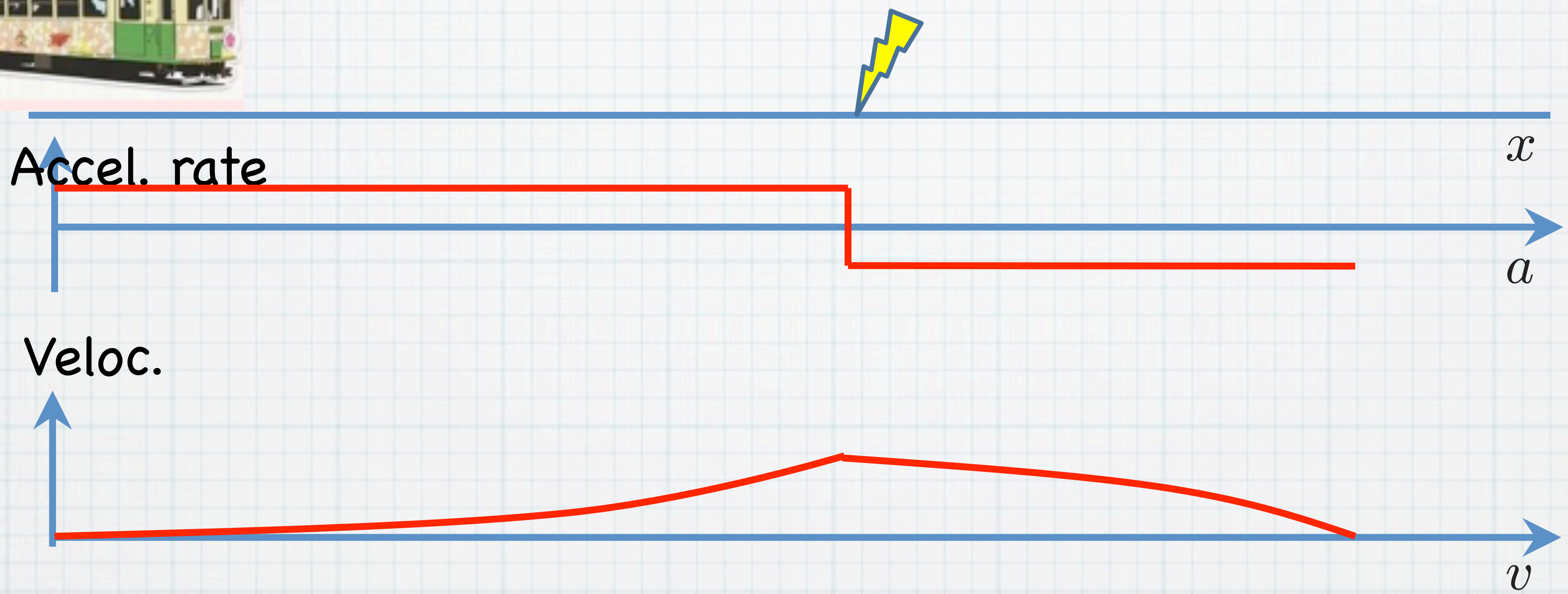
  [Platzer & others, '07-]

$$[\dot{x} = \mathbf{1} \text{ while } x \leq \mathbf{3}]\varphi$$

* Differential equations, explicitly

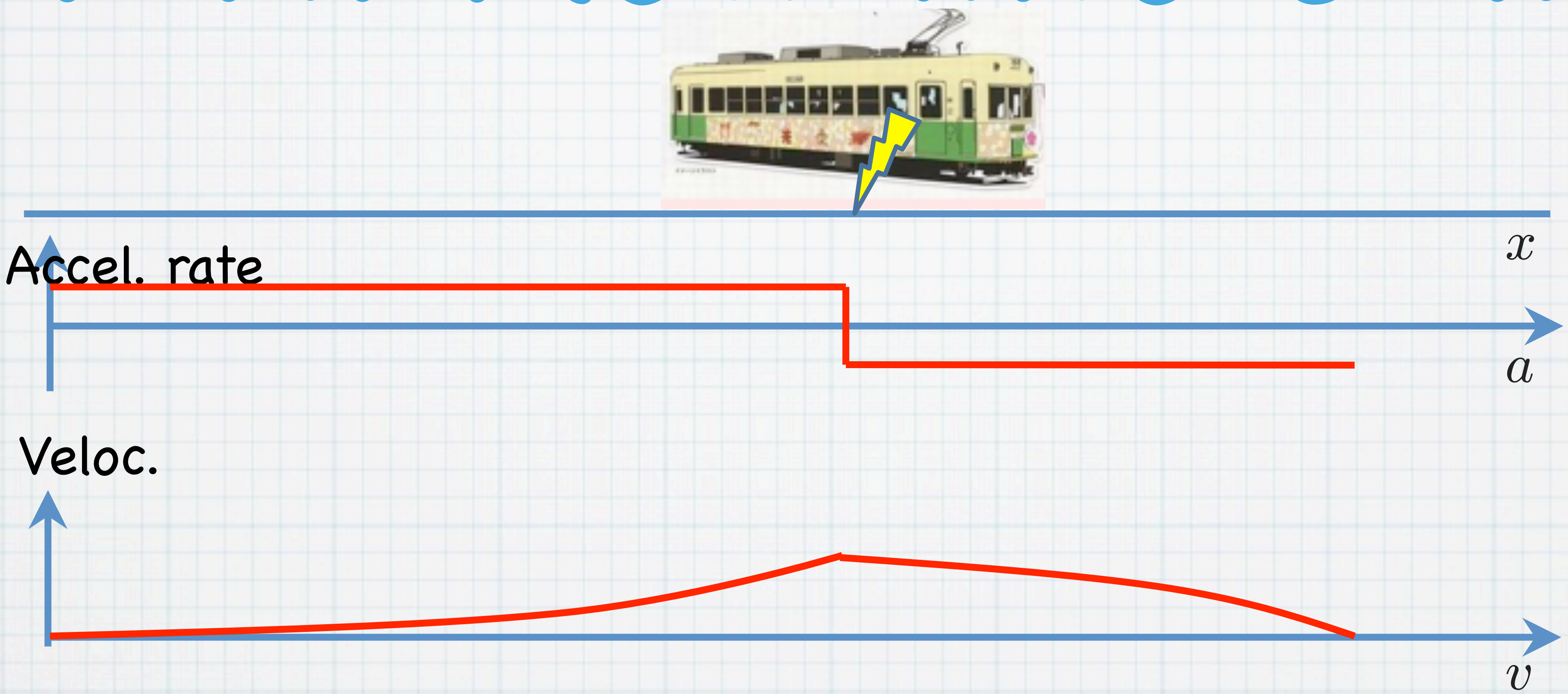
  ➜ distinction jump vs. flow

# “Turn Flow into Jump”



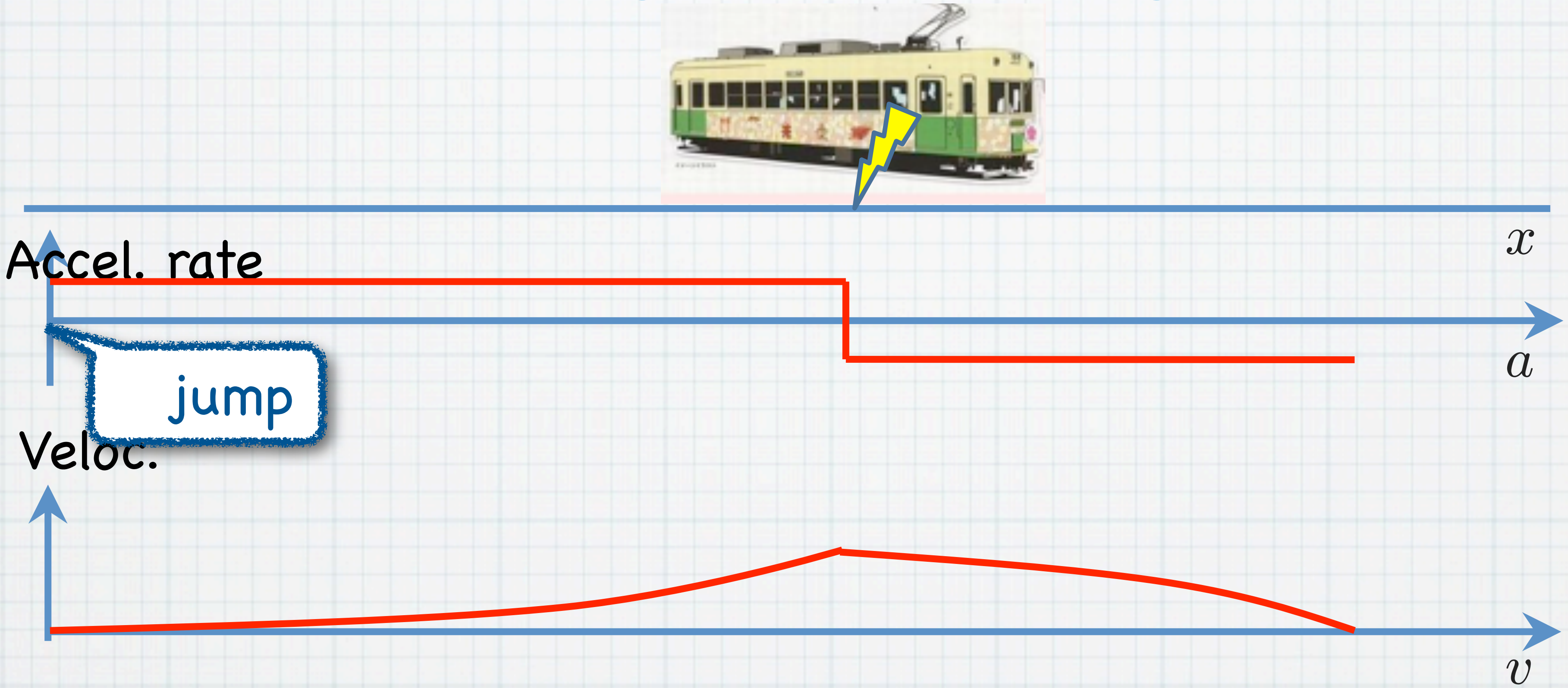

* Flow as infinitely many, infinitely small jumps

# "Turn Flow into Jump"



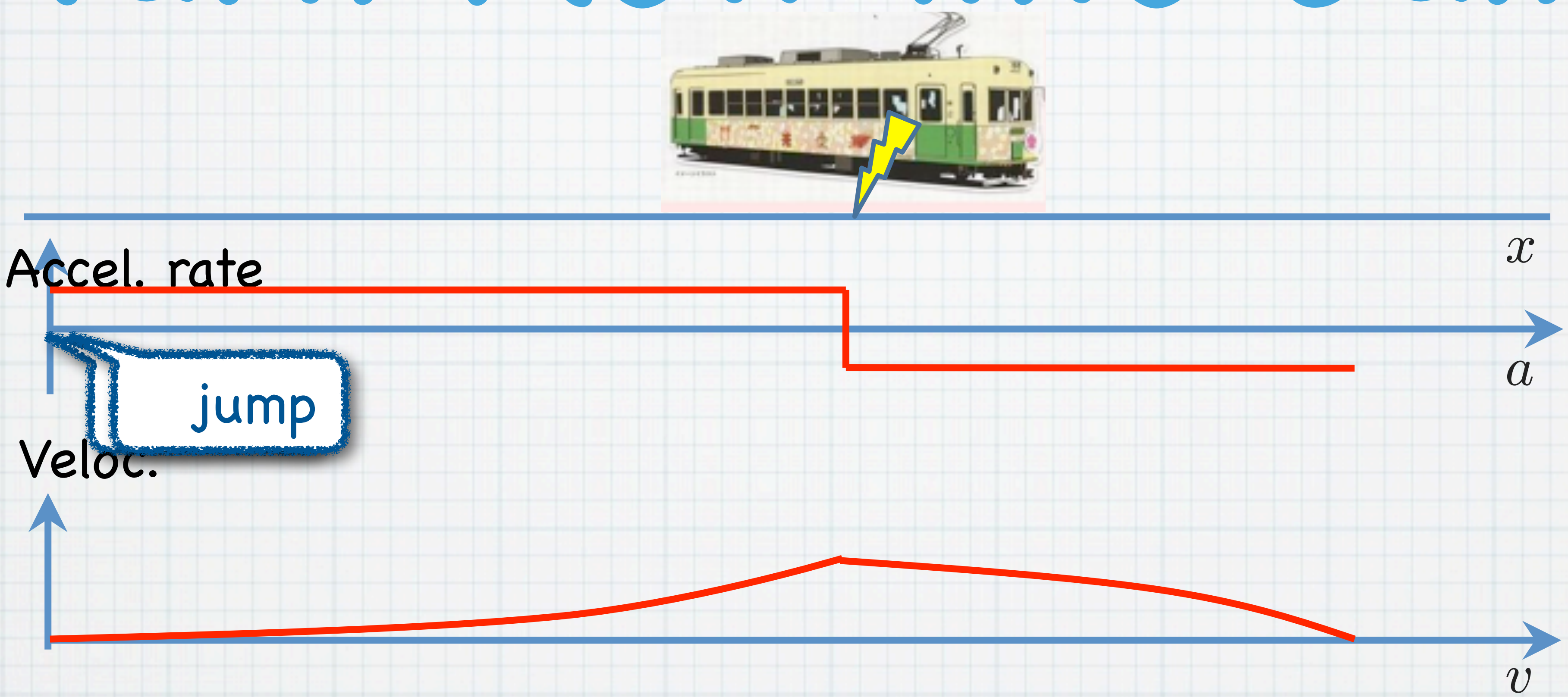

* Flow as infinitely many, infinitely small jumps

# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

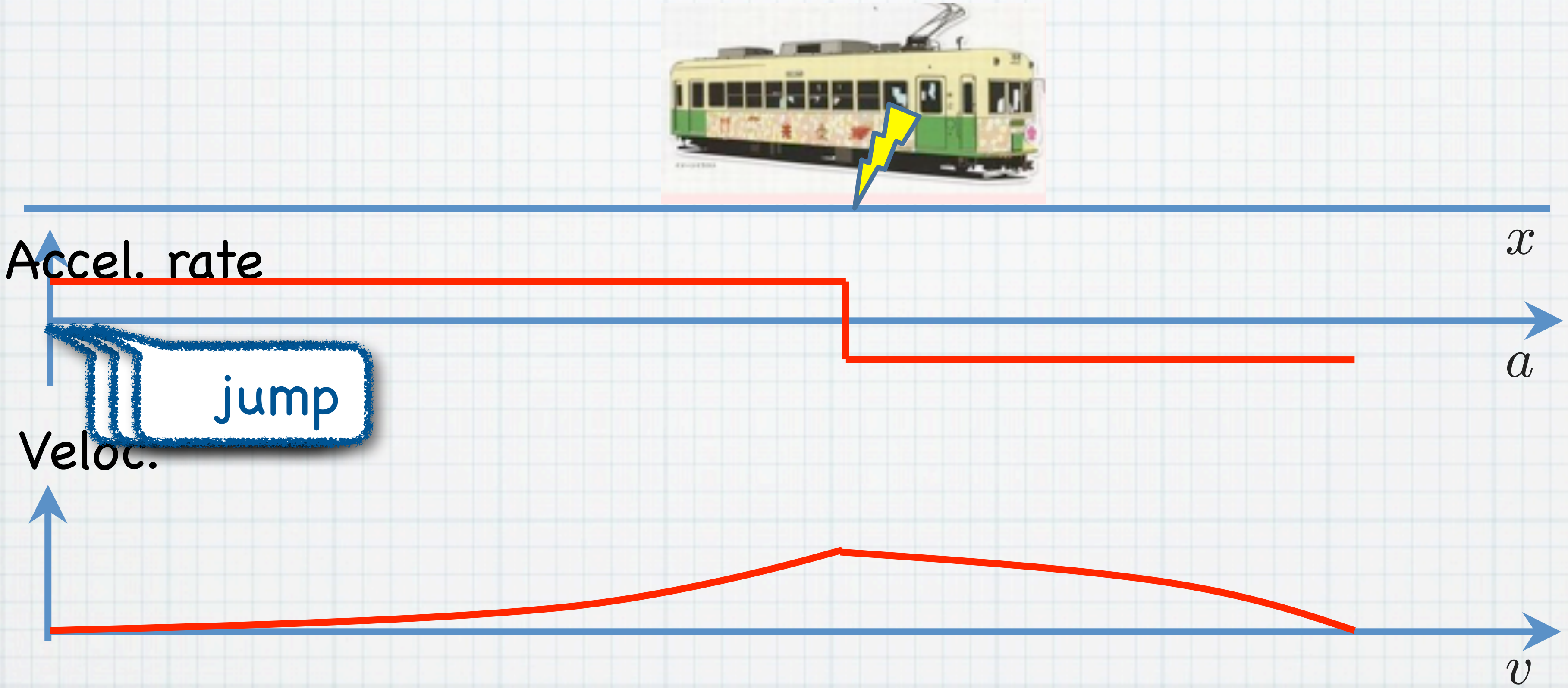
# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

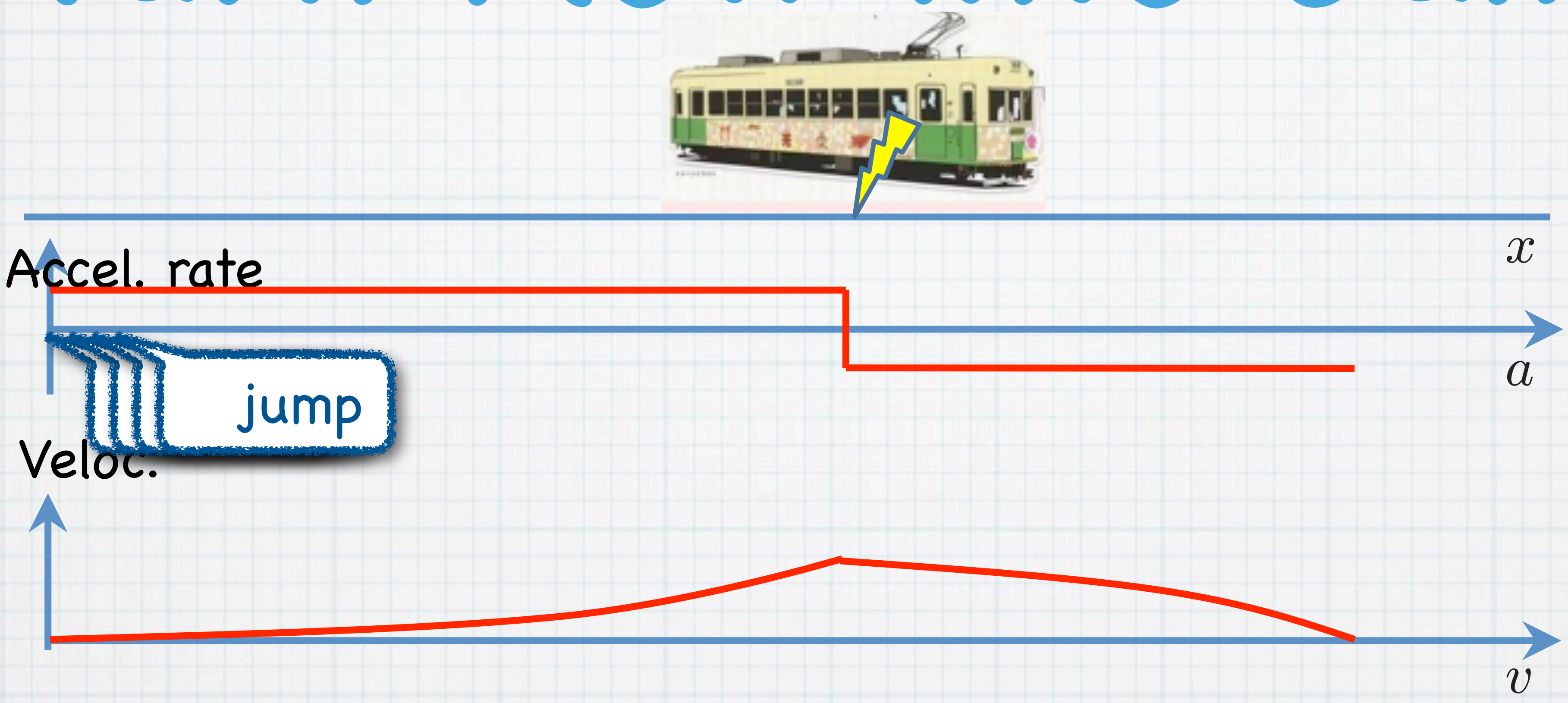
Hasuo (Tokyo)

# "Turn Flow into Jump"



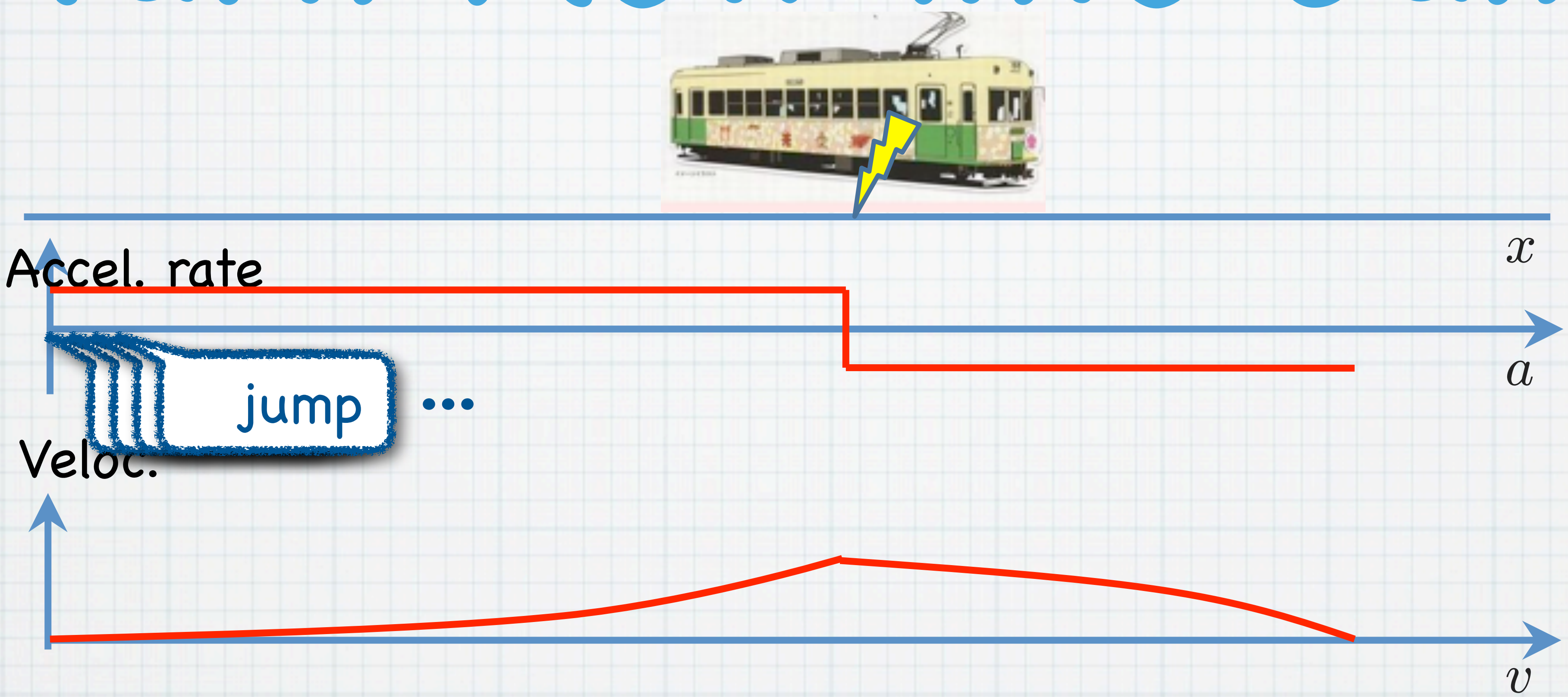

* Flow as infinitely many, infinitely small jumps

# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

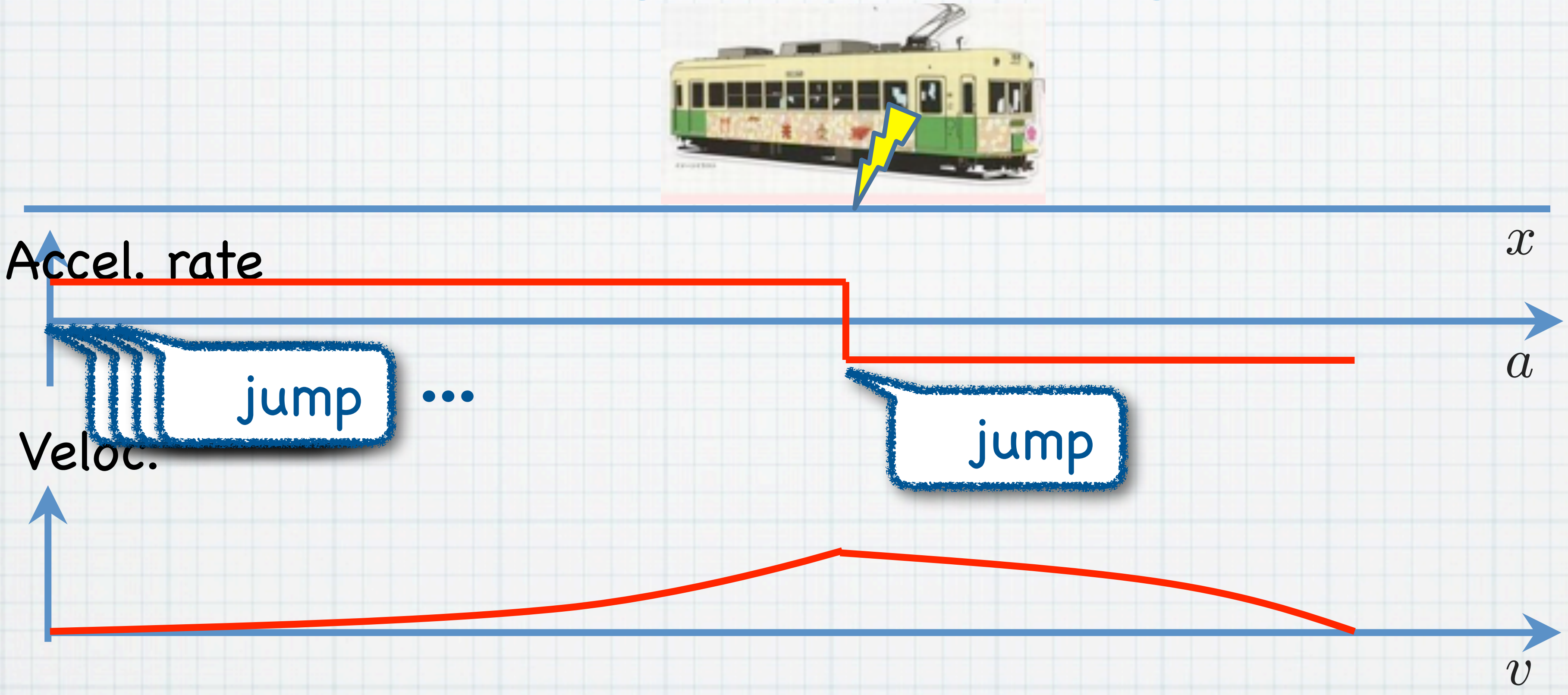
# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

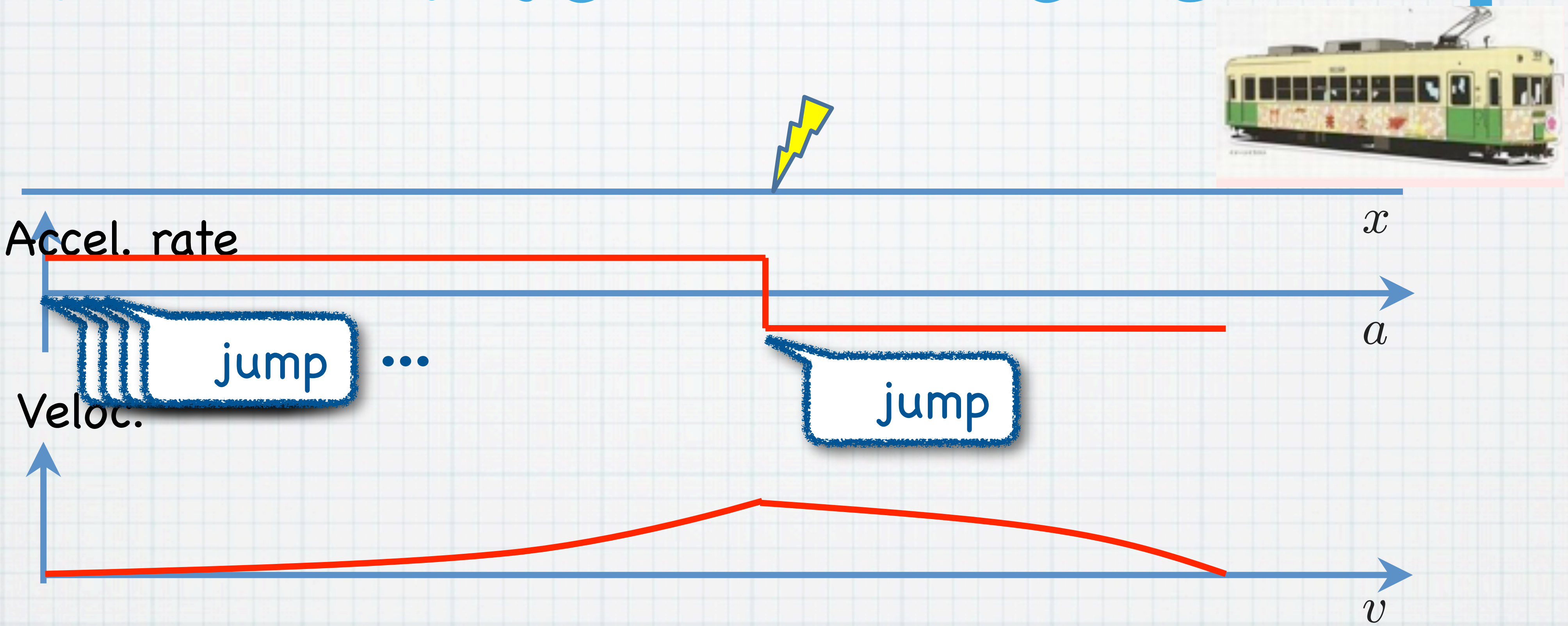
Hasuo (Tokyo)

# "Turn Flow into Jump"



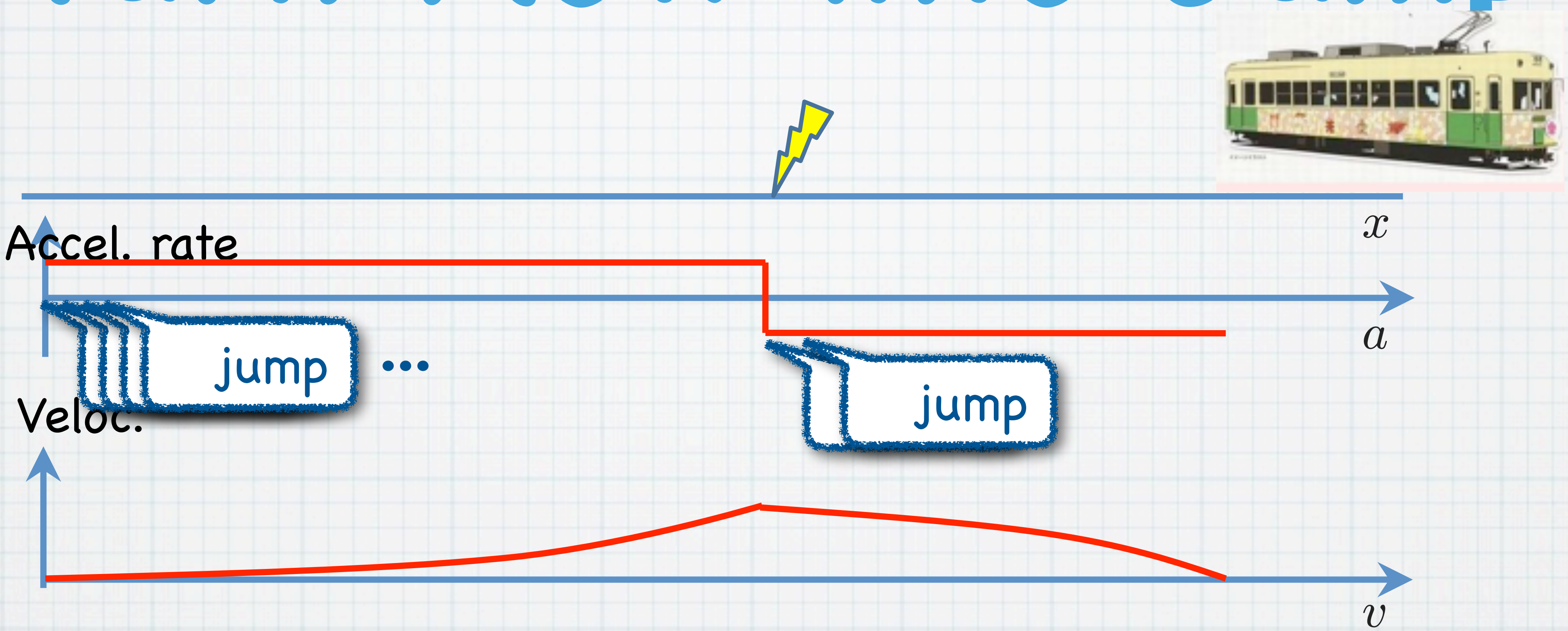

* Flow as infinitely many, infinitely small jumps

# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

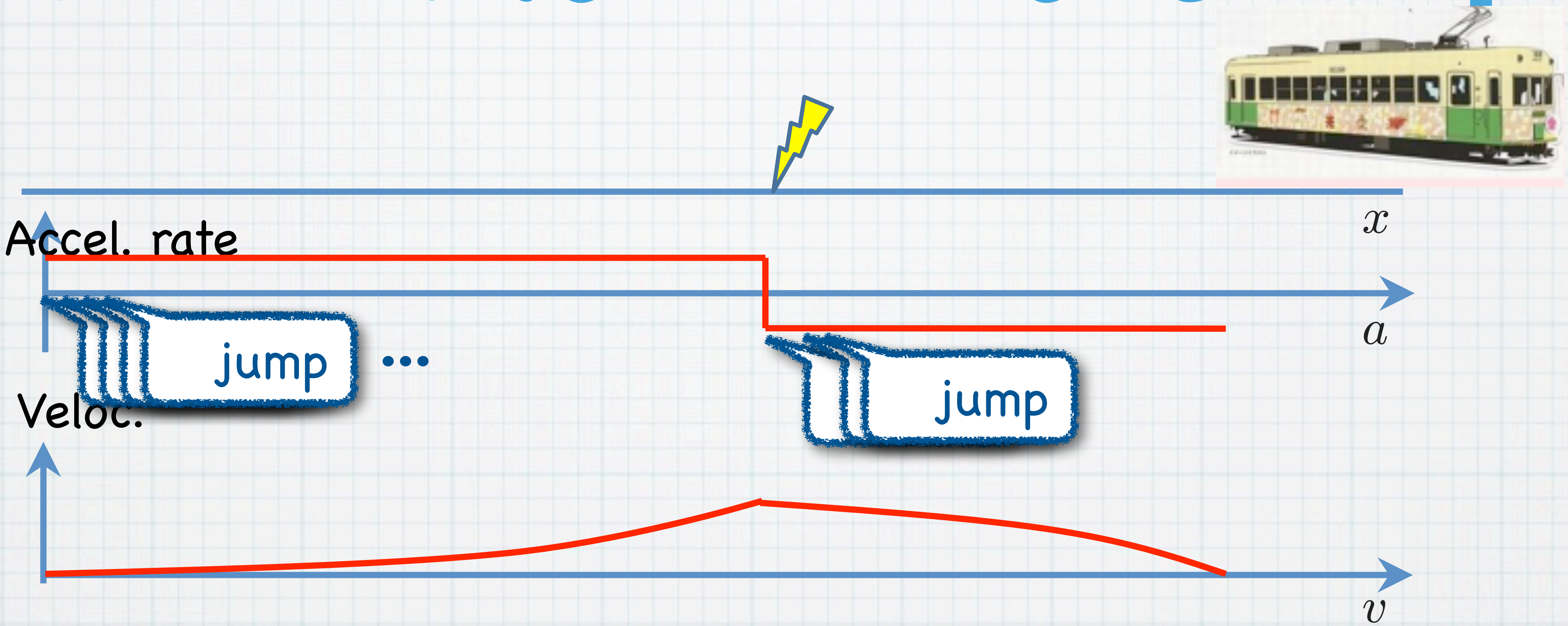
# “Turn Flow into Jump”




* Flow as infinitely many, infinitely small jumps

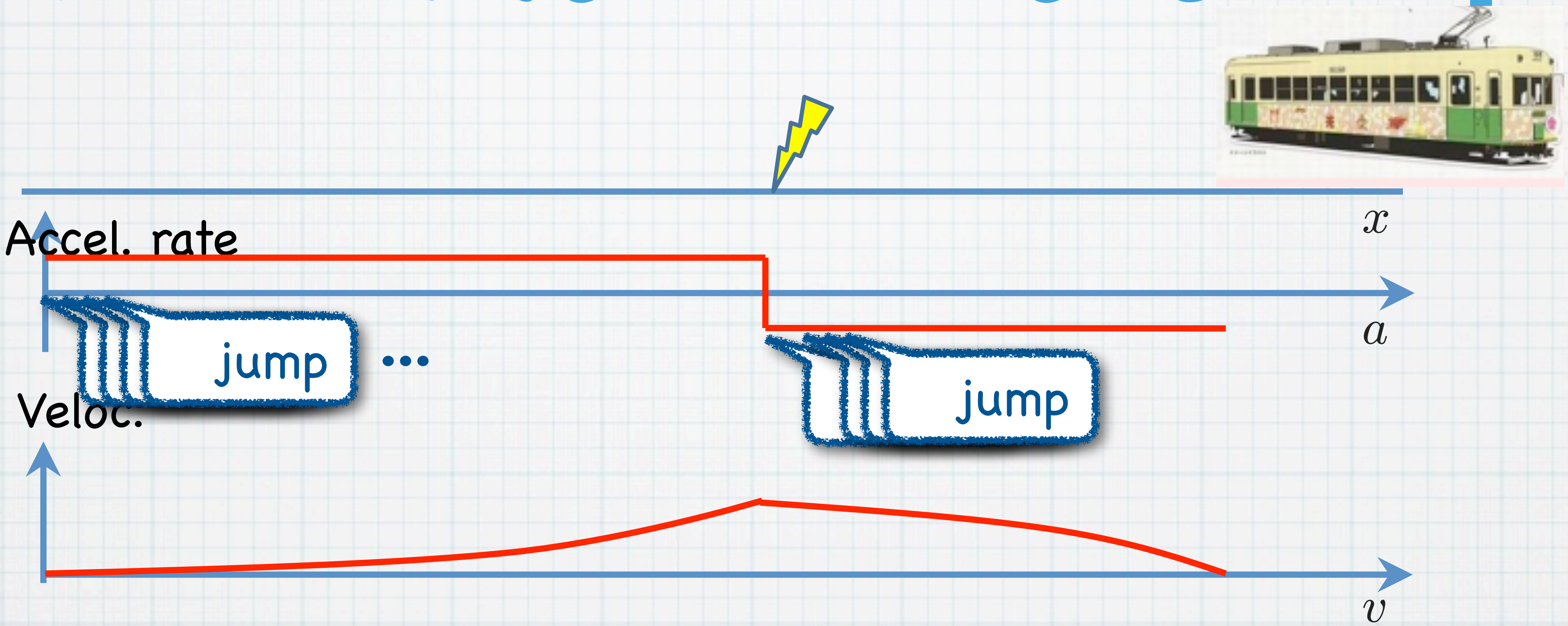
Hasuo (Tokyo)

# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

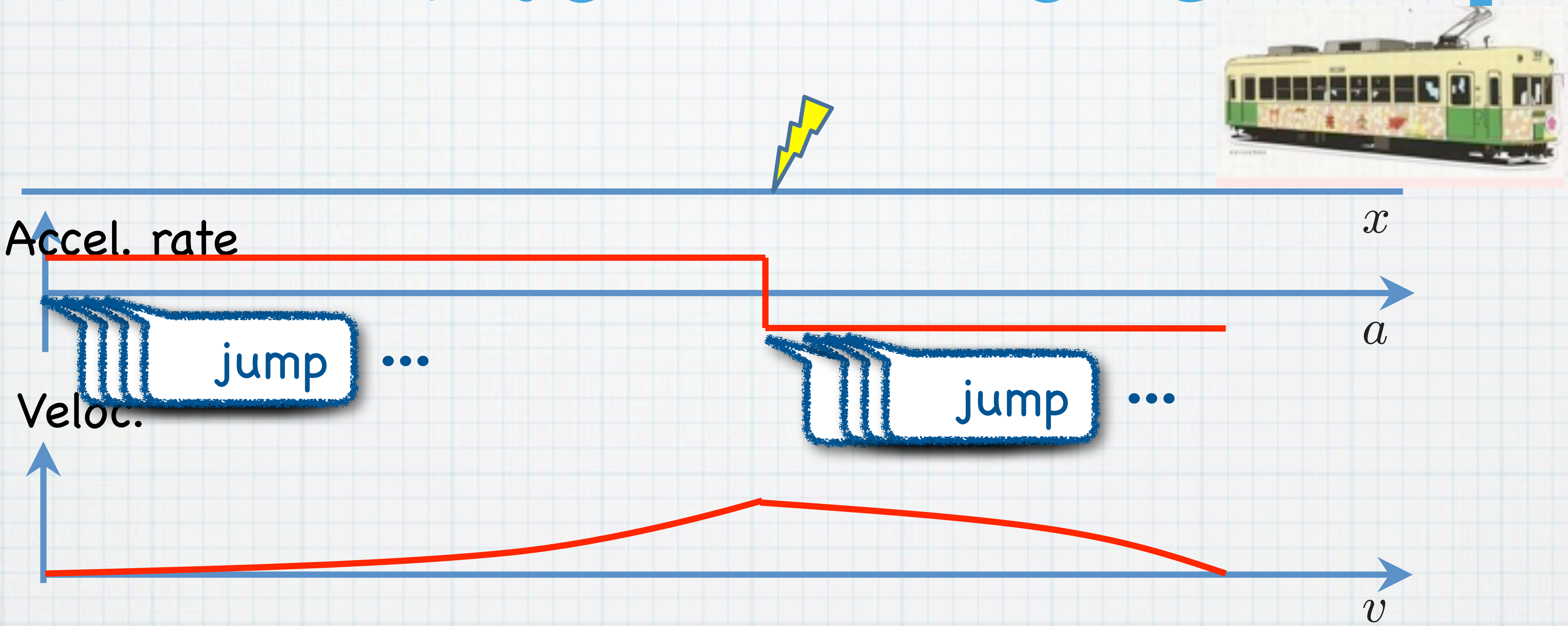
# "Turn Flow into Jump"




* Flow as infinitely many, infinitely small jumps

Hasuo (Tokyo)

# “Turn Flow into Jump”




* Flow as infinitely many, infinitely small jumps

# "Turn Flow into Jump"

$$t := 0\ ;$$
$$\texttt{while}\ (t \le 1)\ \texttt{do}\ \{$$
$$\quad t := t + \texttt{dt}$$
$$\}$$

# "Turn Flow into Jump"

```
t := 0 ;
while (t ≤ 1) do {
  t := t + dt
}
```

* Infinitesimal number dt
  * "Infinitely small" :
    $0 < dt < r$
    for **any** positive real $r$

# "Turn Flow into Jump"

$$t := 0\,;$$
$$\texttt{while}\ (t \le 1)\ \mathrm{do}\ \{$$
$$\quad t := t + \mathrm{dt}$$
$$\}$$

* Infinitesimal number dt
  * "Infinitely small" : $0 < \mathrm{dt} < r$ for **any** positive real $r$
* $t = 1$ after the execution?

# "Turn Flow into Jump"

$$
\begin{array}{l}
t := 0\,; \\
\texttt{while}\ (t \leq 1)\ \mathrm{do}\ \{ \\
\quad t := t + \mathrm{dt} \\
\}
\end{array}
$$

* Infinitesimal number dt
  * "Infinitely small" : $0 < \mathrm{dt} < r$ for **any** positive real $r$
* $t$ = 1 after the execution?
* Nonstandard analysis! [Robinson '60s]

# Previous Work

[Suenaga & H., ICALP'11]

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

**While**

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
}
```

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

**While**

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
}
```

**Assn**

First-order assertion lang.

$\exists z(x=2*z \wedge y=3*z)$

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

## While

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
}
```

## Assn

First-order assertion lang.

∃z(x=2*z ∧ y=3*z)

## Hoare

Hoare-style program logic

$$\frac{\{A \land b\}\, c\, \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\ \{A \land \neg b\}}$$

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

## $\mathbf{While}^{dt}$

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
}
```

## $\mathbf{Assn}^{dt}$

First-order assertion lang.

∃z(x=2*z ∧ y=3*z)

## $\mathbf{Hoare}^{dt}$

Hoare-style program logic

$$\frac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\, \{A \wedge \neg b\}}$$

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

## $\textbf{While}^{\mathrm{dt}}$

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
}
```

## $\textbf{Assn}^{\mathrm{dt}}$

First-order assertion lang.

```
∃z(x=2*z ∧ y=3*z)
```

## $\textbf{Hoare}^{\mathrm{dt}}$

Hoare-style program logic

$$\frac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\, \{A \wedge \neg b\}}$$

Rigorous semantics by nonstandard analysis

# Syntax [Suenaga & H., ICALP'11]

## While$^{dt}$

$$\mathbf{AExp} \ni \quad a \quad ::= \quad x \mid \mathsf{c}_r \mid a_1 \;\mathsf{aop}\; a_2 \mid \mathsf{dt}$$

where $\mathsf{c}_r$ is a const. for $r \in \mathbb{R}$, $\mathsf{aop} \in \{+, -, \cdot, \wedge, /\}$

$$\mathbf{BExp} \ni \quad b \quad ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2$$

$$\mathbf{Cmd} \ni \quad c \quad ::= \quad \mathtt{skip} \mid x := a \mid c_1; c_2 \mid \mathtt{if}\; b \;\mathtt{then}\; c_1 \;\mathtt{else}\; c_2 \mid \mathtt{while}\; b \;\mathtt{do}\; c$$

## Assn$^{dt}$

$$A \quad ::= \quad \mathtt{true} \mid \mathtt{false} \mid A_1 \wedge A_2 \mid \neg A \mid a_1 < a_2 \mid \forall x \in {}^*\mathbb{N}.\, A \mid \forall x \in {}^*\mathbb{R}.\, A$$

## Hoare$^{dt}$

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\,x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\; b\;\mathtt{then}\; c_1\;\mathtt{else}\; c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\; b\;\mathtt{do}\; c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

While$^{\mathrm{dt}}$

# Syntax [Suenaga & H., ICALP'11]

$$
\begin{array}{llll}
\mathbf{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \,\mathtt{aop}\, a_2 \mid \mathtt{dt} \\
& & & \text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}\,, /\} \\
\mathbf{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\mathbf{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & & \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}
$$

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\,x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1 ; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\ b\ \mathtt{do}\ c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

**While$^{\mathrm{dt}}$**

# Syntax [Suenaga & H., ICALP'11]

$$\mathbf{AExp} \ni \quad a \quad ::= \quad x \mid \mathtt{c}_r \mid a_1 \;\mathtt{aop}\; a_2 \mid \mathtt{dt}$$

where $\mathtt{c}_r$ is a const. for $r \in \mathbb{R}$, $\mathtt{aop} \in \{+, -, \cdot, \hat{}\,, /\}$

$$\mathbf{BExp} \ni \quad b \quad ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2$$

$$\mathbf{Cmd} \ni \quad c \quad ::= \quad \mathtt{skip} \mid x := a \mid c_1; c_2 \mid \mathtt{if}\; b \;\mathtt{then}\; c_1 \;\mathtt{else}\; c_2 \mid \mathtt{while}\; b \;\mathtt{do}\; c$$

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\; x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\; b \;\mathtt{then}\; c_1 \;\mathtt{else}\; c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\; b \;\mathtt{do}\; c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

**While$^{\mathtt{dt}}$**

While + dt

# Syntax [Suenaga & H., ICALP'11]

$$\begin{array}{llll}
\mathbf{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \;\mathtt{aop}\; a_2 \mid \mathtt{dt} \\
& & & \text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+,-,\cdot,\hat{},/\} \\
\mathbf{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\mathbf{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1; c_2 \\
& & & \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}$$

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\,x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\ b\ \mathtt{do}\ c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

# Syntax [Suenaga & H., ICALP'11]

While + dt

**While$^{\mathrm{dt}}$**

$$
\begin{array}{llll}
\mathrm{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \;\mathtt{aop}\; a_2 \mid \mathtt{dt} \\
& & & \text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\} \\
\mathrm{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\mathrm{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1; c_2 \\
& & & \mid \mathtt{if}\; b \;\mathtt{then}\; c_1 \;\mathtt{else}\; c_2 \mid \mathtt{while}\; b \;\mathtt{do}\; c
\end{array}
$$

**Assn$^{\mathrm{dt}}$**

$$
\begin{array}{lll}
A & ::= & \mathtt{true} \mid \mathtt{false} \mid A_1 \wedge A_2 \mid \neg A \mid a_1 < a_2 \mid \\
& & \forall x \in {}^*\mathbb{N}.\, A \mid \forall x \in {}^*\mathbb{R}.\, A
\end{array}
$$

**Hoare$^{\mathrm{dt}}$**

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\; x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\; b\;\mathtt{then}\; c_1\;\mathtt{else}\; c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\; b\;\mathtt{do}\; c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

# Syntax [Suenaga & H., ICALP'11]

## While$^{\mathrm{dt}}$

While + dt

$$\begin{array}{llll}
\mathrm{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \;\mathtt{aop}\; a_2 \mid \mathtt{dt} \\
& & & \text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\} \\
\mathrm{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\mathrm{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & & \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}$$

## Assn$^{\mathrm{dt}}$

Assn, *-transformed

$$\begin{array}{lll}
A & ::= & \mathtt{true} \mid \mathtt{false} \mid A_1 \wedge A_2 \mid \neg A \mid a_1 < a_2 \mid \\
& & \forall x \in {}^*\mathbb{N}.\, A \mid \forall x \in {}^*\mathbb{R}.\, A
\end{array}$$

## Hoare$^{\mathrm{dt}}$

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\,x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1 ; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\ b\ \mathtt{do}\ c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

While$^{\mathrm{dt}}$

While + dt

# Syntax [Suenaga & H., ICALP'11]

AExp $\ni$ $a$ ::= $x \mid \mathtt{c}_r \mid a_1\ \mathtt{aop}\ a_2 \mid \mathtt{dt}$

where $\mathtt{c}_r$ is a const. for $r \in \mathbb{R}$, $\mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}$

BExp $\ni$ $b$ ::= $\mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2$

$$\frac{}{\{A\}\ \mathtt{skip}\ \{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\ A[a/x]\ \}\ x := a\ \{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\ c_1\ \{C\} \quad \{C\}\ c_2\ \{B\}}{\{A\}\ c_1; c_2\ \{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\ c_1\ \{B\} \quad \{A \wedge \neg b\}\ c_2\ \{B\}}{\{A\}\ \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2\ \{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\ c\ \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\ \{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\ c\ \{B'\} \quad \models B' \Rightarrow B}{\{A\}\ c\ \{B\}}\ (\textsc{Conseq})$$

Hoare$^{\mathrm{dt}}$

**While**$^{dt}$

While + dt

# Syntax

[Suenaga & H., ICALP'11]

$$\mathbf{AExp} \ni \quad a \quad ::= \quad x \mid \mathtt{c}_r \mid a_1 \;\mathtt{aop}\; a_2 \mid \mathtt{dt}$$

where $\mathtt{c}_r$ is a const. for $r \in \mathbb{R}$, $\mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}$

$$\mathbf{BExp} \ni \quad b \quad ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2$$

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{\,A[a/x]\,\}\;x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\mathtt{if}\;b\;\mathtt{then}\;c_1\;\mathtt{else}\;c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\;b\;\mathtt{do}\;c\,\{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

**Hoare**$^{dt}$

**Precisely** the same rules

# Syntax [Suenaga & H., ICALP'11]

**While$^{dt}$** — While + dt

$$\begin{array}{llll}
\text{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop}\ a_2 \mid \mathtt{dt} \\
& \multicolumn{3}{l}{\text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}} \\
\text{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\text{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & & \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}$$

**Assn$^{dt}$** — Assn, *-transformed

$$A ::= \mathtt{true} \mid \mathtt{false} \mid A_1 \wedge A_2 \mid \neg A \mid a_1 < a_2 \mid \forall x \in {}^*\mathbb{N}.\, A \mid \forall x \in {}^*\mathbb{R}.\, A$$

**Hoare$^{dt}$** — Precisely the same rules

$$\frac{}{\{A\}\ \mathtt{skip}\ \{A\}}\ (\textsc{Skip}) \qquad \frac{}{\{A[a/x]\}\ x := a\ \{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\ c_1\ \{C\} \quad \{C\}\ c_2\ \{B\}}{\{A\}\ c_1; c_2\ \{B\}}\ (\textsc{Seq}) \qquad \frac{\{A \wedge b\}\ c_1\ \{B\} \quad \{A \wedge \neg b\}\ c_2\ \{B\}}{\{A\}\ \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2\ \{B\}}\ (\textsc{If})$$

$$\frac{\{A \wedge b\}\ c\ \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\ \{A \wedge \neg b\}}\ (\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\ c\ \{B'\} \quad \models B' \Rightarrow B}{\{A\}\ c\ \{B\}}\ (\textsc{Conseq})$$

Tokyo)

# Syntax [Suenaga & H., ICALP'11]

**While$^{dt}$**

**While + dt**

$$\mathrm{AExp} \ni \quad a \quad ::= \quad x \mid \mathtt{c}_r \mid a_1 \;\mathtt{aop}\; a_2 \mid \mathtt{dt}$$

where $\mathtt{c}_r$ is a const. for $r \in \mathbb{R}$, $\mathtt{aop} \in \{+, -, \cdot, \wedge, /\}$

$$\mathrm{BExp} \ni \quad b \quad ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2$$

$$\mathrm{Cmd} \ni \quad c \quad ::= \quad \mathtt{skip} \mid x := a \mid c_1; c_2 \mid \mathtt{if}\; b \;\mathtt{then}\; c_1 \;\mathtt{else}\; c_2 \mid \mathtt{while}\; b \;\mathtt{do}\; c$$

**Thm.**
HOARE$^{dt}$ rules are sound and relatively complete.

**Hoare$^{dt}$**

Precise...

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\,(\textsc{Skip}) \qquad \frac{}{\{A[a/x]\}\, x := a\, \{A\}}$$

$$\frac{\{A\}\, c_1\, \{C\} \quad \{C\}\, c_2\, \{B\}}{\{A\}\, c_1; c_2\, \{B\}}\,(\textsc{Seq}) \qquad \frac{\{A \wedge b\}\, c_1\, \{B\} \quad \{A \wedge \neg b\}\, c_2\, \{B\}}{\{A\}\,\mathtt{if}\; b\;\mathtt{then}\; c_1\;\mathtt{else}\; c_2\,\{B\}}\,(\textsc{If})$$

$$\frac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\,\mathtt{while}\; b\;\mathtt{do}\; c\,\{A \wedge \neg b\}}\,(\textsc{While}) \qquad \frac{\models A \Rightarrow A' \quad \{A'\}\, c\, \{B'\} \quad \models B' \Rightarrow B}{\{A\}\, c\, \{B\}}\,(\textsc{Conseq})$$

Tokyo)

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

**While**$^{\mathrm{dt}}$

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
```

**Assn**$^{\mathrm{dt}}$

First-order assertion lang.

$\exists z(x=2*z \wedge y=3*z)$

**Hoare**$^{\mathrm{dt}}$

Hoare-style program logic

$$\frac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\, \{A \wedge \neg b\}}$$

Rigorous semantics by nonstandard analysis

- **Hoare**$^{\mathrm{dt}}$ : sound and relatively complete

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

**While**$^{\mathrm{dt}}$

Programming lang.

```
while (t<a) do {
  t:=t+1;
  if ...
```

**Assn**$^{\mathrm{dt}}$

First-order assertion lang.

∃z(x=2*z ∧ y=3*z)

**Hoare**$^{\mathrm{dt}}$

Hoare-style program logic

$$\frac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\, \{A \wedge \neg b\}}$$

Rigorous semantics by nonstandard analysis

- **Hoare**$^{\mathrm{dt}}$ : sound and relatively complete
- Modeling & verification of hybrid systems

# Previous Work

[Suenaga & H., ICALP'11]


The standard textbook [Winskel]

| While$^{dt}$ | Assn$^{dt}$ | Hoare$^{dt}$ |
|---|---|---|
| Programming lang. | First-order assertion lang. | Hoare-style program logic |
| `while (t<a) do {`<br>`  t:=t+1;`<br>`  if ...` | `∃z(x=2*z ∧ y=3*z)` | $\dfrac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\, \{A \wedge \neg b\}}$ |

Rigorous semantics by nonstandard analysis

* **Hoare**$^{dt}$ : sound and relatively complete
* Modeling & verification of hybrid systems
* Program analysis techniques transferred (inv. gen., ...) automatic prover [H. & Suenaga, CAV'12]

Static Analysis

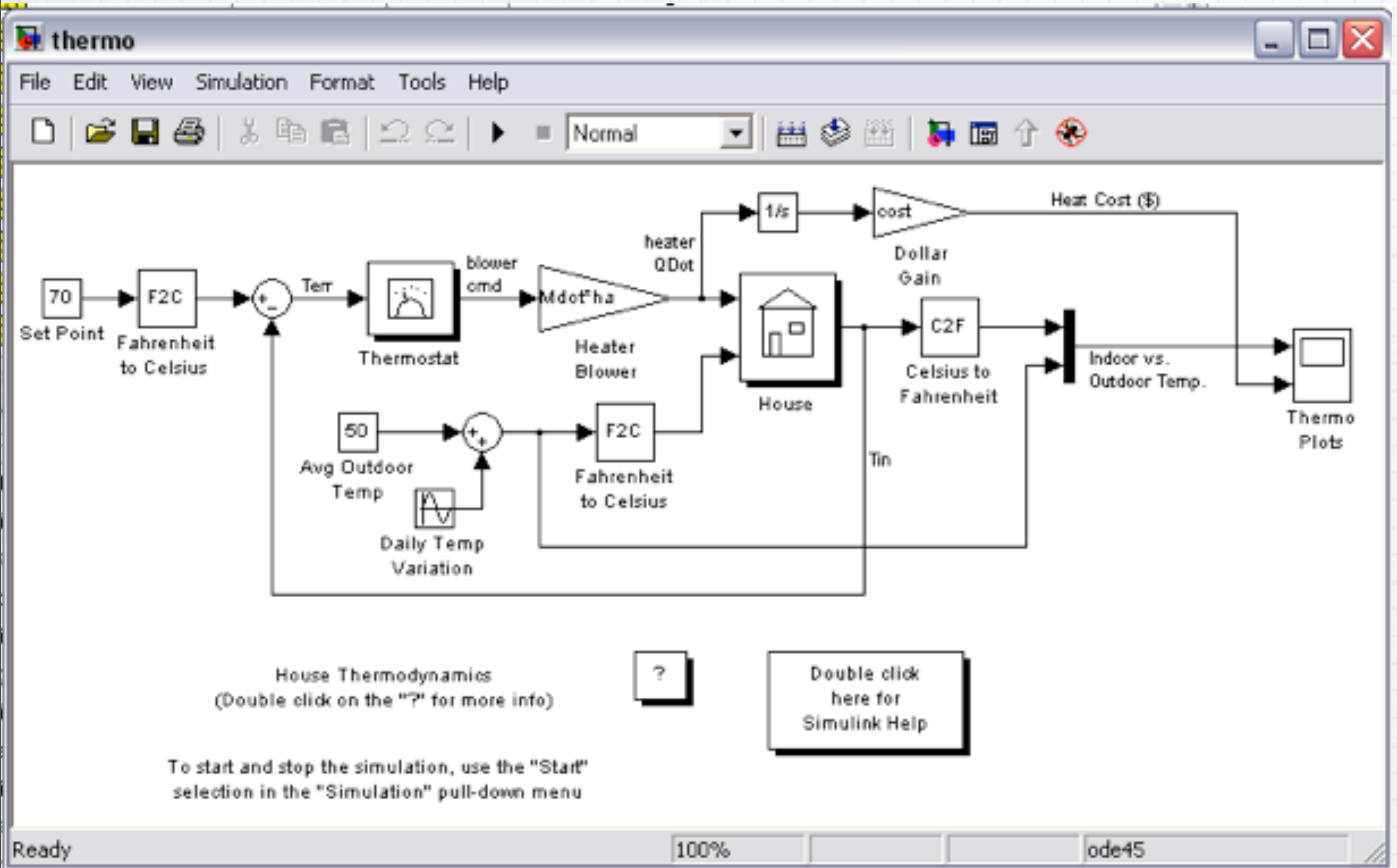
Nonstandard Analysis

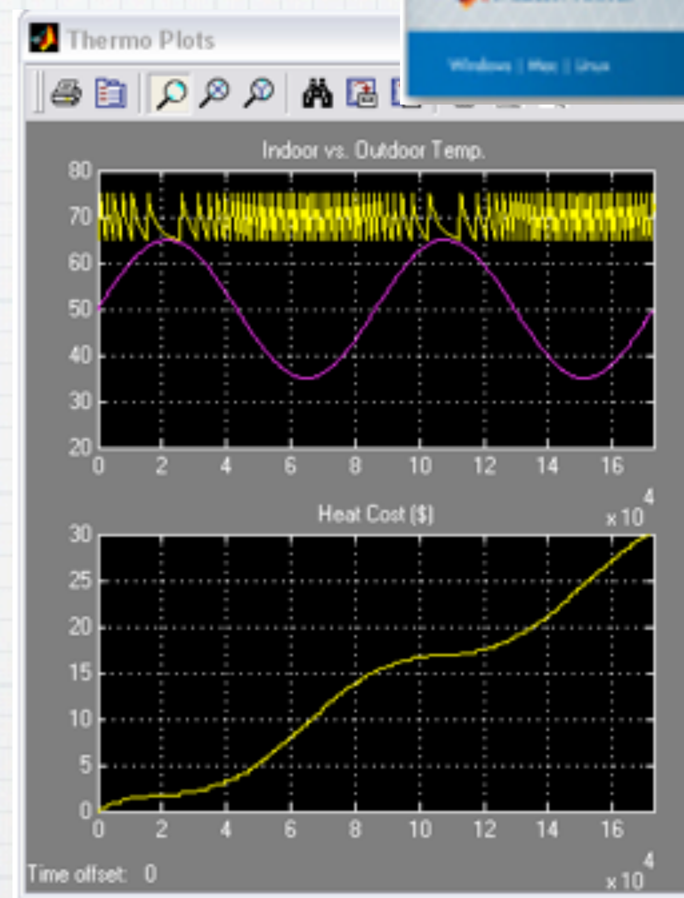
# Nonstandard Static Analysis

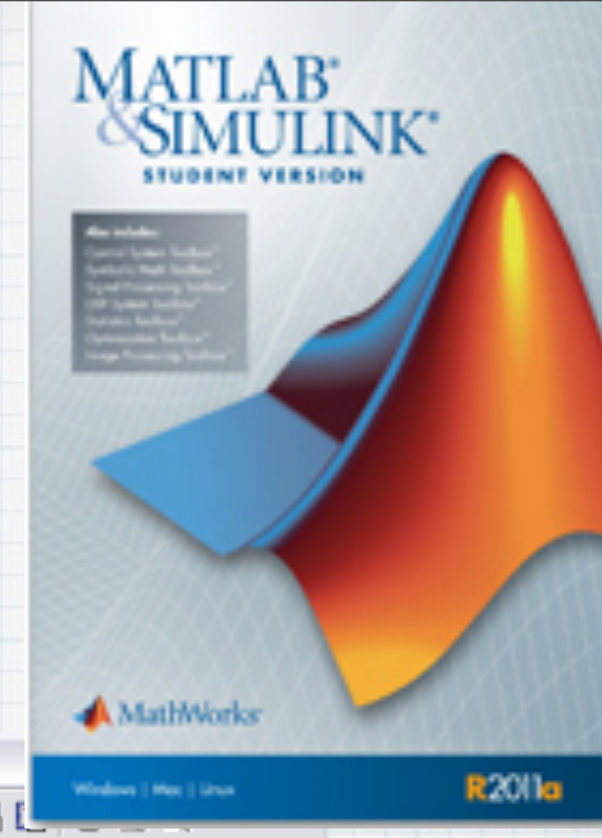
# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language |
| Program logic | **Hoare**$^{dt}$ | **Type system** |

# Simulink



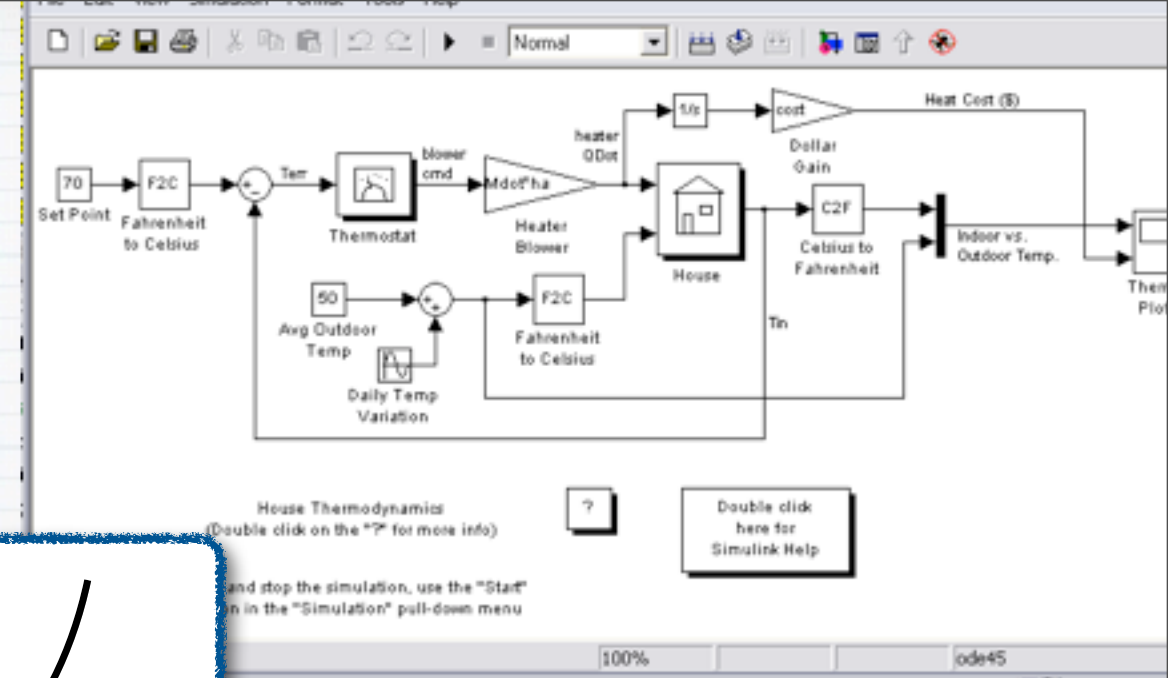



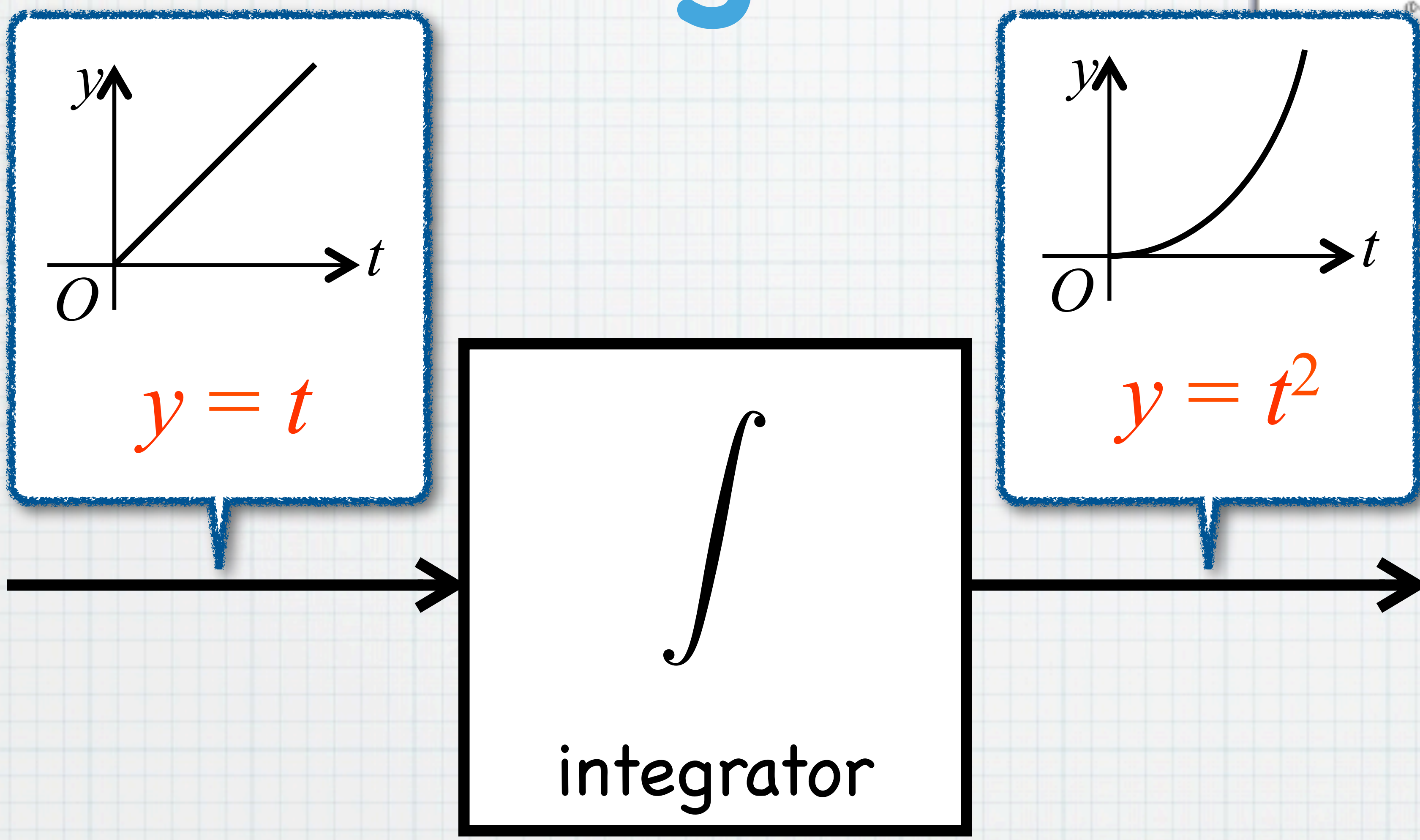




* Industry-standard tool
* Modeling + simulation (& code gen., ...)

Hasuo (Tokyo)

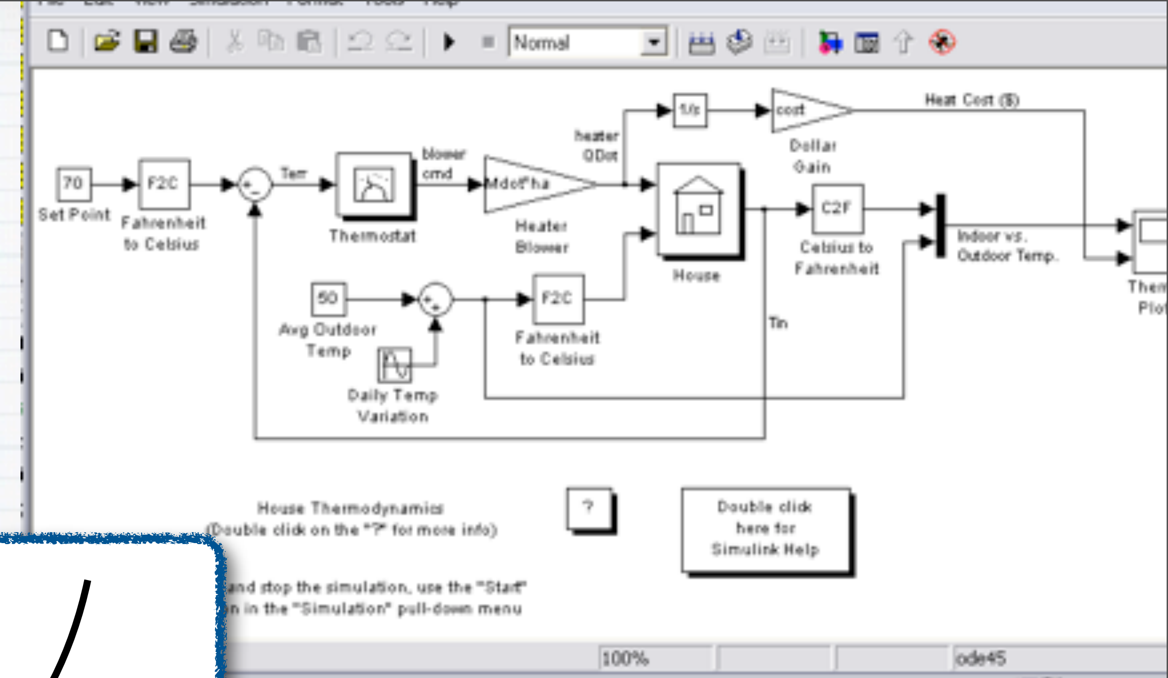
# Signal Processing

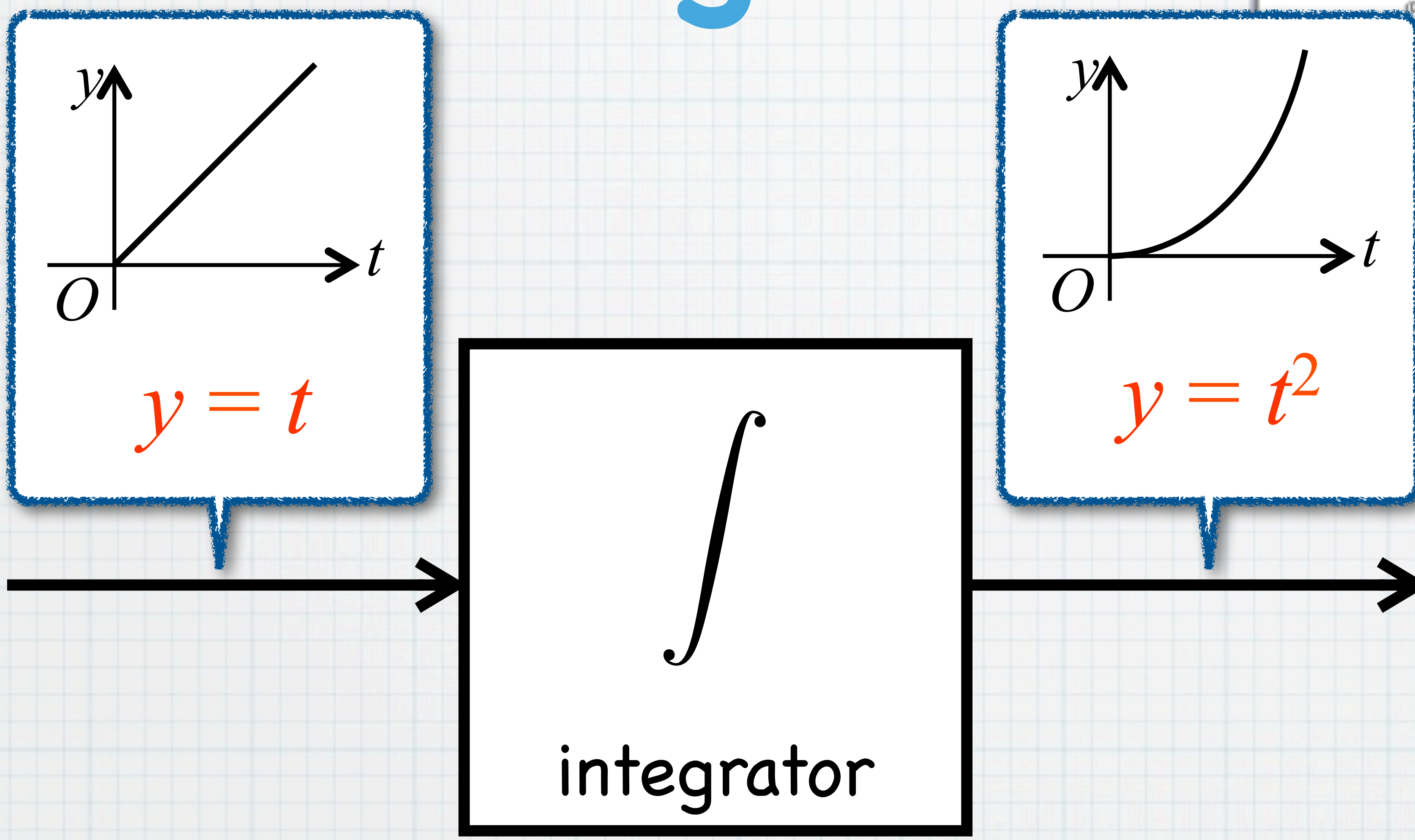



Hasuo (Tokyo)

# Signal Processing

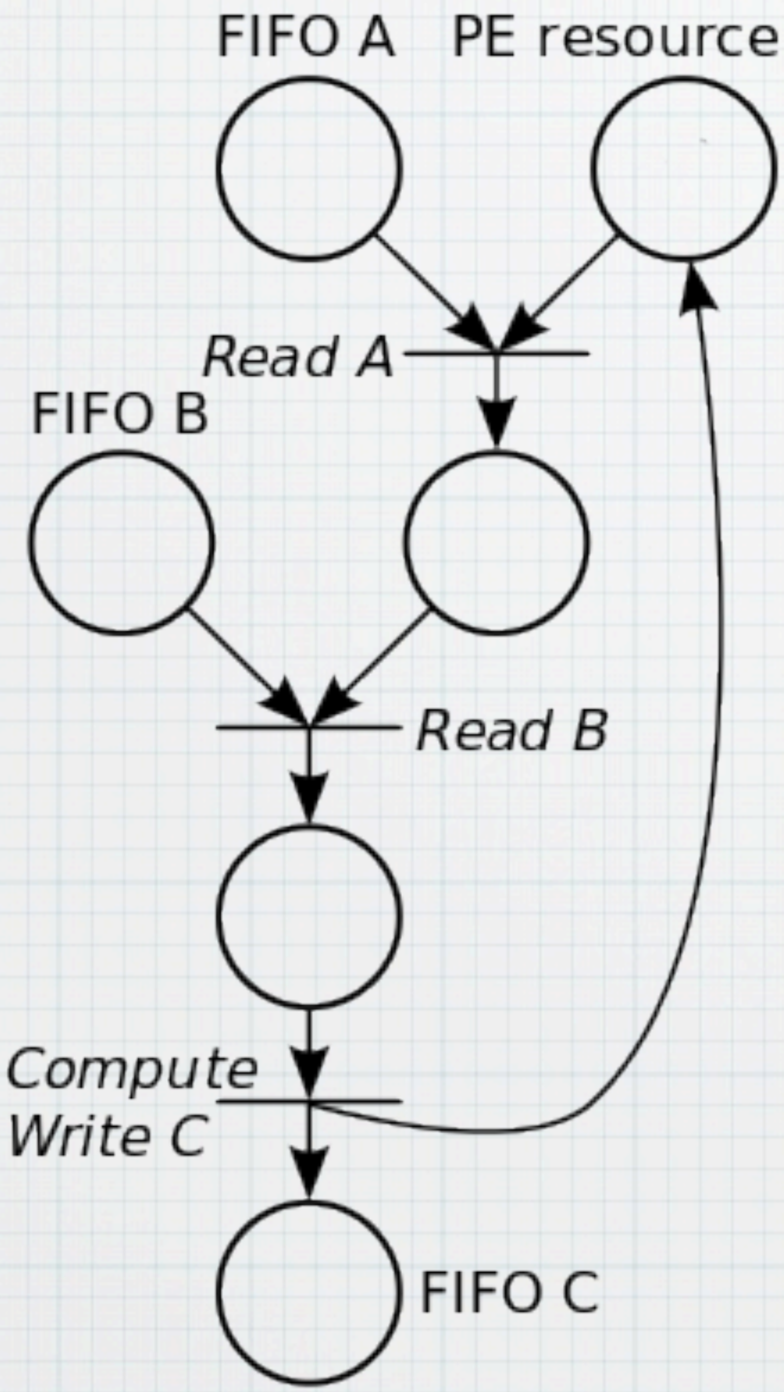


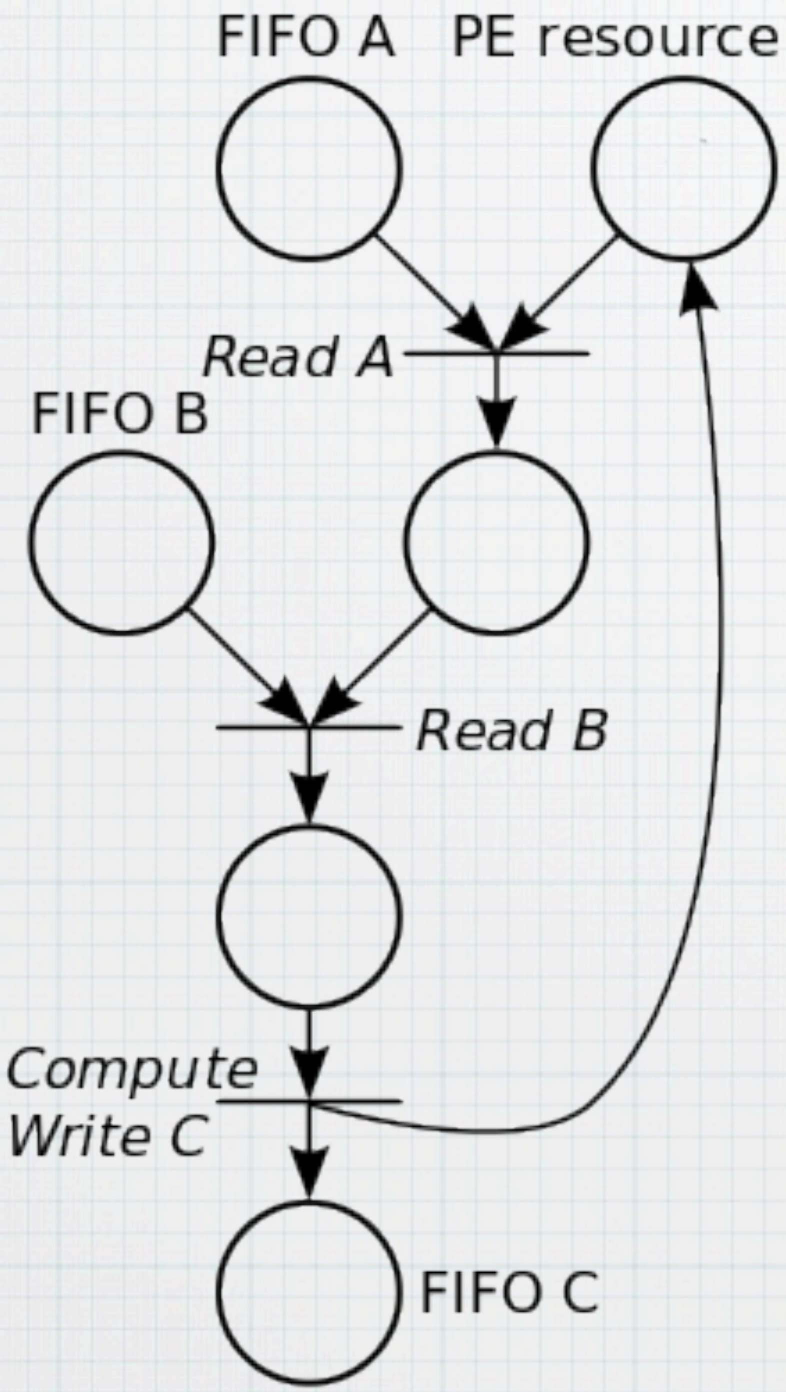

- **Signal**: value dependent on **continuous-time**
- This looks somehow familiar...

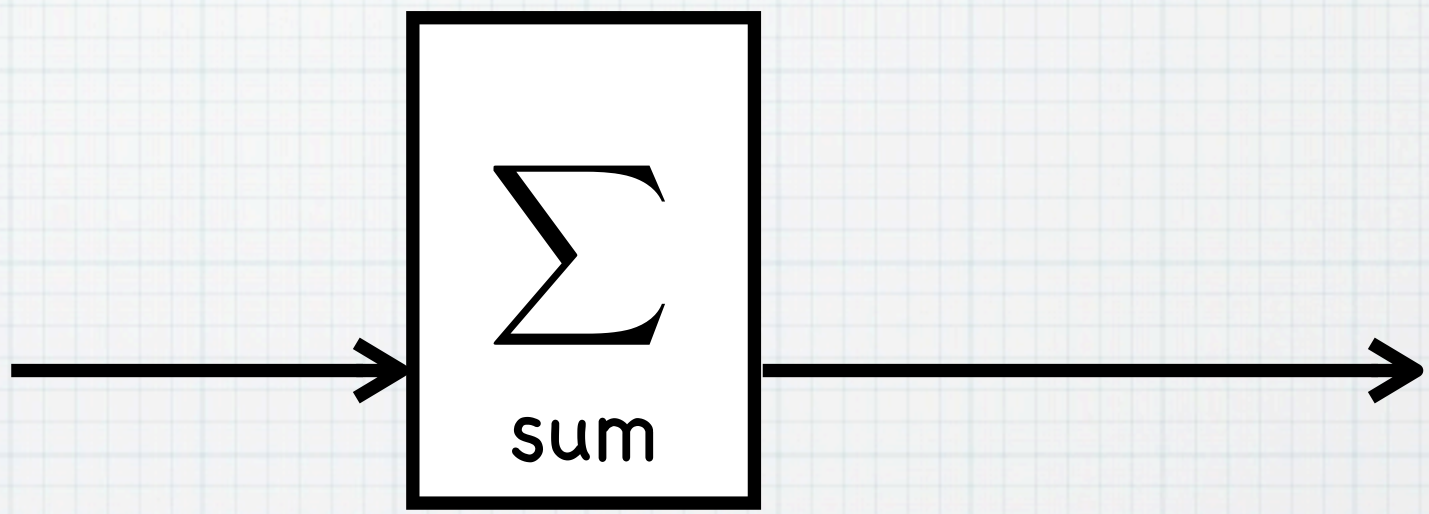
$$f : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{C}$$

# Kahn Process Diagrams



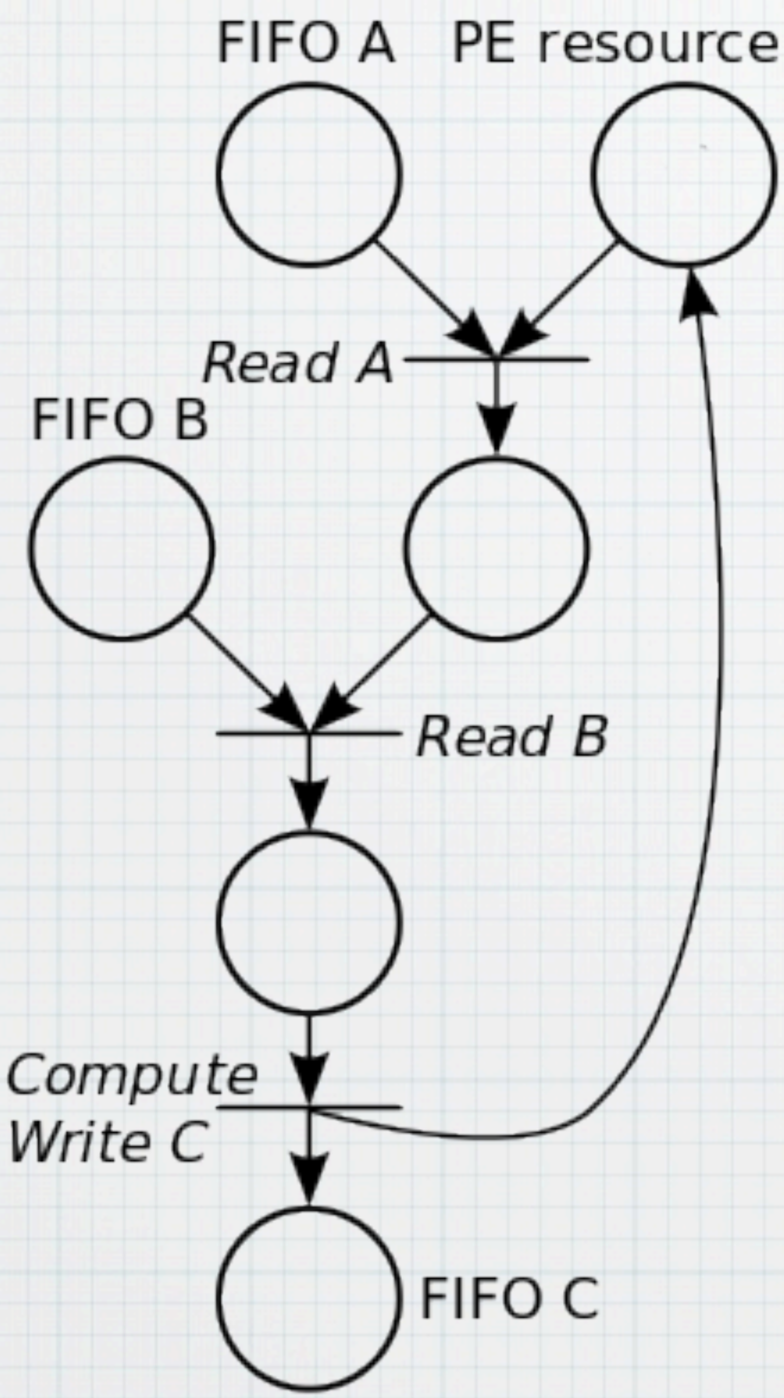
# Kahn Process Diagrams

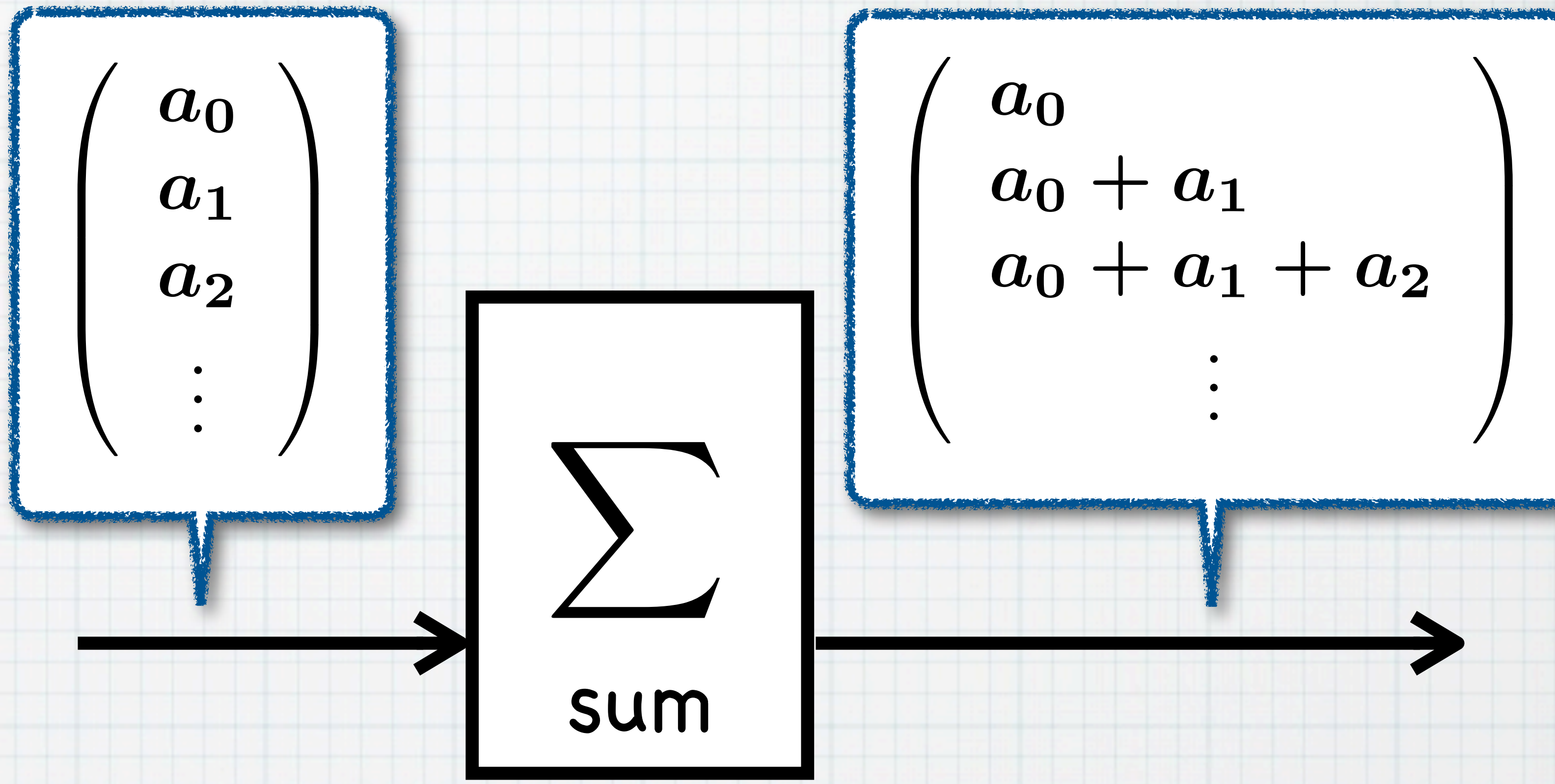


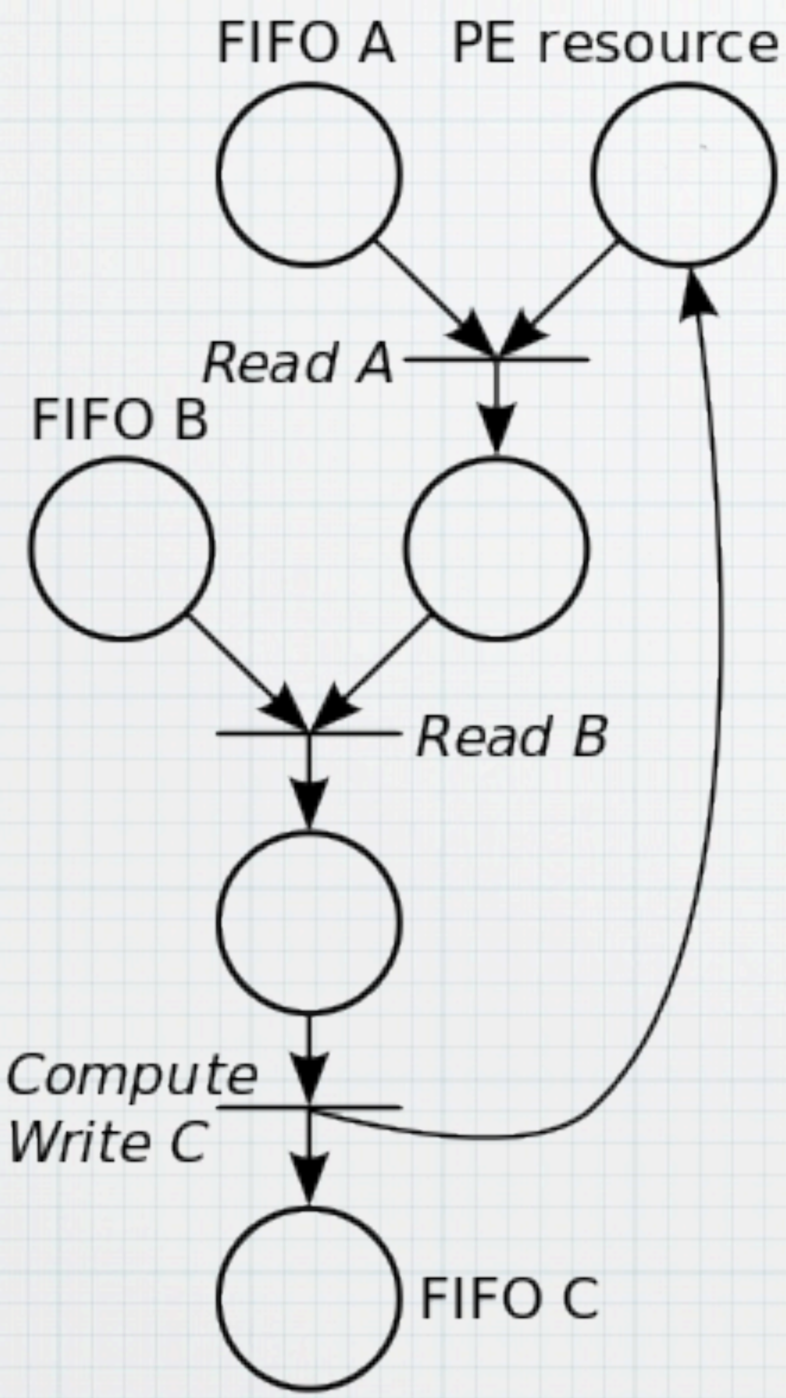

Hasuo (Tokyo)

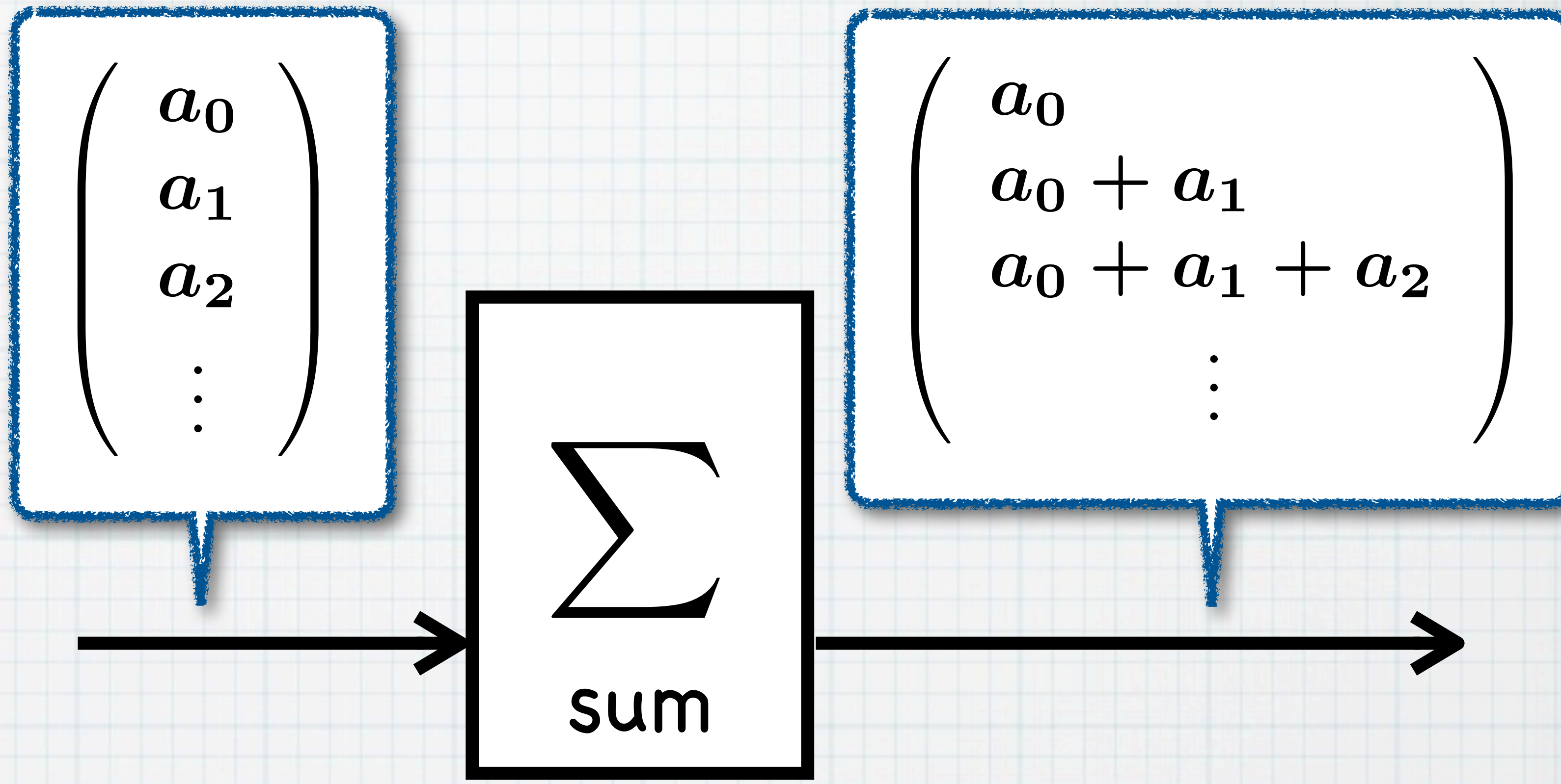
# Kahn Process Diagrams



# Kahn Process Diagrams

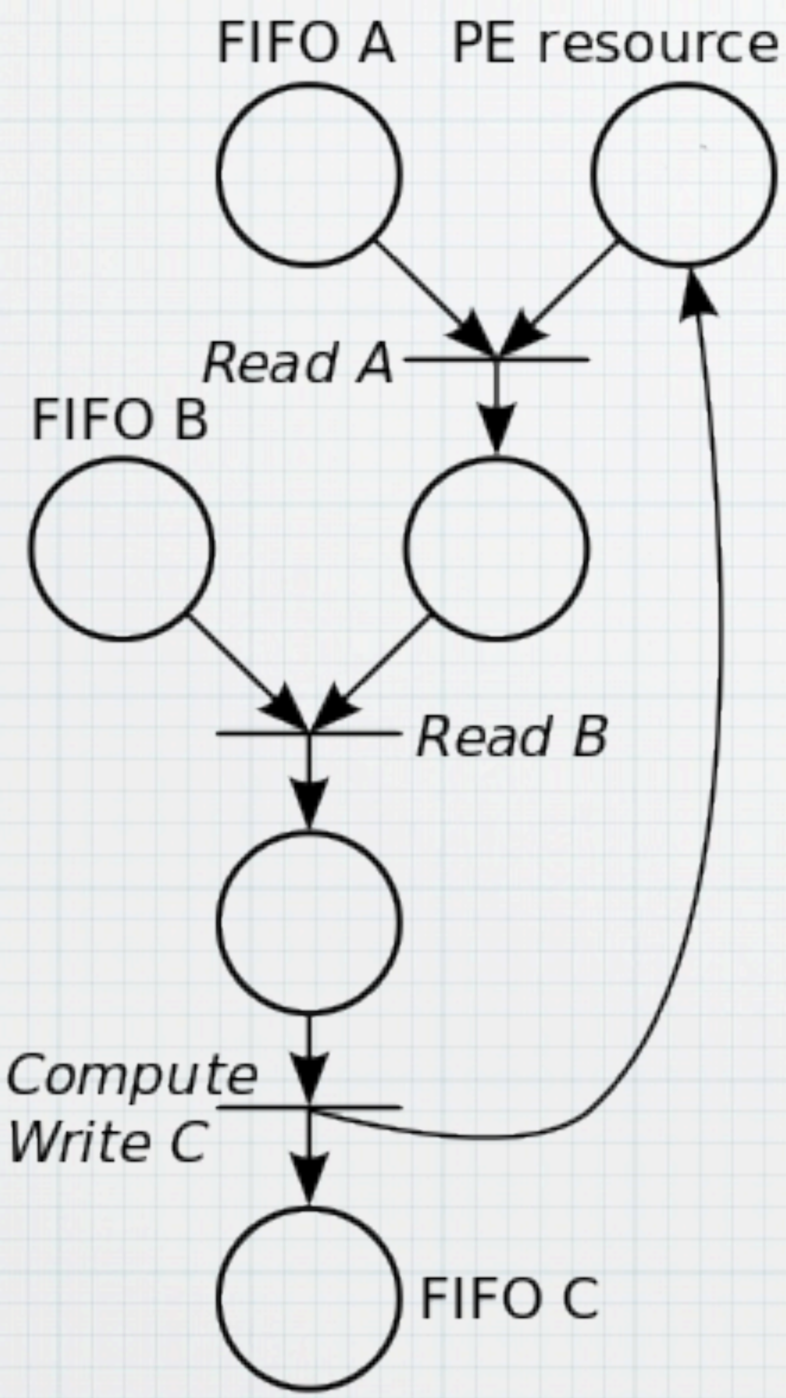


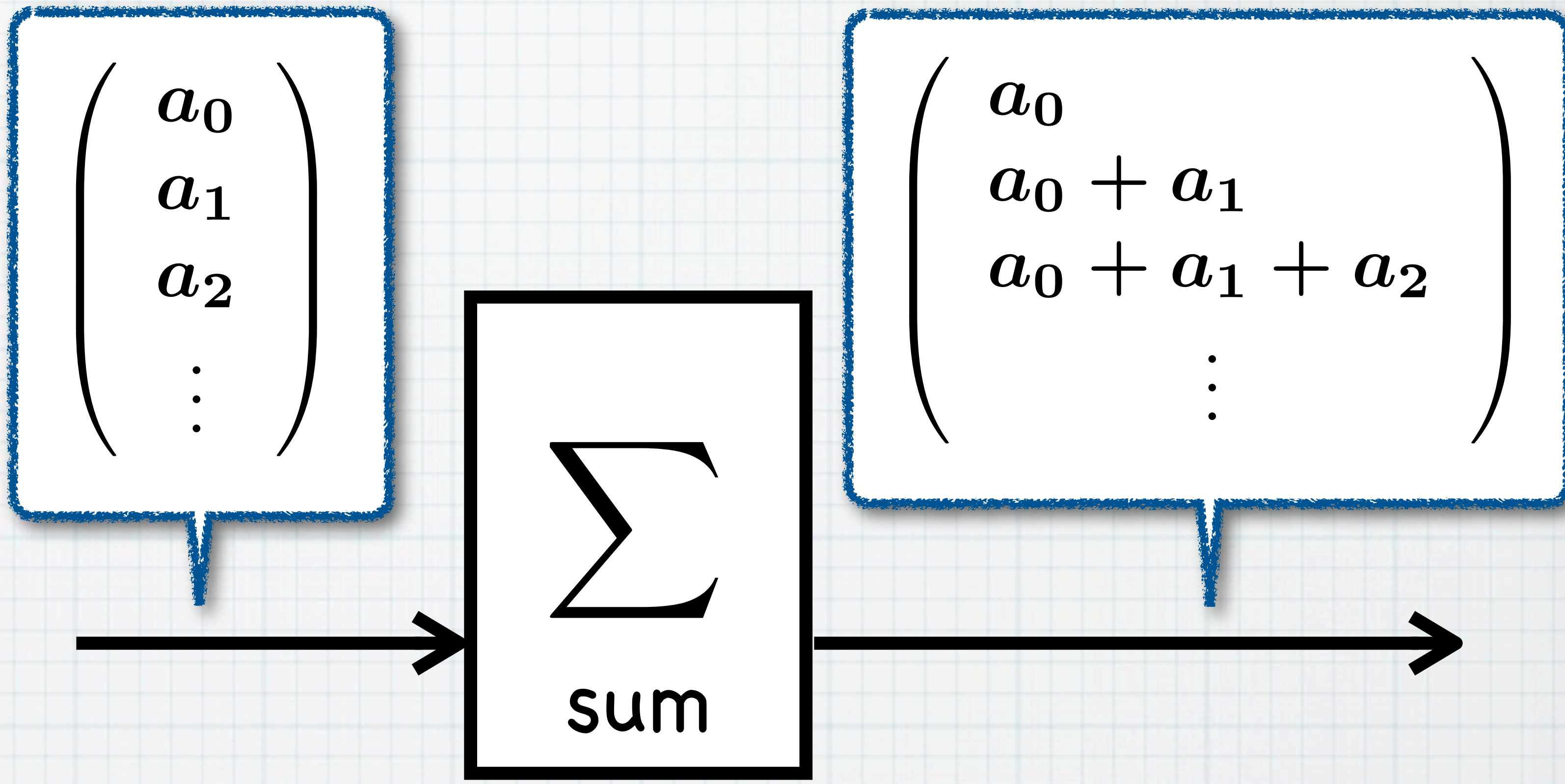




- **Stream processing system**
  - Discrete-time $s : \mathbb{N} \longrightarrow \mathbb{C}$

Hasuo (Tokyo)

# Kahn Process Diagrams




- **Stream processing system**
  - Discrete-time $s : \mathbb{N} \longrightarrow \mathbb{C}$
- Well in the realm of PL study!

Hasuo (Tokyo)

# Contribution

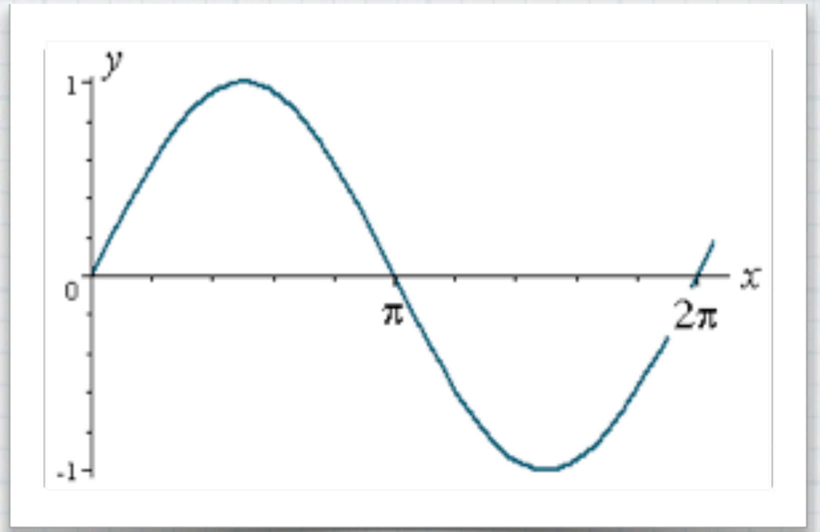
| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{\mathrm{dt}}$ Imperative | **SProc**$^{\mathrm{dt}}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{\mathrm{dt}}$ | **Type system** (partial correctness) |

* **Example** Deductive verif. of "the sine curve never exceeds 1"

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \ \texttt{fby}^1 \ (s + c \times \mathrm{dt}); \\ & c = 1 \ \texttt{fby}^1 \ (c - s \times \mathrm{dt}) \end{array}\right] : \prod_{w \in \mathbb{R}_{\geq 0}} \{u \in \mathbb{C} \mid t_0 \leq w \vee u \leq 1 + \varepsilon\}$$

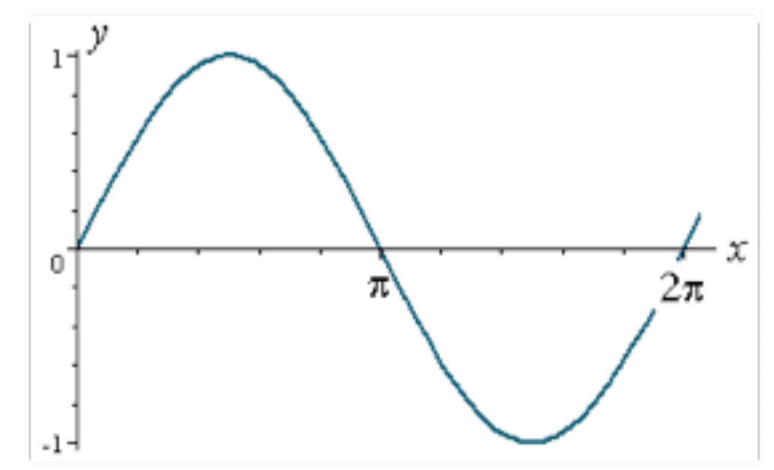



Hasuo (Tokyo)

# Nonstandard Static Analysis

* **Example** Deductive verif. of "The sine curve never exceeds 1"

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = \mathbf{0}\ \texttt{fby}^{1}\ (s + c \times \texttt{dt}); \\ & c = \mathbf{1}\ \texttt{fby}^{1}\ (c - s \times \texttt{dt}) \end{array}\right] : \prod_{w \in \mathbb{R}_{\geq 0}} \{u \in \mathbb{C} \mid t_0 \leq w \vee u \leq 1 + \varepsilon\}$$



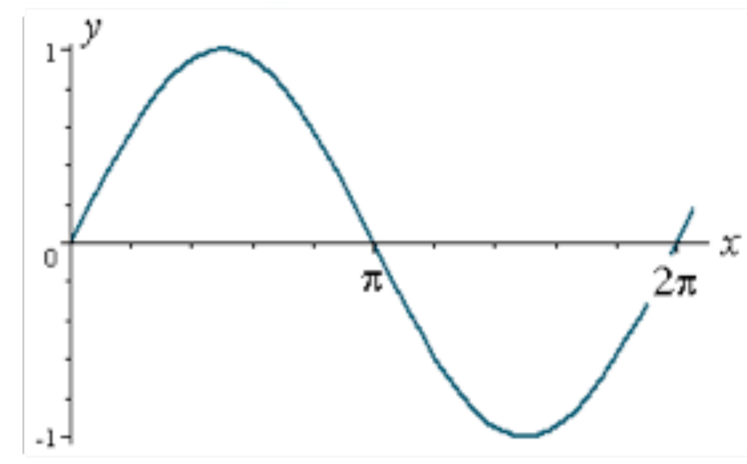

* Via the rules like

$$\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1,m] \quad \Delta;\Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1,l] \quad \Delta;\Gamma \vdash e_k : \tau''_k \text{ for } k \in [1,n]}{\Delta \vdash \left[\begin{array}{l} \texttt{node } f(x_1,\ldots,x_m) \texttt{ returns } (e_1,\ldots,e_n) \\ \texttt{where } y_1 = e'_1; \ldots ; y_l = e'_l \end{array}\right] : (\tau_1,\ldots,\tau_m) \to (\tau''_1,\ldots,\tau''_n)} \ (\textsc{Node})$$

# Nonstandard Static Analysis

- **Example** Deductive verif. of "The sine curve never exceeds 1"

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \; \texttt{fby}^1 \; (s + c \times \texttt{dt}); \\ & c = 1 \; \texttt{fby}^1 \; (c - s \times \texttt{dt}) \end{array}\right] \quad : \quad \prod_{w \in \mathbb{R}_{\geq 0}} \{ u \in \mathbb{C} \mid t_0 \leq w \vee u \leq 1 + \varepsilon \}$$




- Via the rules like

$$\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1,m] \quad \Delta;\Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1,l] \quad \Delta;\Gamma \vdash e_k : \tau''_k \text{ for } k \in [1,n]}{\Delta \vdash \left[\begin{array}{l} \texttt{node } f(x_1,\dots,x_m) \texttt{ returns } (e_1,\dots,e_n) \\ \texttt{where } y_1 = e'_1; \dots ; y_l = e'_l \end{array}\right] : (\tau_1,\dots,\tau_m) \to (\tau''_1,\dots,\tau''_n)} \ (\textsc{Node})$$

**(fixed pt. induction)**

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

- I. Stream Processing Language **SProc**
- II. Nonstandard Analysis
- III. **SProc**$^{dt}$ and Type System
- IV. Hyperstream Sampling

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

* I. Stream Processing Language **SProc**
* II. Nonstandard Analysis
* III. **SProc**$^{dt}$ and Type System
* IV. Hyperstream Sampling

# Part I:

## Stream Processing Language

## SProc

# The Language SProc

$$
\begin{array}{rcl}
\mathbf{SExp}_{\mathbb{C}} \ni e & ::= & x \mid c \mid e_1 + e_2 \mid e_1 \times e_2 \mid e_1 \,\hat{}\, e_2 \mid e_1 \;\texttt{fby}^j\; e_2 \\
& & \mid \texttt{if}\; b \;\texttt{then}\; e_1 \;\texttt{else}\; e_2 \mid \texttt{proj}_k\; f(e_1, \dots, e_m) \\
& & \quad \text{where } x \in \mathbf{SVar};\ c \in \mathbb{C};\ j \in \mathbb{N};\ b \in \mathbf{SExp}_{\mathbb{B}}; \\
& & \qquad f \in \mathbf{NdName}_{m,n};\ \text{and } k \in [1, n] \\
\mathbf{SExp}_{\mathbb{B}} \ni b & ::= & \texttt{true} \mid \texttt{false} \mid b_1 \wedge b_2 \mid \neg b \mid e_1 = e_2 \mid \texttt{isReal}(e) \mid e_1 < e_2 \\
& & \quad \text{where } e, e_i \in \mathbf{SExp}_{\mathbb{C}} \\
\mathbf{Nodes} \ni \mathsf{nd} & ::= & \left[\begin{array}{l} \texttt{node}\; f(x_1, \dots, x_m) \;\texttt{returns}\; (e_1, \dots, e_n) \\ \texttt{where}\; y_1 = e'_1; y_2 = e'_2; \dots; y_l = e'_l \end{array}\right] \\
& & \quad \text{where } f \in \mathbf{NdName}_{m,n};\ x_i, y_i \in \mathbf{SVar};\ e_i, e'_i \in \mathbf{SExp}_{\mathbb{C}}; \\
& & \quad x_1, \dots, x_m, y_1, \dots, y_n \text{ are all distinct; and the variables} \\
& & \quad \text{occurring in } e_i, e'_i \text{ are restricted to } x_i \text{ and } y_i \\
\mathbf{Programs} \ni \mathsf{pg} & ::= & [\mathsf{nd}_1, \mathsf{nd}_2, \dots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}] \\
& & \quad \text{where } \mathsf{nd}_i, \mathsf{nd}_{\mathsf{Main}} \in \mathbf{Nodes}; \text{ and the node names occurring} \\
& & \quad \text{in } \mathsf{nd}_i \text{ or } \mathsf{nd}_{\mathsf{Main}} \text{ are restricted to } f_1, \dots, f_m \text{ and } f_{\mathsf{Main}}, \\
& & \quad \text{the (distinct) names of } \mathsf{nd}_1, \dots, \mathsf{nd}_m \text{ and } \mathsf{nd}_{\mathsf{Main}}
\end{array}
$$

* Our (textual) language for stream processing
* Simplification of **Lustre**
  [Caspi, Pilaud, Halbwachs, Plaice, POPL'87]
* First-order functional, with recursion

# The Language
## SProc

A program that computes the stream $\texttt{nat} = (0, 1, 2, \ldots)$

$$\texttt{nat} \quad = \quad 0 \ \texttt{fby} \ (1 + \texttt{nat})$$

A program that computes the stream
nat = (0, 1, 2, ...)

(0, 0, 0, ...)

(1, 1, 1, ...)

$$\mathtt{nat} \quad = \quad 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

# The Language
## SProc

A program that computes the stream nat = (0, 1, 2, ...)

(0, 0, 0, ...)

(1, 1, 1, ...)

$$\texttt{nat} \quad = \quad 0 \ \texttt{fby} \ (1 + \texttt{nat})$$

$$(a_0, a_1, a_2, \ldots) \ \texttt{fby} \ (b_0, b_1, b_2, \ldots) := (a_0, b_0, b_1, b_2, \ldots)$$

# The Language SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\texttt{nat} \quad = \quad 0 \ \texttt{fby} \ (1 + \texttt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$(a_0, a_1, a_2, \ldots)$ fby $(b_0, b_1, b_2, \ldots)$
$:= \quad (a_0, b_0, b_1, b_2, \ldots)$

* Operationally:

| | |
|---|---|
| nat | |
| $0$ fby $(1 + \texttt{nat})$ | |
| $1 + \texttt{nat}$ | |

# The Language
## SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \;\mathtt{fby}\; (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$(a_0, a_1, a_2, \ldots)\ \mathtt{fby}\ (b_0, b_1, b_2, \ldots)$
$:= \quad (a_0, b_0, b_1, b_2, \ldots)$

* Operationally:

| | |
|---|---|
| nat | |
| 0 fby (1 + nat) | 0 |
| 1 + nat | |

Hasuo (Tokyo)

Monday, January 28, 2013

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} = 0 \ \mathtt{fby}\ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$(a_0, a_1, a_2, \ldots)$ fby $(b_0, b_1, b_2, \ldots)$
$:= \quad (a_0, b_0, b_1, b_2, \ldots)$

* Operationally:

| | |
|---|---|
| nat | 0 |
| 0 fby (1 + nat) | 0 |
| 1 + nat | |

# The Language
## SProc

A program that computes the stream
nat = (0, 1, 2, ...)

(0, 0, 0, ...)

(1, 1, 1, ...)

$$\mathtt{nat} \quad = \quad 0 \;\; \mathtt{fby} \;\; (1 + \mathtt{nat})$$

$$(a_0, a_1, a_2, \ldots) \;\; \mathtt{fby} \;\; (b_0, b_1, b_2, \ldots)$$
$$:= \quad (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | |
|---|---|
| $\mathtt{nat}$ | 0 |
| $0 \;\mathtt{fby}\; (1 + \mathtt{nat})$ | 0 |
| $1 + \mathtt{nat}$ | 1 |

# The Language
## SProc

A program that
computes the stream
nat = (0, 1, 2, ...)

$$\texttt{nat} \quad = \quad 0 \ \ \texttt{fby} \ \ (1 + \texttt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(a_0, a_1, a_2, \ldots) \ \texttt{fby} \ (b_0, b_1, b_2, \ldots) \\ := \quad (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | | |
|---|---|---|
| $\texttt{nat}$ | 0 | |
| $0 \ \texttt{fby} \ (1 + \texttt{nat})$ | 0 | 1 |
| $1 + \texttt{nat}$ | 1 | |

# The Language SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} \;=\; 0\ \mathtt{fby}\ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$(a_0, a_1, a_2, \ldots)\ \mathtt{fby}\ (b_0, b_1, b_2, \ldots)$
$:= \quad (a_0, b_0, b_1, b_2, \ldots)$

* Operationally:

| | | |
|---|---|---|
| nat | 0 | 1 |
| 0 fby (1 + nat) | 0 | 1 |
| 1 + nat | 1 | |

# The Language
## SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} = 0 \ \mathtt{fby}\ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(a_0, a_1, a_2, \ldots)\ \mathtt{fby}\ (b_0, b_1, b_2, \ldots) := (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | | |
|---|---|---|
| nat | 0 | 1 |
| 0 fby (1 + nat) | 0 | 1 |
| 1 + nat | 1 | 2 |

# The Language
## SProc

A program that computes the stream
nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(a_0, a_1, a_2, \ldots) \ \mathtt{fby} \ (b_0, b_1, b_2, \ldots) := (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | | | |
|---|---|---|---|
| nat | 0 | 1 | |
| 0 fby (1 + nat) | 0 | 1 | 2 |
| 1 + nat | 1 | 2 | |

Hasuo (Tokyo)

# The Language SProc

A program that computes the stream
nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$(a_0, a_1, a_2, \ldots)$ fby $(b_0, b_1, b_2, \ldots)$
$:= \quad (a_0, b_0, b_1, b_2, \ldots)$

* Operationally:

| | | | |
|---|---|---|---|
| nat | 0 | 1 | 2 |
| 0 fby (1 + nat) | 0 | 1 | 2 |
| 1 + nat | 1 | 2 | |

# The Language
## SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$(a_0, a_1, a_2, \dots) \ \mathtt{fby} \ (b_0, b_1, b_2, \dots)$
$:= \quad (a_0, b_0, b_1, b_2, \dots)$

* Operationally:

| | | | |
|---|---|---|---|
| nat | 0 | 1 | 2 |
| 0 fby (1 + nat) | 0 | 1 | 2 |
| 1 + nat | 1 | 2 | 3 |

# The Language SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} = 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(a_0, a_1, a_2, \ldots) \ \mathtt{fby} \ (b_0, b_1, b_2, \ldots) := (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | | | | |
|---|---|---|---|---|
| nat | 0 | 1 | 2 | |
| 0 fby (1 + nat) | 0 | 1 | 2 | 3 |
| 1 + nat | 1 | 2 | 3 | |

# The Language
## SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(a_0, a_1, a_2, \ldots) \ \mathtt{fby} \ (b_0, b_1, b_2, \ldots) := (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | | | | | |
|---|---|---|---|---|---|
| nat | 0 | 1 | 2 | ... | |
| 0 fby (1 + nat) | 0 | 1 | 2 | 3 | ... |
| 1 + nat | 1 | 2 | 3 | | |

# The Language SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \;\; \mathtt{fby} \;\; (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(a_0, a_1, a_2, \ldots) \;\; \mathtt{fby} \;\; (b_0, b_1, b_2, \ldots)$$
$$:= \quad (a_0, b_0, b_1, b_2, \ldots)$$

* Operationally:

| | | | | | |
|---|---|---|---|---|---|
| $\mathtt{nat}$ | 0 | 1 | 2 | ... | |
| $0\ \mathtt{fby}\ (1 + \mathtt{nat})$ | 0 | 1 | 2 | 3 | ... |
| $1 + \mathtt{nat}$ | 1 | 2 | 3 | ... | |

Hasuo (Tokyo)

# The Language
## SProc

A program that computes the stream nat = (0, 1, 2, ...)

$$\mathtt{nat} \quad = \quad 0 \ \mathtt{fby} \ (1 + \mathtt{nat})$$

(0, 0, 0, ...)

(1, 1, 1, ...)

$$(\color{red}{a_0, a_1, a_2, \ldots}) \ \mathtt{fby} \ (\color{blue}{b_0, b_1, b_2, \ldots}) \\ := \quad (\color{red}{a_0}, \color{blue}{b_0, b_1, b_2, \ldots})$$

* Operationally:

| | | | | | |
|---|---|---|---|---|---|
| $\mathtt{nat}$ | 0 | 1 | 2 | ... | |
| $0 \ \mathtt{fby} \ (1 + \mathtt{nat})$ | 0 | 1 | 2 | 3 | ... |
| $1 + \mathtt{nat}$ | 1 | 2 | 3 | ... | |

* We use **Kahn's denotational semantics** [Kahn, '74]
  * $\mathbb{C}^\infty \ := \ \mathbb{C}^* \cup \mathbb{C}^{\mathbb{N}}$ is a cpo

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$<br>Imperative | **SProc**$^{dt}$<br>hyperstream processing language<br>1st-order functional<br>(like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system**<br>(partial correctness) |

* I. Stream Processing Language **SProc**
* II. Nonstandard Analysis
* III. **SProc**$^{dt}$ and Type System
* IV. Hyperstream Sampling

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

- I. Stream Processing Language **SProc**
- II. Nonstandard Analysis
- III. **SProc**$^{dt}$ and Type System
- IV. Hyperstream Sampling

# Contributio



| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While$^{dt}$** Imperative | **SProc$^{dt}$** hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare$^{dt}$** | **Type system** (partial correctness) |

- I. Stream Processing Language **SProc**
- II. Nonstandard Analysis
- III. **SProc$^{dt}$** and Type System
- IV. Hyperstream Sampling

# Part II:

# Nonstandard Analysis

# Nonstandard Analysis

* Analysis with an **infinitesimal** $\partial$

* Done naively ➜ contradiction!

# Nonstandard Analysis

* Analysis with an **infinitesimal** $\partial$

"Infinitely small"
$0 < \partial < r$
$(\forall r \in \mathbb{R}_{+})$

* Done naively ➜ contradiction!

# Nonstandard Analysis

* Analysis with an **infinitesimal** $\partial$

"Infinitely small"
$0 < \partial < r$
$(\forall r \in \mathbb{R}_+)$

* Done naively → contradiction!

**Logical foundation via an ultrafilter**

[Robinson, 1960]

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**

The set of *hyperreal numbers* is

$$ {}^{*}\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

Ignore

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \; := \; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \qquad \ni \; \big[\, (a_0, a_1, a_2, \ldots) \,\big] $$

Ignore

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} := \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \quad \ni [\,(a_0, a_1, a_2, \dots)\,] $$

Ignore

0th section

1st section

2nd section

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \qquad \ni \big[\,(a_0, a_1, a_2, \ldots)\,\big] $$

0th section, 1st section, 2nd section

Ignore

* Operations: sectionwise

$$ \begin{array}{ll} & \big[\,(a_0, a_1, \ldots)\,\big] \\ + & \big[\,(b_0, b_1, \ldots)\,\big] \\ = & \big[\,(a_0 + b_0, a_1 + b_1, \ldots)\,\big] \end{array} $$

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} := \mathbb{R}^{\mathbb{N}} / {\sim_{\mathcal{F}}} \ni [\,(a_0, a_1, a_2, \ldots)\,] $$

Ignore

0th section 1st section 2nd section

- Operations: sectionwise

$$ \begin{array}{ll} & [\,(a_0, a_1, \ldots)\,] \\ + & [\,(b_0, b_1, \ldots)\,] \\ = & [\,(a_0 + b_0, a_1 + b_1, \ldots)\,] \end{array} $$

- (Std.) reals are hyperreals

$$ \mathbb{R} \hookrightarrow {}^*\mathbb{R}, \qquad r \mapsto [\,(r, r, \ldots)\,] $$

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \; := \; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \; \ni \; \big[\, (a_0, a_1, a_2, \ldots) \,\big] $$

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \;\ni\; \big[\,(a_0, a_1, a_2, \ldots)\,\big] $$

* Predicates:
  sectionwise,
  "for almost all $i$"

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \;\ni\; \left[\,(a_0, a_1, a_2, \ldots)\,\right]$$

- <u>Predicates:</u>
  sectionwise,
  "for almost all $i$"

$$\begin{array}{lll} [(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}] & & \\ \iff \quad a_i < b_i & & \text{"for almost every } i\text{"} \\ \impliedby \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} & & \text{is finite} \end{array}$$

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} \;\ni\; \big[\, (a_0, a_1, a_2, \ldots) \,\big] $$

* Predicates:
  sectionwise,
  "for almost all $i$"

$$ \big[(a_i)_{i\in\mathbb{N}}\big] < \big[(b_i)_{i\in\mathbb{N}}\big] $$

$\iff \quad a_i < b_i \quad$ "for almost every $i$"

$\Longleftarrow \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} \quad$ is finite

Precise defn. is via an ultrafilter $\mathcal{F}$ :

$$ \big[(a_i)_{i\in\mathbb{N}}\big] < \big[(b_i)_{i\in\mathbb{N}}\big] $$

$$ \iff \quad \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F} $$

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$

$\iff$ $a_i < b_i$ "for almost every $i$"

$\Longleftarrow$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^{*}\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

$$ [(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}] $$
$$ \iff \quad a_i < b_i \qquad \text{“for almost every } i\text{”} $$
$$ \Longleftarrow \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} \qquad \text{is finite} $$

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$ is infinitesimal.

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$\iff$ $a_i < b_i$ "for almost every $i$"
$\iff$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{N}, \frac{1}{N+1}, \ldots)$$

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N} \qquad, \ldots)$$

Hasuo (Tokyo)

Monday, January 28, 2013

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$

$\iff$ $a_i < b_i$ "for almost every $i$"

$\iff$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

---

**Prop.** $\omega^{-1} = \left[\, (1, \frac{1}{2}, \frac{1}{3}, \ldots) \,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{N}, \frac{1}{N+1}, \ldots)$$

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N}, \ldots)$$

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$

$\iff$ $a_i < b_i$ "for almost every $i$"

$\iff$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{N}, \frac{1}{N+1}, \ldots)$$

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N}, \ldots)$$

Hasuo (Tokyo)

Monday, January 28, 2013

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$\iff \quad a_i < b_i \quad$ "for almost every $i$"
$\iff \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} \quad$ is finite

**Prop.** $\omega^{-1} = \left[\, (1, \frac{1}{2}, \frac{1}{3}, \ldots) \,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{N}, \frac{1}{N+1}, \ldots)$$

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N}, \ldots)$$

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} := \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$\iff$ $a_i < b_i$ "for almost every $i$"
$\Longleftarrow$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \dots)\,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{N}, \frac{1}{N+1}, \dots)$$

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}, \frac{1}{N}, \dots)$$

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

$$ [(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}] $$
$$ \iff \quad a_i < b_i \qquad \text{"for almost every } i\text{"} $$
$$ \Longleftarrow \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} \qquad \text{is finite} $$

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \dots)\,\right]$ is infinitesimal.

$$ \omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{N}, \frac{1}{N+1}, \dots) $$

✗ ✗ ✗ ✗ ✓

$$ \frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}, \frac{1}{N}, \dots) $$

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$$\iff \quad a_i < b_i \qquad \text{“for almost every } i\text{”}$$
$$\Longleftarrow \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} \qquad \text{is finite}$$

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{N}, \frac{1}{N+1}, \ldots)$$

✗ ✗ ✗ ✗ ✓ ✓

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N}, \ldots)$$

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$$^{*}\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$\iff$ $a_i < b_i$ "for almost every $i$"
$\Longleftarrow$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \dots)\,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{N}, \frac{1}{N+1}, \dots)$$

∧ ✗ ∧ ✗ ∧ ✗ ∧ ✗ ∧ ✓ ∧ ✓ ...

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}, \frac{1}{N}, \dots)$$

Hasuo (Tokyo)

# Hyperreals

## = Reals + Infinitesimals + Infinites + ...

**Defn.**
The set of *hyperreal numbers* is

$${}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$\iff$ $a_i < b_i$ "for almost every $i$"
$\Longleftarrow$ $\{i \in \mathbb{N} \mid a_i \not< b_i\}$ is finite

---

**Prop.** $\omega^{-1} = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$ is infinitesimal.

$$\omega^{-1} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{N}, \frac{1}{N+1}, \ldots)$$

OK! ^ ✗ ✗ ✗ ✗ ✓ ✓ ...

$$\frac{1}{N} = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N}, \ldots)$$

# Hyperreals

## = Reals + Infinitesimals + ...

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

$$ [(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}] \iff \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F} $$

# Hype

## = Reals + Inf

## Ultrafilter

(existence by AC)

**Defn.**
An *ultrafilter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is such that:

1. For each $X \subseteq \mathbb{N}$, exactly one of
$$X \quad \text{and} \quad \mathbb{N} \setminus X$$
is in $\mathcal{F}$.
2. $X, Y \in \mathcal{F} \Longrightarrow X \cap Y \in \mathcal{F}$
3. $X \in \mathcal{F}, X \subseteq Y \Longrightarrow Y \in \mathcal{F}$
4. $\emptyset \notin \mathcal{F}$

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}] \iff \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F}$$

Hasuo (Tokyo)

# Hype

## = Reals + Inf

# Ultrafilter

(existence by AC)

**Defn.**
An *ultrafilter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is such that:

1. For each $X \subseteq \mathbb{N}$, exactly one of
$$X \quad \text{and} \quad \mathbb{N} \setminus X$$
is in $\mathcal{F}$.
2. $X, Y \in \mathcal{F} \Longrightarrow X \cap Y \in \mathcal{F}$
3. $X \in \mathcal{F}, X \subseteq Y \Longrightarrow Y \in \mathcal{F}$
4. $\emptyset \notin \mathcal{F}$

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i \in \mathbb{N}}] < [(b_i)_{i \in \mathbb{N}}] \iff \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F}$$

**Thm. (Transfer Principle)**
$A$: a first-order formula in $\mathscr{L}_X$.
$^*A$: its $*$-*transform*. Then

$$\mathbb{R} \models A \iff {}^*\mathbb{R} \models {}^*A \;.$$

Hasuo (Tokyo)

# Hype

## = Reals + Inf

## Ultrafilter

(existence by AC)

**Defn.**
An *ultrafilter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is such that:

1. For each $X \subseteq \mathbb{N}$, exactly one of
$$X \quad \text{and} \quad \mathbb{N} \setminus X$$
is in $\mathcal{F}$.
2. $X, Y \in \mathcal{F} \Longrightarrow X \cap Y \in \mathcal{F}$
3. $X \in \mathcal{F}, X \subseteq Y \Longrightarrow Y \in \mathcal{F}$
4. $\emptyset \notin \mathcal{F}$

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i \in \mathbb{N}}] < [(b_i)_{i \in \mathbb{N}}]$$
$$\Longleftrightarrow \quad \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F}$$

**Thm. (Transfer Principle)**
$A$: a first-order formula in $\mathscr{L}_X$.
$^*A$: its $*$-*transform*. Then

$$\mathbb{R} \models A \quad \Longleftrightarrow \quad {}^*\mathbb{R} \models {}^*A\ .$$

Same as $A$, except:
$\forall x \in \mathbb{R}$ in $A$ is
$\forall x \in {}^*\mathbb{R}$ in $^*A$

Hasuo (Tokyo)

Monday, January 28, 2013

# Hype[...]

## = Reals + Inf[...]

**Ultrafilter**

(existence by AC)

**Defn.**
An *ultrafilter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is such that:

1. For each $X \subseteq \mathbb{N}$, exactly one of
$$X \quad \text{and} \quad \mathbb{N} \setminus X$$
is in $\mathcal{F}$.
2. $X, Y \in \mathcal{F} \Longrightarrow X \cap Y \in \mathcal{F}$
3. $X \in \mathcal{F}, X \subseteq Y \Longrightarrow Y \in \mathcal{F}$
4. $\emptyset \notin \mathcal{F}$

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

$$[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}]$$
$$\Longleftrightarrow \quad \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F}$$

**Thm. (Transfer Principle)**
$A$: a first-order formula in $\mathscr{L}_{\mathbf{X}}$.
$^*A$: its $*$-*transform*. Then

$$\mathbb{R} \models A \quad \Longleftrightarrow \quad {^*\mathbb{R}} \models {^*A}\,.$$

Same as $A$, except:
$\forall x \in \mathbb{R}$ in $A$ is
$\forall x \in {^*\mathbb{R}}$ in $^*A$

* $\mathbb{R}$ and $^*\mathbb{R}$ are "logically the same"
* → transfer program logic too!

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

- I. Stream Processing Language **SProc**
- II. Nonstandard Analysis
- III. **SProc**$^{dt}$ and Type System
- IV. Hyperstream Sampling

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$<br>Imperative | **SProc**$^{dt}$<br>hyperstream processing language<br>1st-order functional<br>(like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system**<br>(partial correctness) |

* I. Stream Processing Language **SProc**
* II. Nonstandard Analysis
* III. **SProc**$^{dt}$ and Type System
* IV. Hyperstream Sampling

# Part III:

## $\mathbf{SProc}^{\mathrm{dt}}$ and Its Type System

# $\mathbf{SProc}^{\mathrm{dt}}$: Syntax

$\mathbf{SProc} + \mathtt{dt} + \mathtt{fby}^{\frac{r}{\mathrm{dt}}}$

$$\mathbf{SExp}_{\mathbb{C}} \ni e \ ::= \ x \mid c \mid e_1 + e_2 \mid e_1 \times e_2 \mid e_1 \mathbin{\hat{}} e_2 \mid e_1 \ \mathtt{fby}^j \ e_2 \mid \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 \mid \mathtt{proj}_k\ f(e_1, \dots, e_m) \mid \mathtt{dt} \mid e_1\ \mathtt{fby}^{\frac{r}{\mathrm{dt}}}\ e_2$$

where $x \in \mathbf{SVar}$; $c \in \mathbb{C}$; $j \in \mathbb{N}$; $b \in \mathbf{SExp}_{\mathbb{B}}$; $f \in \mathbf{NdName}_{m,n}$; $k \in [1, n]$; and $r \in \mathbb{R}_{\geq 0}$ ;

$$\mathbf{SExp}_{\mathbb{B}} \ni b \ ::= \ \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid e_1 = e_2 \mid \mathtt{isReal}(e) \mid e_1 < e_2$$

where $e, e_i \in \mathbf{SExp}_{\mathbb{C}}$

$$\mathbf{Nodes} \ni \mathsf{nd} \ ::= \ \left[ \begin{array}{l} \mathtt{node}\ f(x_1, \dots, x_m)\ \mathtt{returns}\ (e_1, \dots, e_n) \\ \mathtt{where}\ y_1 = e'_1; y_2 = e'_2; \dots ; y_l = e'_l \end{array} \right]$$

where $f \in \mathbf{NdName}_{m,n}$; $x_i, y_i \in \mathbf{SVar}$; $e_i, e'_i \in \mathbf{SExp}_{\mathbb{C}}$; $x_1, \dots, x_m, y_1, \dots, y_n$ are all distinct; and the variables occurring in $e_i, e'_i$ are restricted to $x_i$ and $y_i$

$$\mathbf{Programs} \ni \mathsf{pg} \ ::= \ [\mathsf{nd}_1, \mathsf{nd}_2, \dots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}]$$

where $\mathsf{nd}_i, \mathsf{nd}_{\mathsf{Main}} \in \mathbf{Nodes}$; and the node names occurring in $\mathsf{nd}_i$ or $\mathsf{nd}_{\mathsf{Main}}$ are restricted to $f_1, \dots, f_m$ and $f_{\mathsf{Main}}$, the (distinct) names of $\mathsf{nd}_1, \dots, \mathsf{nd}_m$ and $\mathsf{nd}_{\mathsf{Main}}$

# $\mathbf{SProc}^{\mathtt{dt}}$: Syntax

$$\mathbf{SExp}_{\mathbb{C}} \ni e \ ::= \ x \mid c \mid e_1 + e_2 \mid e_1 \times e_2 \mid e_1 \,\hat{}\, e_2 \mid e_1 \,\mathtt{fby}^j\, e_2 \mid \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 \mid \mathtt{proj}_k\, f(e_1,\dots,e_m) \mid \mathtt{dt} \mid e_1\, \mathtt{fby}^{\frac{r}{\mathtt{dt}}}\, e_2$$

where $x \in \mathbf{SVar}$; $c \in \mathbb{C}$; $j \in \mathbb{N}$; $b \in \mathbf{SExp}_{\mathbb{B}}$; $f \in \mathbf{NdName}_{m,n}$; $k \in [1,n]$; and $r \in \mathbb{R}_{\geq 0}$ ;

$$\mathbf{SExp}_{\mathbb{B}} \ni b \ ::= \ \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid e_1 = e_2 \mid \mathtt{isReal}(e) \mid e_1 < e_2$$

where $e, e_i \in \mathbf{SExp}_{\mathbb{C}}$

$$\mathbf{Nodes} \ni \mathsf{nd} \ ::= \ \left[\begin{array}{l}\mathtt{node}\ f(x_1,\dots,x_m)\ \mathtt{returns}\ (e_1,\dots,e_n) \\ \mathtt{where}\ y_1 = e'_1; y_2 = e'_2; \dots ; y_l = e'_l\end{array}\right]$$

where $f \in \mathbf{NdName}_{m,n}$; $x_i, y_i \in \mathbf{SVar}$; $e_i, e'_i \in \mathbf{SExp}_{\mathbb{C}}$; $x_1,\dots,x_m,y_1,\dots,y_n$ are all distinct; and the variables occurring in $e_i, e'_i$ are restricted to $x_i$ and $y_i$

$$\mathbf{Programs} \ni \mathsf{pg} \ ::= \ [\mathsf{nd}_1, \mathsf{nd}_2, \dots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}]$$

where $\mathsf{nd}_i, \mathsf{nd}_{\mathsf{Main}} \in \mathbf{Nodes}$; and the node names occurring in $\mathsf{nd}_i$ or $\mathsf{nd}_{\mathsf{Main}}$ are restricted to $f_1,\dots,f_m$ and $f_{\mathsf{Main}}$, the (distinct) names of $\mathsf{nd}_1,\dots,\mathsf{nd}_m$ and $\mathsf{nd}_{\mathsf{Main}}$

- Idea: vars/exps denote "**hyperstreams**"
  - = streams with **infinitesimal sampling interval** dt

# $\mathbf{SProc}^{\mathtt{dt}}$: Syntax

$$
\begin{array}{rcl}
\mathbf{SExp}_{\mathbb{C}} \ni e & ::= & x \mid c \mid e_1 + e_2 \mid e_1 \times e_2 \mid e_1 \,\hat{}\, e_2 \mid e_1 \ \mathtt{fby}^{j}\ e_2 \\
& & \mid \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 \mid \mathtt{proj}_k\ f(e_1, \ldots, e_m) \\
& & \mid \mathtt{dt} \mid e_1\ \mathtt{fby}^{\frac{r}{\mathtt{dt}}}\ e_2 \\
& & \text{where } x \in \mathbf{SVar};\ c \in \mathbb{C};\ j \in \mathbb{N};\ b \in \mathbf{SExp}_{\mathbb{B}}; \\
& & \quad f \in \mathbf{NdName}_{m,n};\ k \in [1,n]; \\
& & \quad \text{and } r \in \mathbb{R}_{\geq 0}\ ; \\
\mathbf{SExp}_{\mathbb{B}} \ni b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid e_1 = e_2 \mid \mathtt{isReal}(e) \mid e_1 < e_2 \\
& & \text{where } e, e_i \in \mathbf{SExp}_{\mathbb{C}} \\
\mathbf{Nodes} \ni \mathsf{nd} & ::= & \left[\begin{array}{l} \mathtt{node}\ f(x_1, \ldots, x_m)\ \mathtt{returns}\ (e_1, \ldots, e_n) \\ \mathtt{where}\ y_1 = e'_1; y_2 = e'_2; \ldots ; y_l = e'_l \end{array}\right] \\
& & \text{where } f \in \mathbf{NdName}_{m,n};\ x_i, y_i \in \mathbf{SVar};\ e_i, e'_i \in \mathbf{SExp}_{\mathbb{C}}; \\
& & x_1, \ldots, x_m, y_1, \ldots, y_n \text{ are all distinct; and the variables} \\
& & \text{occurring in } e_i, e'_i \text{ are restricted to } x_i \text{ and } y_i \\
\mathbf{Programs} \ni \mathsf{pg} & ::= & [\mathsf{nd}_1, \mathsf{nd}_2, \ldots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}] \\
& & \text{where } \mathsf{nd}_i, \mathsf{nd}_{\mathsf{Main}} \in \mathbf{Nodes}; \text{ and the node names occurring} \\
& & \text{in } \mathsf{nd}_i \text{ or } \mathsf{nd}_{\mathsf{Main}} \text{ are restricted to } f_1, \ldots, f_m \text{ and } f_{\mathsf{Main}}, \\
& & \text{the (distinct) names of } \mathsf{nd}_1, \ldots, \mathsf{nd}_m \text{ and } \mathsf{nd}_{\mathsf{Main}}
\end{array}
$$

- Idea: vars/exps denote "**hyperstreams**"
  - = streams with **infinitesimal sampling interval** `dt`
- `dt = (dt, dt, ...)` , const. hyperstream

# SProc$^{\mathrm{dt}}$: Syntax

$\mathbf{SExp}_{\mathbb{C}} \ni e \;::=\; x \mid c \mid e_1 + e_2 \mid e_1 \times e_2 \mid e_1 \,\hat{}\, e_2 \mid e_1 \,\mathtt{fby}^j\, e_2 \mid \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 \mid \mathtt{proj}_k\, f(e_1, \ldots, e_m) \mid \mathtt{dt} \mid e_1\, \mathtt{fby}^{r}_{\mathtt{dt}}\, e_2$

where $x \in \mathbf{SVar}$; $c \in \mathbb{C}$; $j \in \mathbb{N}$; $b \in \mathbf{SExp}_{\mathbb{B}}$; $f \in \mathbf{NdName}_{m,n}$; $k \in [1, n]$; and $r \in \mathbb{R}_{\geq 0}$ ;

$\mathbf{SExp}_{\mathbb{B}} \ni b \;::=\; \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid e_1 = e_2 \mid \mathtt{isReal}(e) \mid e_1 < e_2$

where $e, e_i \in \mathbf{SExp}_{\mathbb{C}}$

$\mathbf{Nodes} \ni \mathsf{nd} \;::=\; \left[\begin{array}{l}\mathtt{node}\ f(x_1, \ldots, x_m)\ \mathtt{returns}\ (e_1, \ldots, e_n)\\ \mathtt{where}\ y_1 = e'_1; y_2 = e'_2; \ldots ; y_l = e'_l\end{array}\right]$

where $f \in \mathbf{NdName}_{m,n}$; $x_i, y_i \in \mathbf{SVar}$; $e_i, e'_i \in \mathbf{SExp}_{\mathbb{C}}$; $x_1, \ldots, x_m, y_1, \ldots, y_n$ are all distinct; and the variables occurring in $e_i, e'_i$ are restricted to $x_i$ and $y_i$

$\mathbf{Programs} \ni \mathsf{pg} \;::=\; [\mathsf{nd}_1, \mathsf{nd}_2, \ldots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}]$

where $\mathsf{nd}_i, \mathsf{nd}_{\mathsf{Main}} \in \mathbf{Nodes}$; and the node names occurring in $\mathsf{nd}_i$ or $\mathsf{nd}_{\mathsf{Main}}$ are restricted to $f_1, \ldots, f_m$ and $f_{\mathsf{Main}}$, the (distinct) names of $\mathsf{nd}_1, \ldots, \mathsf{nd}_m$ and $\mathsf{nd}_{\mathsf{Main}}$

- Idea: vars/exps denote "**hyperstreams**"
  - = streams with **infinitesimal sampling interval** dt
- dt = (dt, dt, ...) , const. hyperstream
- $(a_0, a_1, a_2, \ldots)\ \mathtt{fby}\ (b_0, b_1, b_2, \ldots) := (a_0, b_0, b_1, b_2, \ldots)$

Delay by **1 step** (dt seconds)

$\mathtt{fby}^{r}_{\mathrm{dt}}$

Delay by **r seconds** (infinite steps)

Hasuo (Tokyo)

# $\mathbf{SProc}^{\mathrm{dt}}$: Example (The Sine Curve)

$$\begin{array}{lll} s & = & 0 \ \mathtt{fby}\ (s + c \times \mathtt{dt})\ ; \\ c & = & 1 \ \mathtt{fby}\ (c - s \times \mathtt{dt}) \end{array}$$

# $\mathrm{SProc}^{\mathrm{dt}}$: Example (The Sine Curve)

$$
\begin{aligned}
s &= 0\ \mathrm{fby}\ (s + c \times \mathrm{dt})\ ; \\
c &= 1\ \mathrm{fby}\ (c - s \times \mathrm{dt})
\end{aligned}
$$

$$
\begin{aligned}
&(s_0, s_1, s_2, \ldots) \\
&= (0,\ s_0 + c_0 \times \mathrm{dt},\ s_1 + c_1 \times \mathrm{dt},\ \ldots)
\end{aligned}
$$

# $\mathbf{SProc}^{\mathrm{dt}}$: Example (The Sine Curve)

$$s = 0 \text{ fby } (s + c \times \mathrm{dt})\ ;$$
$$c = 1 \text{ fby } (c - s \times \mathrm{dt})$$

$$(s_0, s_1, s_2, \ldots) = (0,\ s_0 + c_0 \times \mathrm{dt},\ s_1 + c_1 \times \mathrm{dt},\ \ldots)$$

➥ $s_0 = 0\ ; \quad s_{n+1} = s_n + c_n \times \mathrm{dt}$

# $\mathrm{SProc}^{\mathrm{dt}}$: Example (The Sine Curve)

$$\begin{aligned} s &= 0 \text{ fby } (s + c \times \mathrm{dt}) \; ; \\ c &= 1 \text{ fby } (c - s \times \mathrm{dt}) \end{aligned}$$

$$(s_0, s_1, s_2, \ldots) = \big(0,\; s_0 + c_0 \times \mathrm{dt},\; s_1 + c_1 \times \mathrm{dt},\; \ldots\big)$$

➥ $s_0 = 0 \; ; \quad s_{n+1} = s_n + c_n \times \mathrm{dt}$

➥ $s_0 = 0 \; ; \quad \dfrac{s(t + \mathrm{dt}) - s(t)}{\mathrm{dt}} = c(t)$

# SProc$^{\mathrm{dt}}$: Example (The Sine Curve)

$$
\begin{aligned}
s &= 0 \;\mathrm{fby}\; (s + c \times \mathrm{dt}) \;; \\
c &= 1 \;\mathrm{fby}\; (c - s \times \mathrm{dt})
\end{aligned}
$$

$$
(s_0, s_1, s_2, \ldots) = \big(0,\; s_0 + c_0 \times \mathrm{dt},\; s_1 + c_1 \times \mathrm{dt},\; \ldots\big)
$$

$$\hookrightarrow \quad s_0 = 0 \;; \quad s_{n+1} = s_n + c_n \times \mathrm{dt}$$

$$\hookrightarrow \quad s_0 = 0 \;; \quad \frac{s(t + \mathrm{dt}) - s(t)}{\mathrm{dt}} = c(t)$$

$$\hookrightarrow \quad s_0 = 0 \;; \quad \dot{s}(t) = c(t)$$

# SProc$^{\mathrm{dt}}$: Example (The Sine Curve)

$$s \;=\; 0 \;\mathtt{fby}\; (s + c \times \mathrm{dt})\;;$$
$$c \;=\; 1 \;\mathtt{fby}\; (c - s \times \mathrm{dt})$$

$s(0) = 0 \qquad c(0) = 1$

$\dot{s}(t) = c(t) \qquad \dot{c}(t) = -s(t)$

$$(s_0, s_1, s_2, \ldots) = (0,\; s_0 + c_0 \times \mathrm{dt},\; s_1 + c_1 \times \mathrm{dt},\; \ldots)$$

➥ $s_0 = 0 \;; \quad s_{n+1} = s_n + c_n \times \mathrm{dt}$

➥ $s_0 = 0 \;; \quad \dfrac{s(t+\mathrm{dt}) - s(t)}{\mathrm{dt}} = c(t)$

➥ $s_0 = 0 \;; \quad \dot{s}(t) = c(t)$

# $\mathbf{SProc}^{\mathrm{dt}}$: Denotational Semantics (Sketch)

$$\begin{aligned} s &= 0 \ \mathrm{fby}\ (s + c \times \mathrm{dt})\ ; \\ c &= 1 \ \mathrm{fby}\ (c - s \times \mathrm{dt}) \end{aligned}$$

# $\mathbf{SProc}^{\mathrm{dt}}$: Denotational Semantics (Sketch)

Cf.

$$[\![\mathrm{dt}]\!] = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$$

$$\begin{aligned} s &= 0\ \mathtt{fby}\ (s + c \times \mathrm{dt})\ ; \\ c &= 1\ \mathtt{fby}\ (c - s \times \mathrm{dt}) \end{aligned}$$

# $\mathbf{SProc}^{\mathrm{dt}}$: Denotational Semantics (Sketch)

Cf.

$$[\![\mathrm{dt}]\!] = \left[\, (1, \tfrac{1}{2}, \tfrac{1}{3}, \ldots) \,\right]$$

$$\begin{array}{lcl} s & = & 0 \,\mathtt{fby}\, (s + c \times \mathrm{dt}) \;; \\ c & = & 1 \,\mathtt{fby}\, (c - s \times \mathrm{dt}) \end{array}$$

"expand into sections"

$$\left( \begin{array}{lcl} s & = & 0 \,\mathtt{fby}\, (s + c \times \frac{1}{1}) \;; \\ c & = & 1 \,\mathtt{fby}\, (c - s \times \frac{1}{1}) \end{array} , \begin{array}{lcl} s & = & 0 \,\mathtt{fby}\, (s + c \times \frac{1}{2}) \;; \\ c & = & 1 \,\mathtt{fby}\, (c - s \times \frac{1}{2}) \end{array} , \begin{array}{lcl} s & = & 0 \,\mathtt{fby}\, (s + c \times \frac{1}{3}) \;; \\ c & = & 1 \,\mathtt{fby}\, (c - s \times \frac{1}{3}) \end{array} \ldots \right)$$

# $\mathrm{SProc}^{\mathrm{dt}}$: Denotational Semantics (Sketch)

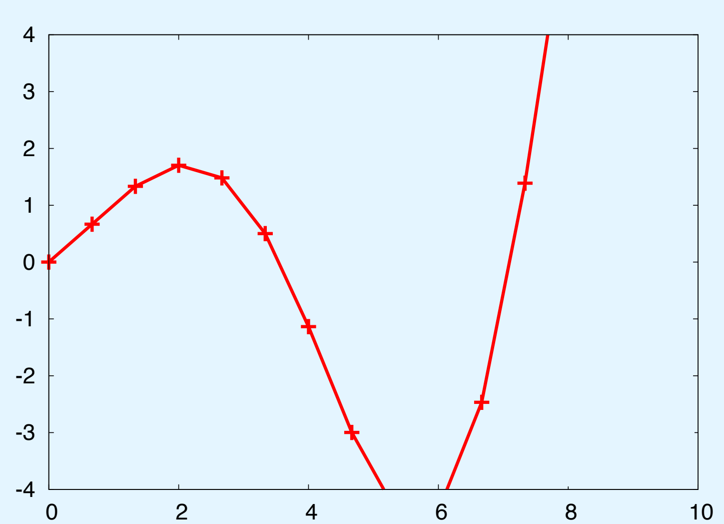
Cf.

$$[\![\mathrm{dt}]\!] = \left[\,\left(1, \frac{1}{2}, \frac{1}{3}, \ldots\right)\right]$$

$$\begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \mathrm{dt})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \mathrm{dt}) \end{aligned}$$

"expand into sections"

$$\left( \begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \tfrac{1}{1})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \tfrac{1}{1}) \end{aligned}\ ,\ \begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \tfrac{1}{2})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \tfrac{1}{2}) \end{aligned}\ ,\ \begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \tfrac{1}{3})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \tfrac{1}{3}) \end{aligned}\ \ldots \right)$$




Hasuo (Tokyo)

# SProc$^{\mathrm{dt}}$: Denotational Semantics (Sketch)

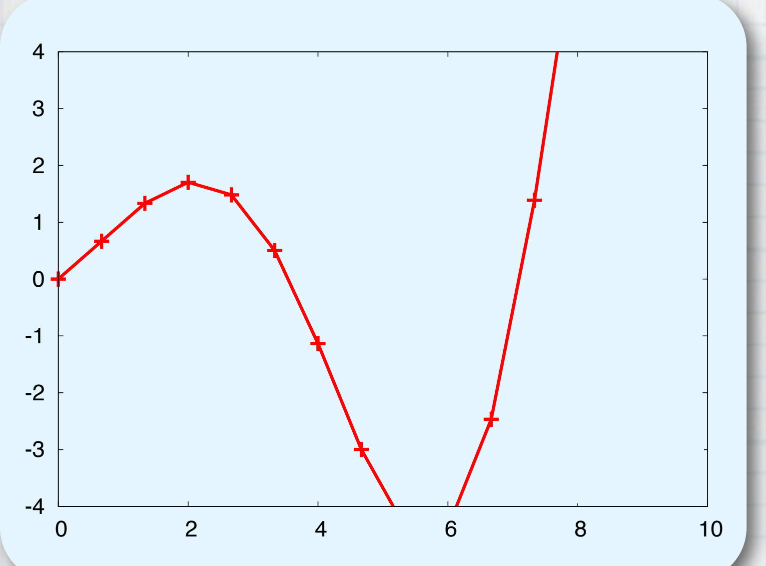
Cf.

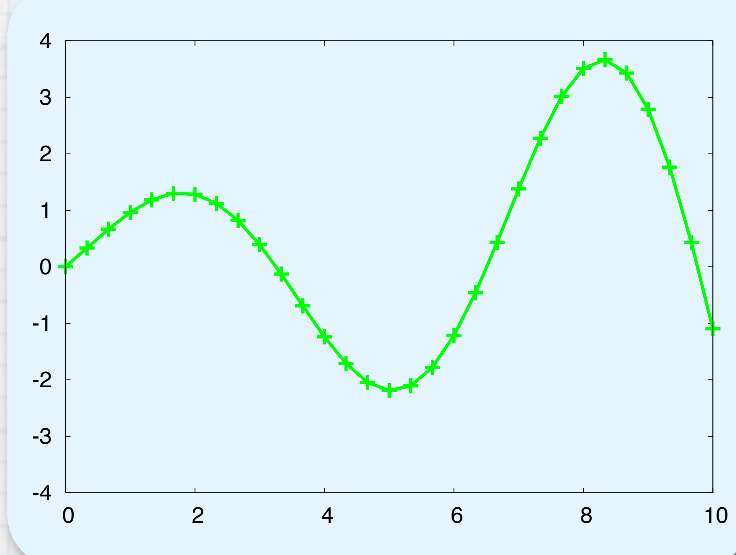
$$[\![\mathrm{dt}]\!] = \left[\,(1, \frac{1}{2}, \frac{1}{3}, \ldots)\,\right]$$

$$\begin{aligned} s &= 0 \text{ fby } (s + c \times \mathrm{dt})\ ; \\ c &= 1 \text{ fby } (c - s \times \mathrm{dt}) \end{aligned}$$

"expand into sections"

$$\left( \begin{aligned} s &= 0 \text{ fby } (s + c \times \tfrac{1}{1})\ ; \\ c &= 1 \text{ fby } (c - s \times \tfrac{1}{1}) \end{aligned}, \quad \begin{aligned} s &= 0 \text{ fby } (s + c \times \tfrac{1}{2})\ ; \\ c &= 1 \text{ fby } (c - s \times \tfrac{1}{2}) \end{aligned}, \quad \begin{aligned} s &= 0 \text{ fby } (s + c \times \tfrac{1}{3})\ ; \\ c &= 1 \text{ fby } (c - s \times \tfrac{1}{3}) \end{aligned} \ \ldots \right)$$



# SProc$^{\mathrm{dt}}$: Denotational Semantics (Sketch)

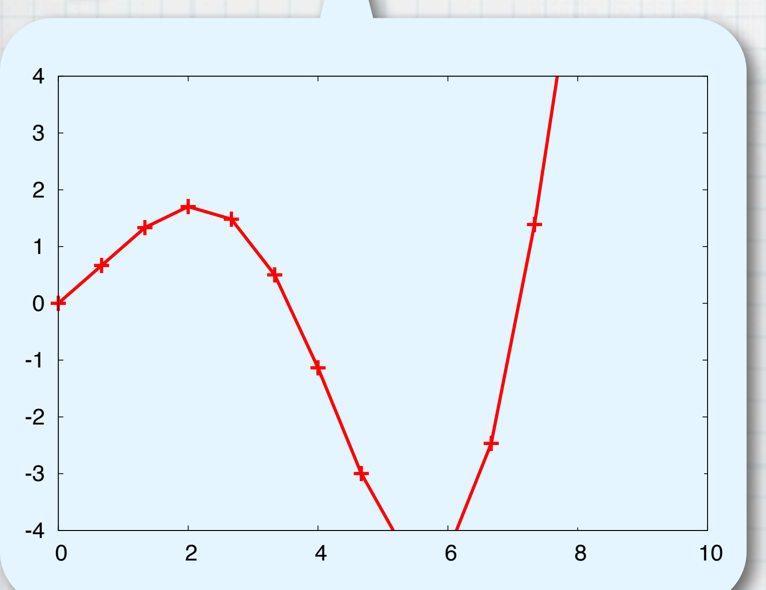
Cf.

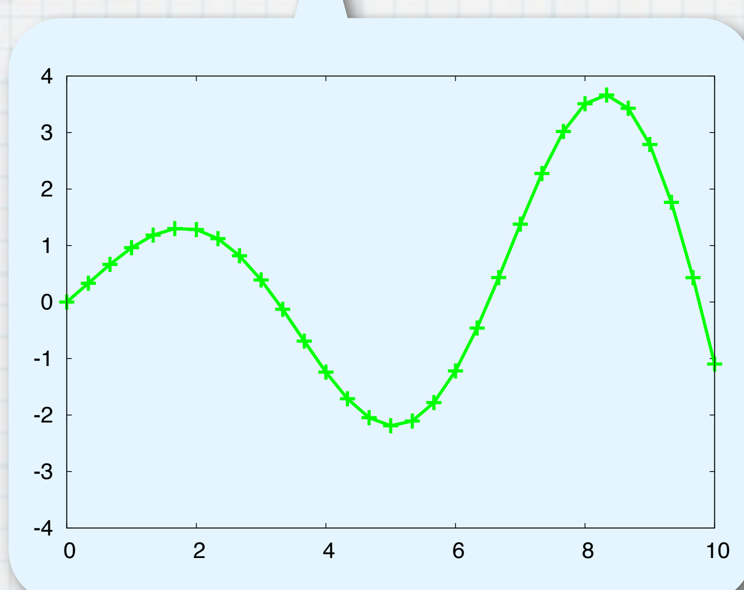
$$[\![\mathrm{dt}]\!] = \left[\, (1, \frac{1}{2}, \frac{1}{3}, \ldots) \,\right]$$

$$\begin{aligned} s &= 0 \;\mathtt{fby}\; (s + c \times \mathrm{dt}) \;; \\ c &= 1 \;\mathtt{fby}\; (c - s \times \mathrm{dt}) \end{aligned}$$

"expand into sections"

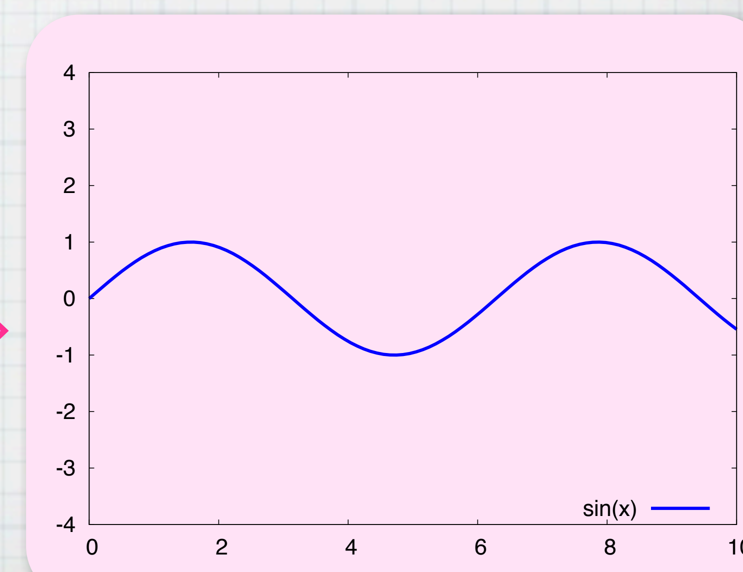
$$\left( \begin{aligned} s &= 0 \;\mathtt{fby}\; (s + c \times \tfrac{1}{1}) \;; \\ c &= 1 \;\mathtt{fby}\; (c - s \times \tfrac{1}{1}) \end{aligned} \;,\; \begin{aligned} s &= 0 \;\mathtt{fby}\; (s + c \times \tfrac{1}{2}) \;; \\ c &= 1 \;\mathtt{fby}\; (c - s \times \tfrac{1}{2}) \end{aligned} \;,\; \begin{aligned} s &= 0 \;\mathtt{fby}\; (s + c \times \tfrac{1}{3}) \;; \\ c &= 1 \;\mathtt{fby}\; (c - s \times \tfrac{1}{3}) \end{aligned} \;\ldots \right)$$




...

"limit"



* Intuitively: like **Euler's method**
* Precisely: via *-transfer
  (an NSA operation, with ultrafilter)

# l Semantics

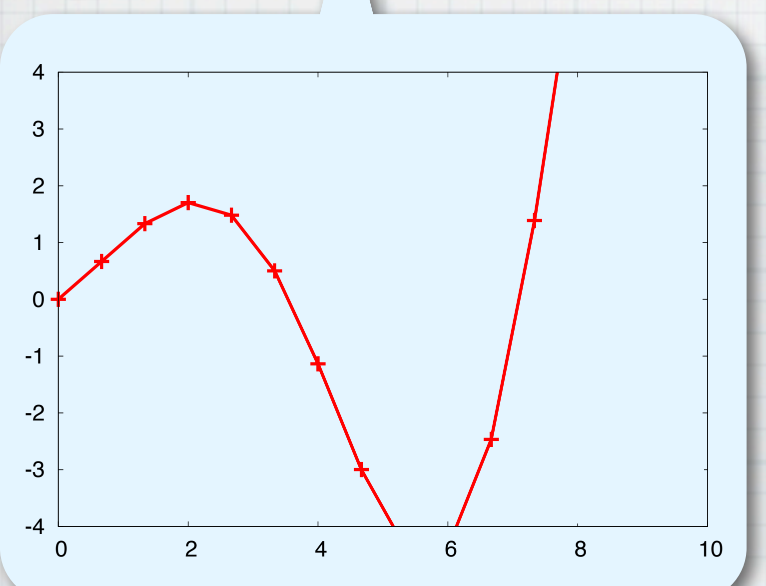
Cf.

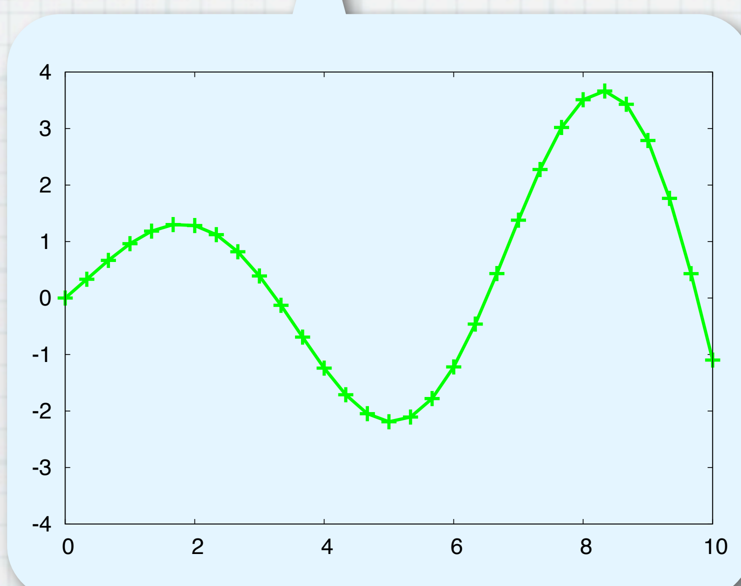
$$[\![\mathrm{dt}]\!] = \left[\, (1, \frac{1}{2}, \frac{1}{3}, \ldots) \,\right]$$

$$\begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \mathrm{dt})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \mathrm{dt}) \end{aligned}$$

"expand into sections"

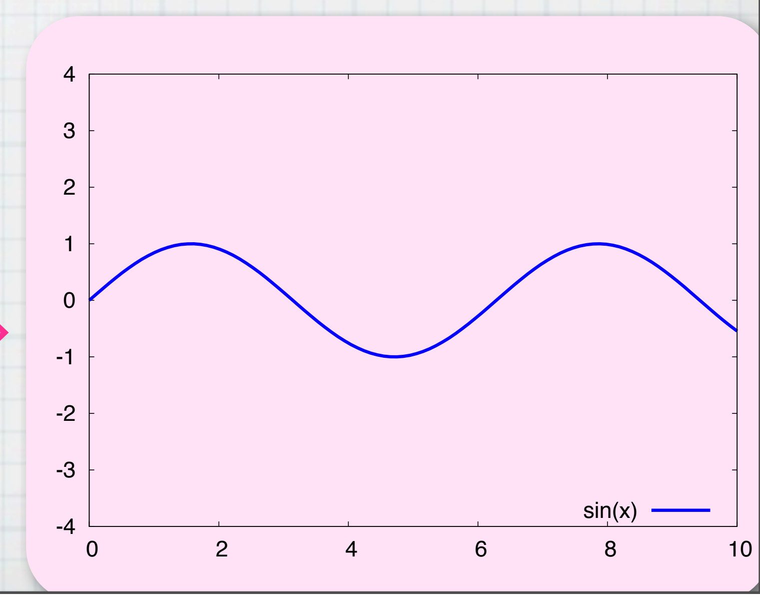
$$\left( \begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \tfrac{1}{1})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \tfrac{1}{1}) \end{aligned} \ ,\ \begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \tfrac{1}{2})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \tfrac{1}{2}) \end{aligned} \ ,\ \begin{aligned} s &= 0 \ \mathtt{fby}\ (s + c \times \tfrac{1}{3})\ ; \\ c &= 1 \ \mathtt{fby}\ (c - s \times \tfrac{1}{3}) \end{aligned} \ \ldots \right)$$




...

"limit"



# SProc$^{dt}$: Type System (For Partial Correctness)

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While$^{dt}$** Imperative | **SProc$^{dt}$** hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare$^{dt}$** | **Type system** (partial correctness) |

# SProc$^{dt}$: Type System (For Partial Correctness)

* That for SProc, "*-transferred"

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

# SProc$^{dt}$: Type System (For Partial Correctness)

* That for **SProc**, "*-transferred"
* Syntax after dependent type systems

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

# $\mathbf{SProc}^{\mathrm{dt}}$: Type System (For Partial Correctness)

* That for **SProc**, "*-transferred"
* Syntax after dependent type systems
* Example:

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{\mathrm{dt}}$ Imperative | **SProc**$^{\mathrm{dt}}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{\mathrm{dt}}$ | **Type system** (partial correctness) |

$$\vdash \left[\begin{array}{ll} \mathtt{node} & \mathtt{Sine()\ returns}\ (s) \\ \mathtt{where} & s = \mathbf{0}\ \mathtt{fby}\ (s + c \times \mathtt{dt}); \\ & c = \mathbf{1}\ \mathtt{fby}\ (c - s \times \mathtt{dt}) \end{array}\right] : \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid t_0 \le v \times \mathtt{dt} \vee u \le 1 + \varepsilon\} .$$

# SProc$^{\mathrm{dt}}$: Type System (For Partial Correctness)

$$
\begin{array}{rcl}
\mathbf{AExp} \ni a & ::= & v \mid c \mid a_1 + a_2 \mid a_1 \times a_2 \mid a_1 \mathbin{\char`\^} a_2 \mid \lceil a_1 \rceil \mid \\
& & \mathtt{dt} \mid \frac{1}{\mathtt{dt}} \quad \text{where } v \in \mathbf{Var} \text{ and } c \in \mathbb{C} \\
\mathbf{Fml} \ni P & ::= & \mathtt{true} \mid \mathtt{false} \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \neg P \mid \\
& & a_1 = a_2 \mid \mathtt{isReal}(a) \mid a_1 < a_2 \mid a_1 \leq a_2 \mid \\
& & \forall v \in {}^*\mathbb{N}.\, P \mid \forall v \in {}^*\mathbb{C}.\, P \\
& & \quad \text{where } v \in \mathbf{Var} \text{ and } a, a_i \in \mathbf{AExp} \\
\mathbf{SType}_{\mathbb{C}} \ni \tau & ::= & \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\} \quad \text{where } u, v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \{u, v\} \\
\mathbf{SType}_{\mathbb{B}} \ni \beta & ::= & \prod_{v \in {}^*\mathbb{N}} P \quad \text{where } v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \{v\} \\
\mathbf{NdType}_{m,n} \ni \nu & ::= & (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \tau'_n) \\
& & \quad \text{where } \tau_i, \tau'_i \in \mathbf{SType}_{\mathbb{C}}
\end{array}
$$

# SProc$^{\mathrm{dt}}$: Type System (For Partial Correctness)

$$
\begin{array}{rcl}
\mathbf{AExp} \ni a & ::= & v \mid c \mid a_1 + a_2 \mid a_1 \times a_2 \mid a_1 \mathbin{\hat{}} a_2 \mid \lceil a_1 \rceil \mid \\
 & & \mathtt{dt} \mid \frac{1}{\mathtt{dt}} \quad \text{where } v \in \mathbf{Var} \text{ and } c \in \mathbb{C} \\
\mathbf{Fml} \ni P & ::= & \mathtt{true} \mid \mathtt{false} \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \neg P \mid \\
 & & a_1 = a_2 \mid \mathtt{isReal}(a) \mid a_1 < a_2 \mid a_1 \le a_2 \mid \\
 & & \forall v \in {}^*\mathbb{N}.\, P \mid \forall v \in {}^*\mathbb{C}.\, P \\
 & & \quad \text{where } v \in \mathbf{Var} \text{ and } a, a_i \in \mathbf{AExp} \\
\mathbf{SType}_{\mathbb{C}} \ni \tau & ::= & \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\} \quad \text{where } u, v \in \mathbf{Var}, \\
 & & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \{u, v\} \\
\mathbf{SType}_{\mathbb{B}} \ni \beta & ::= & \prod_{v \in {}^*\mathbb{N}} P \quad \text{where } v \in \mathbf{Var}, \\
 & & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \ \ldots \\
\mathbf{NdType}_{m,n} \ni \nu & ::= & (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \\
 & & \quad \text{where } \tau_i, \tau'_i \in \mathbf{SType}_{\mathbb{C}} \ldots
\end{array}
$$

$$\frac{}{\Delta;\Gamma \vdash x : \Gamma(x)}\ (\textsc{SVar}) \qquad \frac{}{\Delta;\Gamma \vdash c : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid u = c\}}\ (\textsc{Const})$$

$$\frac{\Delta;\Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}.\forall u_1, u_2, u \in \mathbb{C}.(P_1 \wedge P_2 \wedge u = (u_1 \ \mathsf{aop}\ u_2) \ \Rightarrow P)}{\Delta;\Gamma \vdash e_1 \ \mathsf{aop}\ e_2 : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}}\ (\textsc{AOp}) \quad (\mathsf{aop} \in \{+, \times, \hat{}\})$$

$$\frac{\Delta;\Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}.\forall u \in \mathbb{C}.(\,(v < j \wedge P_1 \ \Rightarrow P) \wedge (v \ge j \wedge P_2[v - j/v] \ \Rightarrow P)\,)}{\Delta;\Gamma \vdash e_1 \ \mathtt{fby}^j\ e_2 : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}}\ (\textsc{Fby}^j)$$

$$\frac{\Delta;\Gamma \vdash b : \prod_{v\in\mathbb{N}} P_b \quad \Delta;\Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}.\forall u \in \mathbb{C}.(P_b \wedge P_1 \Rightarrow P) \wedge (P_b \wedge P_2 \Rightarrow P)}{\Delta;\Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}}\ (\textsc{If})$$

$$\frac{\Delta;\Gamma \vdash e_i : \tau_i \text{ for } i \in [1,m] \quad \Delta(f) = (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \tau'_n)}{\Delta;\Gamma \vdash \mathsf{proj}_k\, f(e_1, \ldots, e_m) : \tau'_k}\ (\textsc{NdCall})$$

$$\frac{\Delta;\Gamma \vdash e : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}.\forall u \in \mathbb{C}.\, P' \Rightarrow P}{\Delta;\Gamma \vdash e : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}}\ (\textsc{CStConseq})$$

---

$$\frac{\Delta;\Gamma \vdash b_i : \prod_{v\in\mathbb{N}} P_i \text{ for } i = 1,2}{\Delta;\Gamma \vdash b_1 \wedge b_2 : \prod_{v\in\mathbb{N}}(P_1 \wedge P_2) \text{ for } i = 1,2}\ (\textsc{And}) \qquad (\textsc{True}), (\textsc{False}), (\textsc{Neg}) \text{ are similar}$$

$$\frac{\Delta;\Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}.\forall u_1, u_2 \in \mathbb{C}.(P_1 \wedge P_2 \Rightarrow (P \Leftrightarrow u_1 = u_2))}{\Delta;\Gamma \vdash e_1 = e_2 : \prod_{v\in\mathbb{N}} P}\ (\textsc{Equal})$$

$$\frac{\Delta;\Gamma \vdash e : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}.\forall u \in \mathbb{C}.(P' \ \Rightarrow (P \Leftrightarrow \mathtt{isReal}(u)))}{\Delta;\Gamma \vdash \mathtt{isReal}(e) : \prod_{v\in\mathbb{N}} P}\ (\textsc{IsReal})$$

$$\frac{\Delta;\Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}.\forall u_1, u_2 \in \mathbb{C}.(P_1 \wedge P_2 \ \Rightarrow (P \Leftrightarrow (\mathtt{isReal}(u_1) \wedge \mathtt{isReal}(u_2) \wedge u_1 < u_2)))}{\Delta;\Gamma \vdash e_1 < e_2 : \prod_{v\in\mathbb{N}} P}\ (\textsc{Less})$$

$$\frac{\Delta;\Gamma \vdash b : \prod_{v\in\mathbb{N}} P' \quad \models \forall v \in \mathbb{N}.\, P' \Leftrightarrow P}{\Delta;\Gamma \vdash b : \prod_{v\in\mathbb{N}} P}\ (\textsc{BStConseq})$$

---

$$\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1,m] \quad \Delta;\Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1,l] \quad \Delta;\Gamma \vdash e_k : \tau''_k \text{ for } k \in [1,n]}{\Delta \vdash \left[\begin{array}{l}\mathtt{node}\ f(x_1, \ldots, x_m)\ \mathtt{returns}\ (e_1, \ldots, e_n) \\ \mathtt{where}\ y_1 = e'_1; \ldots; y_l = e'_l\end{array}\right] : (\tau_1, \ldots, \tau_m) \to (\tau''_1, \ldots, \tau''_n)}\ (\textsc{Node})$$

$$\frac{\begin{array}{l}\Delta \vdash \mathsf{nd} : (\tau'_1, \ldots, \tau'_m) \to (\sigma'_1, \ldots, \sigma'_n) \\ \models \forall v \in \mathbb{N}.\, \forall u \in \mathbb{C}.\, P_i \Rightarrow P'_i \text{ for } i \in [1,m] \qquad \models \forall v \in \mathbb{N}.\, \forall u \in \mathbb{C}.\, Q'_j \Rightarrow Q_j \text{ for } j \in [1,n] \\ (\text{where } \tau_i \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P_i\},\ \tau'_i \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P'_i\},\ \sigma_j \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid Q_j\},\ \sigma'_j \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid Q'_j\})\end{array}}{\Delta \vdash \mathsf{nd} : (\tau_1, \ldots, \tau_m) \to (\sigma_1, \ldots, \sigma_n)}\ (\textsc{NdConseq})$$

---

$$\frac{\Delta \vdash \mathsf{nd}_i : \Delta(f_i) \text{ for } i \in [1,m] \quad \Delta \vdash \mathsf{nd}_{\mathsf{Main}} : \nu}{\vdash [\mathsf{nd}_1, \ldots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}] : \nu}\ (\textsc{Prog}) \quad (f_i \text{ is the name of the nodes } \mathsf{nd}_i)$$

$$\frac{}{\Delta;\Gamma \vdash \mathtt{dt} : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid u = \mathtt{dt}\}}\ (\mathtt{dt})$$

$$\frac{\begin{array}{l}\Delta;\Gamma \vdash e_1 : \prod_v\{u \mid P_1\} \qquad \Delta;\Gamma \vdash e_2 : \prod_v\{u \mid P_2\} \\ \models \forall v \in {}^*\mathbb{N}.\forall u \in {}^*\mathbb{C}.(\,(v < \frac{r}{\mathtt{dt}} \wedge P_1 \ \Rightarrow P) \wedge \\ \qquad (v \ge \frac{r}{\mathtt{dt}} \wedge P_2[(v - \lceil \frac{r}{\mathtt{dt}} \rceil)/v] \ \Rightarrow P)\,)\end{array}}{\Delta;\Gamma \vdash e_1\ \mathtt{fby}^{\frac{r}{\mathtt{dt}}}\ e_2 : \prod_{v\in{}^*\mathbb{N}}\{u \in {}^*\mathbb{C} \mid P\}}\ (\textsc{Fby}^{\frac{r}{\mathtt{dt}}})$$

Hasuo (Tokyo)

# SProc$^{\mathrm{dt}}$: Type System (For Partial Correctness)

$$
\begin{array}{rcl}
\mathbf{AExp} \ni a & ::= & v \mid c \mid a_1 + a_2 \mid a_1 \times a_2 \mid a_1 \mathbin{\hat{}} a_2 \mid \lceil a_1 \rceil \mid \\
& & \mathtt{dt} \mid \frac{1}{\mathtt{dt}} \quad \text{where } v \in \mathbf{Var} \text{ and } c \in \mathbb{C} \\
\mathbf{Fml} \ni P & ::= & \mathtt{true} \mid \mathtt{false} \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \neg P \mid \\
& & a_1 = a_2 \mid \mathtt{isReal}(a) \mid a_1 < a_2 \mid a_1 \le a_2 \mid \\
& & \forall v \in {}^*\mathbb{N}.\, P \mid \forall v \in {}^*\mathbb{C}.\, P \\
& & \quad \text{where } v \in \mathbf{Var} \text{ and } a, a_i \in \mathbf{AExp} \\
\mathbf{SType}_{\mathbb{C}} \ni \tau & ::= & \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\} \quad \text{where } u, v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \{u, v\} \\
\mathbf{SType}_{\mathbb{B}} \ni \beta & ::= & \prod_{v \in {}^*\mathbb{N}} P \quad \text{where } v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \\
\mathbf{NdType}_{m,n} \ni \nu & ::= & (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \\
& & \quad \text{where } \tau_i, \tau'_i \in \mathbf{SType}_{\mathbb{C}}
\end{array}
$$

$$
\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)} \ (\textsc{SVar})
$$

$$
\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)} \ (\textsc{SVar}) \qquad \frac{}{\Delta; \Gamma \vdash c : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid u = c\}} \ (\textsc{Const})
$$

$$
\frac{\Delta; \Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u_1, u_2, u \in \mathbb{C}. (P_1 \wedge P_2 \wedge u = (u_1 \ \mathtt{aop}\ u_2) \Rightarrow P)}{\Delta; \Gamma \vdash e_1 \ \mathtt{aop}\ e_2 : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}} \ (\textsc{AOp}) \quad (\mathtt{aop} \in \{+, \times, \hat{}\,\})
$$

$$
\frac{\Delta; \Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. \big((v < j \wedge P_1 \Rightarrow P) \wedge (v \ge j \wedge P_2[v - j/v] \Rightarrow P)\big)}{\Delta; \Gamma \vdash e_1 \ \mathtt{fby}^j\ e_2 : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}} \ (\textsc{Fby}^j)
$$

$$
\frac{\Delta; \Gamma \vdash b : \prod_{v\in\mathbb{N}} P_b \quad \Delta; \Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. (P_b \wedge P_1 \Rightarrow P) \wedge (P_{\bar b} \wedge P_2 \Rightarrow P)}{\Delta; \Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}} \ (\textsc{If})
$$

$$
\frac{\Delta; \Gamma \vdash e_i : \tau_i \text{ for } i \in [1, m] \quad \Delta(f) = (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \tau'_n)}{\Delta; \Gamma \vdash \mathtt{proj}_k\ f(e_1, \ldots, e_m) : \tau'_k} \ (\textsc{NdCall})
$$

$$
\frac{\Delta; \Gamma \vdash e : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, P' \Rightarrow P}{\Delta; \Gamma \vdash e : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P\}} \ (\textsc{CstConseq})
$$

---

$$
\frac{\Delta; \Gamma \vdash b_i : \prod_{v\in\mathbb{N}} P_i \text{ for } i = 1,2}{\Delta; \Gamma \vdash b_1 \wedge b_2 : \prod_{v\in\mathbb{N}} (P_1 \wedge P_2) \text{ for } i = 1,2} \ (\textsc{And}) \qquad (\textsc{True}), (\textsc{False}), (\textsc{Neg}) \text{ are similar}
$$

$$
\frac{\Delta; \Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u_1, u_2 \in \mathbb{C}. (P_1 \wedge P_2 \Rightarrow (P \Leftrightarrow u_1 = u_2))}{\Delta; \Gamma \vdash e_1 = e_2 : \prod_{v\in\mathbb{N}} P} \ (\textsc{Equal})
$$

$$
\frac{\Delta; \Gamma \vdash e : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. (P' \Rightarrow (P \Leftrightarrow \mathtt{isReal}(u)))}{\Delta; \Gamma \vdash \mathtt{isReal}(e) : \prod_{v\in\mathbb{N}} P} \ (\textsc{IsReal})
$$

$$
\frac{\Delta; \Gamma \vdash e_i : \prod_{v\in\mathbb{N}}\{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u_1, u_2 \in \mathbb{C}. (P_1 \wedge P_2 \Rightarrow (P \Leftrightarrow (\mathtt{isReal}(u_1) \wedge \mathtt{isReal}(u_2) \wedge u_1 < u_2)))}{\Delta; \Gamma \vdash e_1 < e_2 : \prod_{v\in\mathbb{N}} P} \ (\textsc{Less})
$$

$$
\frac{\Delta; \Gamma \vdash b : \prod_{v\in\mathbb{N}} P' \quad \models \forall v \in \mathbb{N}.\, P' \Leftrightarrow P}{\Delta; \Gamma \vdash b : \prod_{v\in\mathbb{N}} P} \ (\textsc{BstConseq})
$$

---

$$
\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1, m] \quad \Delta; \Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1, l] \quad \Delta; \Gamma \vdash e_k : \tau''_k \text{ for } k \in [1, n]}{\Delta \vdash \left[\begin{array}{l} \mathtt{node}\ f(x_1, \ldots, x_m)\ \mathtt{returns}\ (e_1, \ldots, e_n) \\ \mathtt{where}\ y_1 = e'_1; \ldots; y_l = e'_l \end{array}\right] : (\tau_1, \ldots, \tau_m) \to (\tau''_1, \ldots, \tau''_n)} \ (\textsc{Node})
$$

$$
\frac{\begin{array}{l} \Delta \vdash \mathsf{nd} : (\tau'_1, \ldots, \tau'_m) \to (\sigma'_1, \ldots, \sigma'_n) \\ \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, P_i \Rightarrow P'_i \text{ for } i \in [1, m] \qquad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, Q'_j \Rightarrow Q_j \text{ for } j \in [1, n] \\ (\text{where } \tau_i \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P_i\},\ \tau'_i \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid P'_i\},\ \sigma_j \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid Q_j\},\ \sigma'_j \equiv \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid Q'_j\}) \end{array}}{\Delta \vdash \mathsf{nd} : (\tau_1, \ldots, \tau_m) \to (\sigma_1, \ldots, \sigma_n)} \ (\textsc{NdConseq})
$$

---

$$
\frac{\Delta \vdash \mathsf{nd}_i : \Delta(f_i) \text{ for } i \in [1, m] \quad \Delta \vdash \mathsf{nd}_{\mathsf{Main}} : \nu}{\vdash [\mathsf{nd}_1, \ldots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}] : \nu} \ (\textsc{Prog}) \quad (f_i \text{ is the name of the nodes } \mathsf{nd}_i)
$$

$$
\frac{}{\Delta; \Gamma \vdash \mathtt{dt} : \prod_{v\in\mathbb{N}}\{u \in \mathbb{C} \mid u = \mathtt{dt}\}} \ (\mathrm{dt})
$$

$$
\frac{\begin{array}{l} \Delta; \Gamma \vdash e_1 : \prod_v \{u \mid P_1\} \qquad \Delta; \Gamma \vdash e_2 : \prod_v \{u \mid P_2\} \\ \models \forall v \in {}^*\mathbb{N}. \forall u \in {}^*\mathbb{C}. \big( (v < \frac{r}{\mathtt{dt}} \wedge P_1 \Rightarrow P) \wedge \\ \qquad (v \ge \frac{r}{\mathtt{dt}} \wedge P_2[(v - \lceil \frac{r}{\mathtt{dt}} \rceil)/v] \Rightarrow P) \big) \end{array}}{\Delta; \Gamma \vdash e_1 \ \mathtt{fby}^{\frac{r}{\mathtt{dt}}}\ e_2 : \prod_{v\in{}^*\mathbb{N}}\{u \in {}^*\mathbb{C} \mid P\}} \ (\textsc{Fby}^{\frac{r}{\mathtt{dt}}})
$$

Hasuo (Tokyo)

# SProc$^{\mathrm{dt}}$: Type System (For Partial Correctness)

$\mathbf{AExp} \ni a ::= v \mid c \mid a_1 + a_2 \mid a_1 \times a_2 \mid a_1 \mathbin{\hat{}} a_2 \mid \lceil a_1 \rceil \mid \mathtt{dt} \mid \frac{1}{\mathtt{dt}}$ where $v \in \mathbf{Var}$ and $c \in \mathbb{C}$

$\mathbf{Fml} \ni P ::= \mathtt{true} \mid \mathtt{false} \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \neg P \mid a_1 = a_2 \mid \mathtt{isReal}(a) \mid a_1 < a_2 \mid a_1 \leq a_2 \mid \forall v \in {}^*\mathbb{N}.\, P \mid \forall v \in {}^*\mathbb{C}.\, P$ where $v \in \mathbf{Var}$ and $a, a_i \in \mathbf{AExp}$

$\mathbf{SType}_{\mathbb{C}} \ni \tau ::= \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\}$ where $u, v \in \mathbf{Var}$, $P \in \mathbf{Fml}$ and $\mathbf{FV}(P) \subseteq \{u, v\}$

$\mathbf{SType}_{\mathbb{B}} \ni \beta ::= \prod_{v \in {}^*\mathbb{N}} P$ where $v \in \mathbf{Var}$, $P \in \mathbf{Fml}$ and $\mathbf{FV}(P)$ ⊆

$\mathbf{NdType}_{m,n} \ni \nu ::= (\tau_1, \dots, \tau_m) \to (\tau'_1, \dots,$ where $\tau_i, \tau'_i \in \mathbf{SType}_{\mathbb{C}}$

$$\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)}\ (\textsc{SVar})$$

$$\frac{\begin{array}{c}\Delta; \Gamma \vdash b : \prod_{v \in \mathbb{N}} P_b \\ \Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1, 2 \\ \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. (P_b \wedge P_1 \Rightarrow P) \wedge (P_b \wedge P_2 \Rightarrow P)\end{array}}{\Delta; \Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{If})$$

$$\frac{}{\Delta; \Gamma \vdash c : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid u = c\}}\ (\textsc{Const})$$

$$\frac{\Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1, 2 \quad \models \forall v \in \mathbb{N}. \forall u_1, u_2, u \in \mathbb{C}. (P_1 \wedge P_2 \wedge u = (u_1\ \mathsf{aop}\ u_2) \Rightarrow P)}{\Delta; \Gamma \vdash e_1\ \mathsf{aop}\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{AOp}) \quad (\mathsf{aop} \in \{+, \times, \hat{}\})$$

$$\frac{\Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1, 2 \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. \big((v < j \wedge P_1 \Rightarrow P) \wedge (v \geq j \wedge P_2[v - j/v] \Rightarrow P)\big)}{\Delta; \Gamma \vdash e_1\ \mathtt{fby}^j\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{Fby}^j)$$

$$\frac{\Delta; \Gamma \vdash e_i : \tau_i \text{ for } i \in [1, m] \quad \Delta(f) = (\tau_1, \dots, \tau_m) \to (\tau'_1, \dots, \tau'_n)}{\Delta; \Gamma \vdash \mathtt{proj}_k\ f(e_1, \dots, e_m) : \tau'_k}\ (\textsc{NdCall})$$

$\Delta; \Gamma \vdash e : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, P' \Rightarrow P$

$$\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1, m] \quad \Delta; \Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1, l] \quad \Delta; \Gamma \vdash e_k : \tau''_k \text{ for } k \in [1, n]}{\Delta \vdash \left[\begin{array}{l}\mathtt{node}\ f(x_1, \dots, x_m)\ \mathtt{returns}\ (e_1, \dots, e_n) \\ \mathtt{where}\ y_1 = e'_1; \dots; y_l = e'_l\end{array}\right] : (\tau_1, \dots, \tau_m) \to (\tau''_1, \dots, \tau''_n)}\ (\textsc{Node})$$

$$\frac{\begin{array}{l}\Delta \vdash \mathsf{nd} : (\tau'_1, \dots, \tau'_m) \to (\sigma'_1, \dots, \sigma'_n) \\ \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, P_i \Rightarrow P'_i \text{ for } i \in [1, m] \qquad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, Q'_j \Rightarrow Q_j \text{ for } j \in [1, n] \\ (\text{where } \tau_i \equiv \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\},\ \tau'_i \equiv \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P'_i\},\ \sigma_j \equiv \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid Q_j\},\ \sigma'_j \equiv \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid Q'_j\})\end{array}}{\Delta \vdash \mathsf{nd} : (\tau_1, \dots, \tau_m) \to (\sigma_1, \dots, \sigma_n)}\ (\textsc{NdConseq})$$

$$\frac{\Delta \vdash \mathsf{nd}_i : \Delta(f_i) \text{ for } i \in [1, m] \quad \Delta \vdash \mathsf{nd}_{\mathsf{Main}} : \nu}{\vdash [\mathsf{nd}_1, \dots, \mathsf{nd}_m; \mathsf{nd}_{\mathsf{Main}}] : \nu}\ (\textsc{Prog}) \quad (f_i \text{ is the name of the nodes } \mathsf{nd}_i)$$

$$\frac{}{\Delta; \Gamma \vdash \mathtt{dt} : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid u = \mathtt{dt}\}}\ (\mathrm{dt})$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash e_1 : \prod_v \{u \mid P_1\} \qquad \Delta; \Gamma \vdash e_2 : \prod_v \{u \mid P_2\} \\ \models \forall v \in {}^*\mathbb{N}. \forall u \in {}^*\mathbb{C}. \big((v < \frac{r}{\mathtt{dt}} \wedge P_1 \Rightarrow P) \wedge \\ \qquad (v \geq \frac{r}{\mathtt{dt}} \wedge P_2[(v - \lceil \frac{r}{\mathtt{dt}} \rceil)/v] \Rightarrow P)\big)\end{array}}{\Delta; \Gamma \vdash e_1\ \mathtt{fby}^{\frac{r}{\mathtt{dt}}}\ e_2 : \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\}}\ (\textsc{Fby}^{\frac{r}{\mathtt{dt}}})$$

Hasuo (Tokyo)

# SProc$^{\mathrm{dt}}$: Type System (For Partial Correctness)

$$
\begin{array}{lcl}
\mathbf{AExp} \ni a & ::= & v \mid c \mid a_1 + a_2 \mid a_1 \times a_2 \mid a_1 \mathbin{\char`\^} a_2 \mid \lceil a_1 \rceil \mid \\
& & \mathtt{dt} \mid \frac{1}{\mathtt{dt}} \quad \text{where } v \in \mathbf{Var} \text{ and } c \in \mathbb{C} \\
\mathbf{Fml} \ni P & ::= & \mathtt{true} \mid \mathtt{false} \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \neg P \mid \\
& & a_1 = a_2 \mid \mathtt{isReal}(a) \mid a_1 < a_2 \mid a_1 \le a_2 \mid \\
& & \forall v \in {}^*\mathbb{N}.\, P \mid \forall v \in {}^*\mathbb{C}.\, P \\
& & \quad \text{where } v \in \mathbf{Var} \text{ and } a, a_i \in \mathbf{AExp} \\
\mathbf{SType}_{\mathbb{C}} \ni \tau & ::= & \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\} \quad \text{where } u, v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \{u, v\} \\
\mathbf{SType}_{\mathbb{B}} \ni \beta & ::= & \prod_{v \in {}^*\mathbb{N}} P \quad \text{where } v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \ldots \\
\mathbf{NdType}_{m,n} \ni \nu & ::= & (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \\
& & \quad \text{where } \tau_i, \tau'_i \in \mathbf{SType}_{\mathbb{C}}
\end{array}
$$

$$\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)}\ (\textsc{SVar}) \qquad \frac{}{\Delta; \Gamma \vdash c : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid u = c\}}\ (\textsc{Const})$$

$$\frac{\Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u_1, u_2, u \in \mathbb{C}. (P_1 \wedge P_2 \wedge u = (u_1\ \mathtt{aop}\ u_2) \Rightarrow P)}{\Delta; \Gamma \vdash e_1\ \mathtt{aop}\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{AOp}) \quad (\mathtt{aop} \in \{+, \times, \mathbin{\char`\^}\})$$

$$\frac{\Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. ((v < j \wedge P_1 \Rightarrow P) \wedge (v \ge j \wedge P_2[v - j/v] \Rightarrow P))}{\Delta; \Gamma \vdash e_1\ \mathtt{fby}^j\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{Fby}^j)$$

$$\frac{\vdash b : \prod_{v \in \mathbb{N}} P_b \quad \Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. (P_b \wedge P_1 \Rightarrow P) \wedge (P_b \wedge P_2 \Rightarrow P)}{\Delta; \Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{If})$$

$$\frac{\Delta; \Gamma \vdash e_i : \tau_i \text{ for } i \in [1, m] \quad \Delta(f) = (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \tau'_n)}{\Delta; \Gamma \vdash \mathtt{proj}_k\ f(e_1, \ldots, e_m) : \tau'_k}\ (\textsc{NdCall})$$

$$\Delta; \Gamma \vdash e : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, P' \Rightarrow P$$

$$\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)}\ (\textsc{SVar})$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash b : \prod_{v \in \mathbb{N}} P_b \\ \Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1, 2 \\ \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. (P_b \wedge P_1 \Rightarrow P) \wedge (P_b \wedge P_2 \Rightarrow P)\end{array}}{\Delta; \Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{If})$$

$$\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1, m] \quad \Delta; \Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1, l] \quad \Delta; \Gamma \vdash e_k : \tau''_k \text{ for } k \in [1, n]}{\Delta \vdash \left[\begin{array}{l}\mathtt{node}\ f(x_1, \ldots, x_m)\ \mathtt{returns}\ (e_1, \ldots, e_n) \\ \mathtt{where}\ y_1 = e'_1; \ldots; y_l = e'_l\end{array}\right] : (\tau_1, \ldots, \tau_m) \to (\tau''_1, \ldots, \tau''_n)}\ (\textsc{Node})$$

$$\frac{}{\Delta; \Gamma \vdash \mathtt{dt} : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid u = \mathtt{dt}\}}\ (\mathtt{dt})$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash e_1 : \prod_v \{u \mid P_1\} \qquad \Delta; \Gamma \vdash e_2 : \prod_v \{u \mid P_2\} \\ \models \forall v \in {}^*\mathbb{N}. \forall u \in {}^*\mathbb{C}. ((v < \frac{r}{\mathtt{dt}} \wedge P_1 \Rightarrow P) \wedge \\ \qquad (v \ge \frac{r}{\mathtt{dt}} \wedge P_2[(v - \lceil \frac{r}{\mathtt{dt}} \rceil)/v] \Rightarrow P))\end{array}}{\Delta; \Gamma \vdash e_1\ \mathtt{fby}^{\frac{r}{\mathtt{dt}}}\ e_2 : \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\}}\ (\textsc{Fby}^{\frac{r}{\mathtt{dt}}})$$

Hasuo (Tokyo)

# SProc$^{\mathrm{dt}}$: Type System (For Partial Correctness)

$$\begin{array}{rcl}
\mathbf{AExp} \ni a & ::= & v \mid c \mid a_1 + a_2 \mid a_1 \times a_2 \mid a_1 \,\hat{}\, a_2 \mid \lceil a_1 \rceil \mid \\
& & \mathtt{dt} \mid \frac{1}{\mathtt{dt}} \quad \text{where } v \in \mathbf{Var} \text{ and } c \in \mathbb{C} \\
\mathbf{Fml} \ni P & ::= & \mathtt{true} \mid \mathtt{false} \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \neg P \mid \\
& & a_1 = a_2 \mid \mathtt{isReal}(a) \mid a_1 < a_2 \mid a_1 \le a_2 \mid \\
& & \forall v \in {}^*\mathbb{N}.\, P \mid \forall v \in {}^*\mathbb{C}.\, P \\
& & \quad \text{where } v \in \mathbf{Var} \text{ and } a, a_i \in \mathbf{AExp} \\
\mathbf{SType}_{\mathbb{C}} \ni \tau & ::= & \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\} \quad \text{where } u, v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \subseteq \{u, v\} \\
\mathbf{SType}_{\mathbb{B}} \ni \beta & ::= & \prod_{v \in {}^*\mathbb{N}} P \quad \text{where } v \in \mathbf{Var}, \\
& & \quad P \in \mathbf{Fml} \text{ and } \mathbf{FV}(P) \ldots \\
\mathbf{NdType}_{m,n} \ni \nu & ::= & (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \\
& & \quad \text{where } \tau_i, \tau'_i \in \mathbf{SType}_{\mathbb{C}}
\end{array}$$

$$\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)}\ (\textsc{SVar}) \qquad \frac{}{\Delta; \Gamma \vdash c : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid u = c\}}\ (\textsc{Const})$$

$$\frac{\Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u_i \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u_1, u_2, u \in \mathbb{C}. (P_1 \wedge P_2 \wedge u = (u_1 \,\mathtt{aop}\, u_2) \Rightarrow P)}{\Delta; \Gamma \vdash e_1 \,\mathtt{aop}\, e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{AOp}) \quad (\mathtt{aop} \in \{+, \times, \hat{}\})$$

$$\frac{\Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1,2 \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. ((v < j \wedge P_1 \Rightarrow P) \wedge (v \ge j \wedge P_2[v - j/v] \Rightarrow P))}{\Delta; \Gamma \vdash e_1 \,\mathtt{fby}^j\, e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{Fby}^j)$$

$$\frac{}{\Delta; \Gamma \vdash x : \Gamma(x)}\ (\textsc{SVar})$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash b : \prod_{v \in \mathbb{N}} P_b \\ \Delta; \Gamma \vdash e_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_i\} \text{ for } i = 1, 2 \\ \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. (P_b \wedge P_1 \Rightarrow P) \wedge (P_b \wedge P_2 \Rightarrow P)\end{array}}{\Delta; \Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P\}}\ (\textsc{If})$$

$$\frac{\Delta; \Gamma \vdash e_i : \tau_i \text{ for } i \in [1, m] \quad \Delta(f) = (\tau_1, \ldots, \tau_m) \to (\tau'_1, \ldots, \tau'_n)}{\Delta; \Gamma \vdash \mathtt{proj}_k\, f(e_1, \ldots, e_m) : \tau'_k}\ (\textsc{NdCall})$$

$$\Delta; \Gamma \vdash e : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P'\} \quad \models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}.\, P' \Rightarrow P$$

invariant! ➔ fixed pt. induction

$$\frac{\Gamma(x_i) \equiv \tau_i \text{ for } i \in [1, m] \quad \Delta; \Gamma \vdash e'_j : \Gamma(y_j) \text{ for } j \in [1, l] \quad \Delta; \Gamma \vdash e_k : \tau''_k \text{ for } k \in [1, n]}{\Delta \vdash \left[\begin{array}{l}\mathtt{node}\ f(x_1, \ldots, x_m)\ \mathtt{returns}\ (e_1, \ldots, e_n) \\ \mathtt{where}\ y_1 = e'_1; \ldots ; y_l = e'_l\end{array}\right] : (\tau_1, \ldots, \tau_m) \to (\tau''_1, \ldots, \tau''_n)}\ (\textsc{Node})$$

$$\frac{}{\Delta; \Gamma \vdash \mathtt{dt} : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid u = \mathtt{dt}\}}\ (\mathtt{dt})$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash e_1 : \prod_v \{u \mid P_1\} \qquad \Delta; \Gamma \vdash e_2 : \prod_v \{u \mid P_2\} \\ \models \forall v \in {}^*\mathbb{N}. \forall u \in {}^*\mathbb{C}. ((v < \frac{r}{\mathtt{dt}} \wedge P_1 \Rightarrow P) \wedge \\ \qquad (v \ge \frac{r}{\mathtt{dt}} \wedge P_2[(v - \lceil \frac{r}{\mathtt{dt}} \rceil)/v] \Rightarrow P))\end{array}}{\Delta; \Gamma \vdash e_1 \,\mathtt{fby}^{\frac{r}{\mathtt{dt}}}\, e_2 : \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid P\}}\ (\textsc{Fby}^{\frac{r}{\mathtt{dt}}})$$

Hasuo (Tokyo)

# $\mathbf{SProc}^{\mathrm{dt}}$: Type System Soundness

**Thm.** (Soundness)
A derivable type judgment $\boldsymbol{\Delta ; \Gamma \vdash e : \tau}$ is valid.

- Proof sketch:
  - Denotation $[\![ \boldsymbol{e} ]\!]$ :

    via **hyperdomains** (cpo's *-transferred via NSA)

    Also in [Beauxis & Mimram, CSL'11]
  - Soundness of $\mathbf{SProc}$ type system:

    by fixed point induction
  - It is *-transferred to $\mathbf{SProc}^{\mathrm{dt}}$

# FAQ on
# Nonstandard Static Analysis

Hasuo (Tokyo)

# FAQ on
# Nonstandard Static Analysis

* **Q.** Is an $\mathbf{SProc}^{\mathrm{dt}}/\mathbf{While}^{\mathrm{dt}}$ program executable?

# FAQ on
# Nonstandard Static Analysis

* **Q.** Is an $\mathbf{SProc}^{\mathrm{dt}}/\mathbf{While}^{\mathrm{dt}}$ program executable?
* **A.** Not exactly.

# FAQ on
# Nonstandard Static Analysis

* **Q.** Is an $\mathbf{SProc}^{\mathrm{dt}}/\mathbf{While}^{\mathrm{dt}}$ program executable?
* **A.** Not exactly.
    * **Modeling** languages
        * Not for numerical approximation, but for **exact** modeling

# FAQ on
# Nonstandard Static Analysis

* **Q.** Is an $\mathbf{SProc}^{\mathrm{dt}}/\mathbf{While}^{\mathrm{dt}}$ program executable?
* **A.** Not exactly.
  * **Modeling** languages
    * Not for numerical approximation, but for **exact** modeling
  * Static analysis ➜ **no need to execute!**
    * Mathematical semantics suffices

Hasuo (Tokyo)

# Un Momento!

* Done: framework for **hyperstreams**

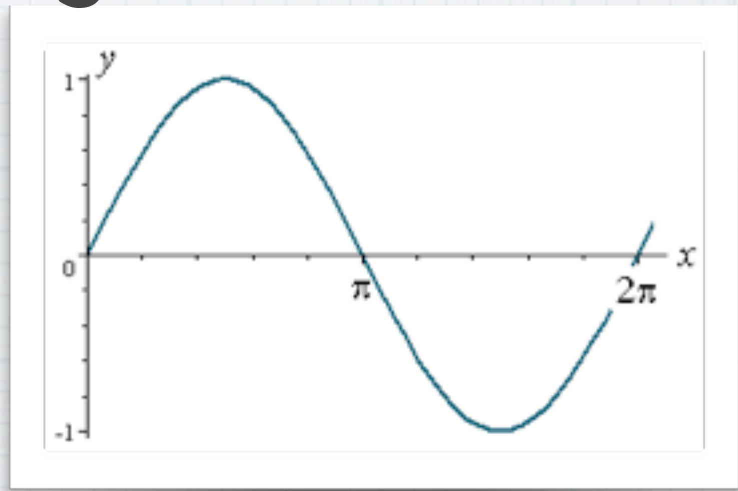
# Un Momento!

* Done: framework for **hyperstreams**
* hyperstream

  = streams with **infinitesimal sampling interval** dt

  = (conti.-time) signal

??

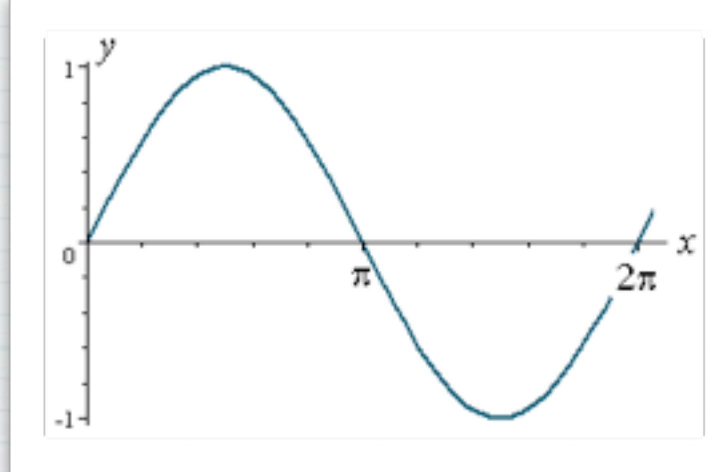


# Un Momento!

- Done: framework for **hyperstreams**
- hyperstream

  = streams with **infinitesimal sampling interval** dt

  ?? = (conti.-time) signal




- $$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \texttt{ fby } (s + c \times \mathrm{dt}); \\ & c = 1 \texttt{ fby } (c - s \times \mathrm{dt}) \end{array}\right] : \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid t_0 \leq v \times \mathrm{dt} \lor u \leq 1 + \varepsilon\} .$$

  - Does this really mean
    "the sine curve never exceeds 1" ??

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

* I. Stream Processing Language **SProc**
* II. Nonstandard Analysis
* III. **SProc**$^{dt}$ and Type System
* IV. Hyperstream Sampling

# Contribution

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

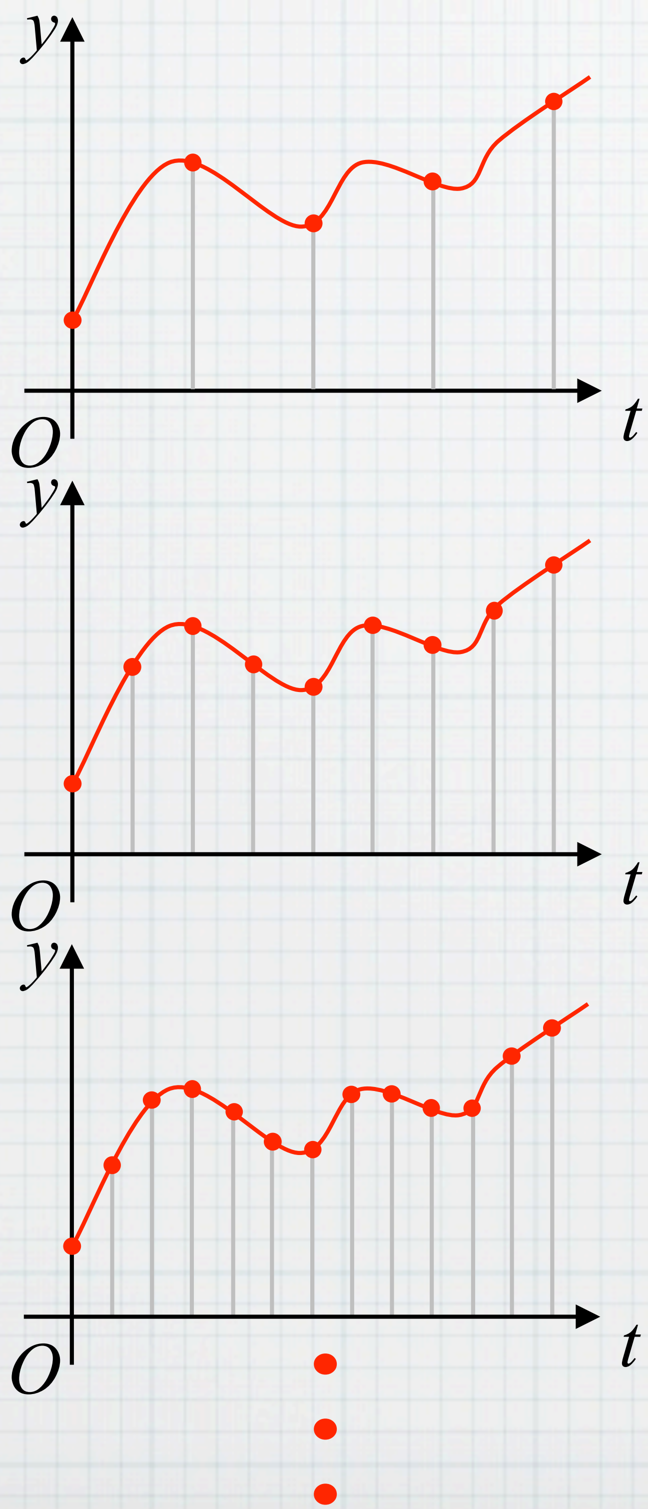
- I. Stream Processing Language **SProc**
- II. Nonstandard Analysis
- III. **SProc**$^{dt}$ and Type System
- IV. Hyperstream Sampling

# Part IV:

# Hyperstream Sampling

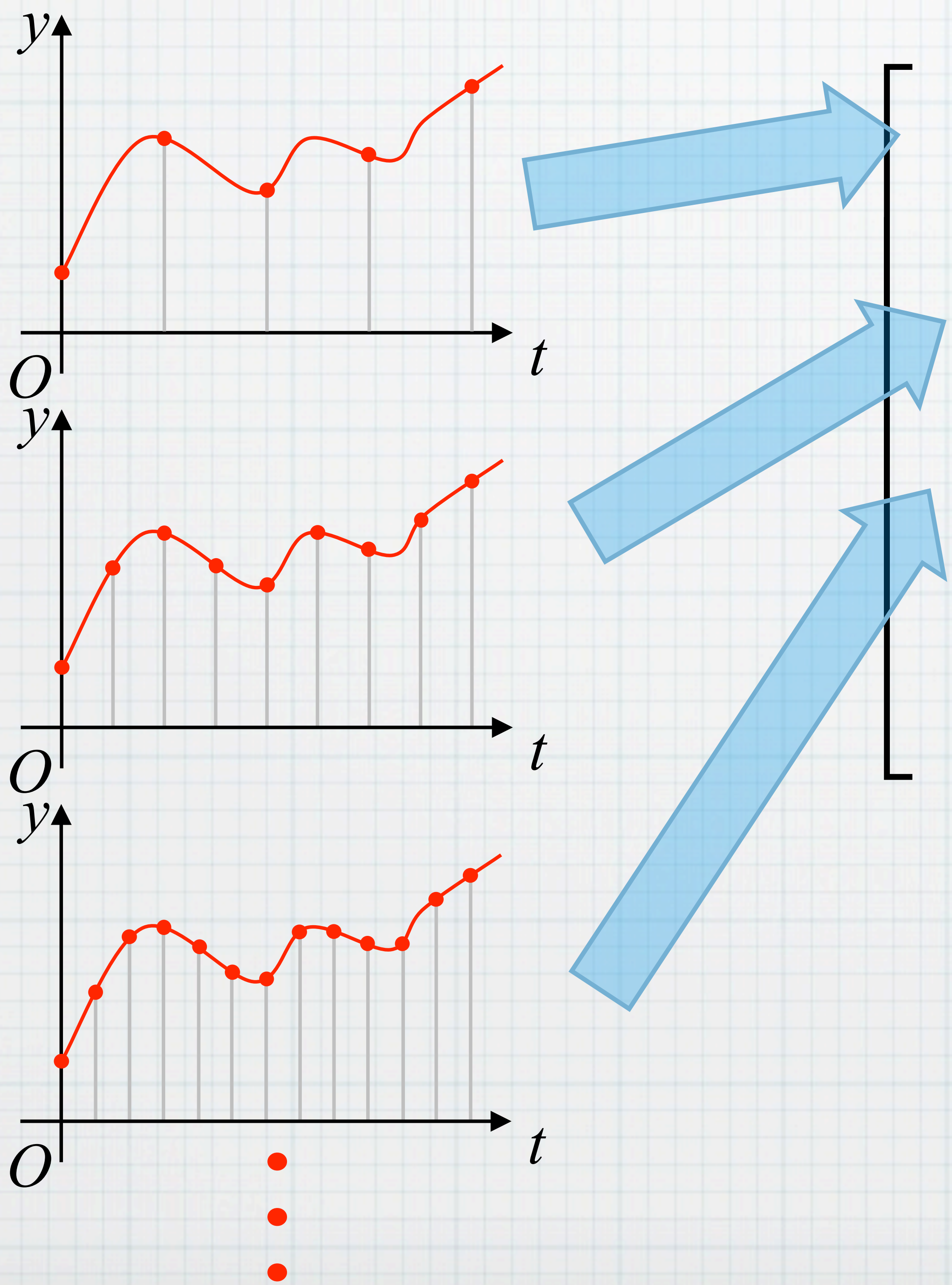
# Hyperstream Sampling

Also in: [Beauxis & Mimram, CSL'11]



# Hyperstream Sampling

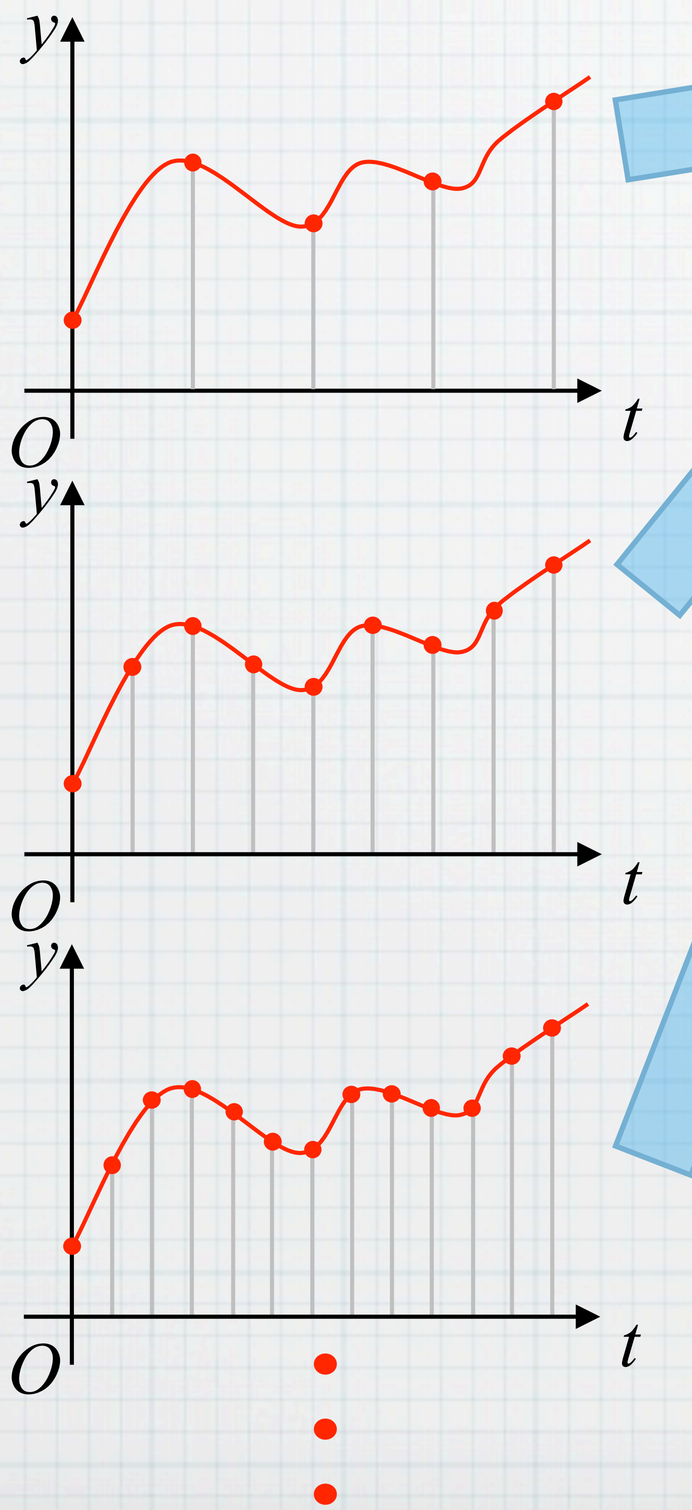
Also in: [Beauxis & Mimram, CSL'11]




$$\begin{bmatrix} \langle f(0), f(1), f(2), \ldots \rangle \\ \langle f(0), f(\frac{1}{2}), f(\frac{2}{2}), \ldots \rangle \\ \langle f(0), f(\frac{1}{3}), f(\frac{2}{3}), \ldots \rangle \\ \vdots \end{bmatrix}$$

# Hyperstream Sampling

### Also in: [Beauxis & Mimram, CSL'11]

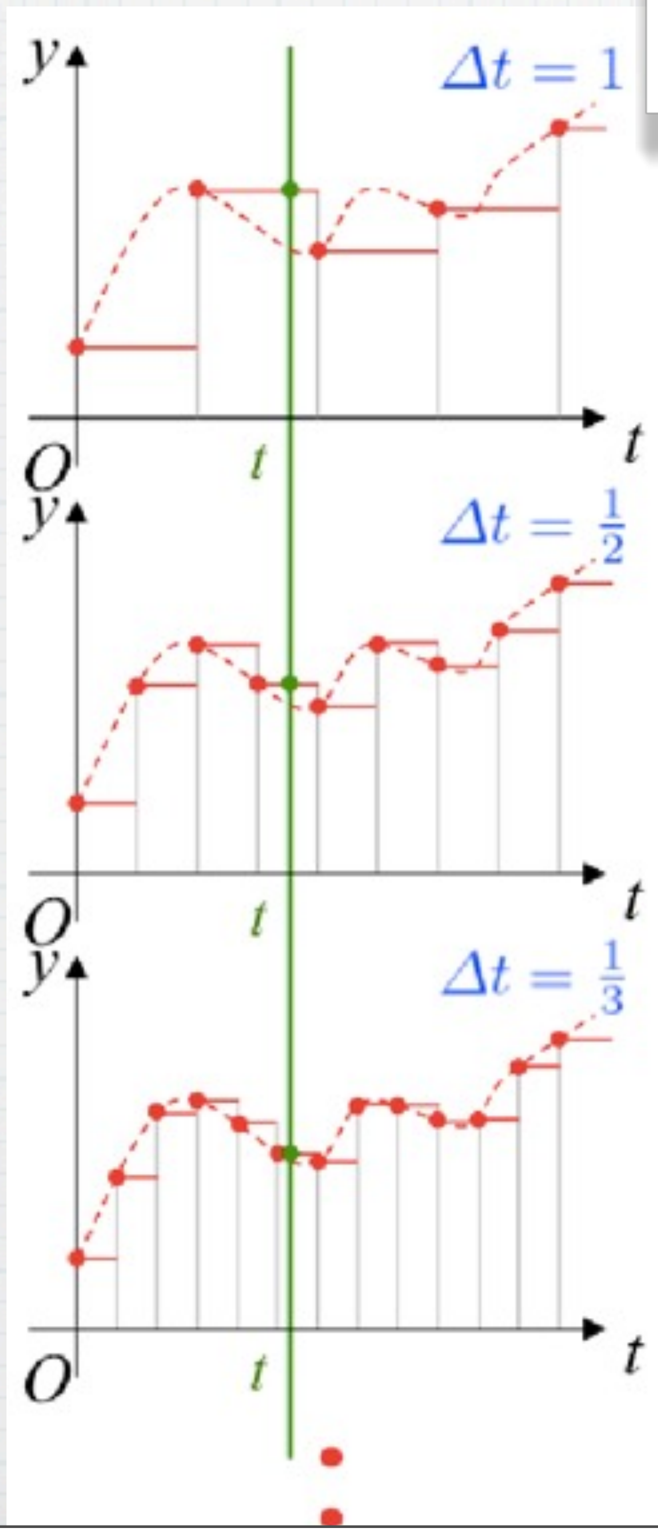



$$\begin{bmatrix} \langle f(0), f(1), f(2), \dots \rangle \\ \langle f(0), f(\frac{1}{2}), f(\frac{2}{2}), \dots \rangle \\ \langle f(0), f(\frac{1}{3}), f(\frac{2}{3}), \dots \rangle \\ \vdots \end{bmatrix}$$

- Sampling by **progressively smaller intervals**
- Hyperstream = "**stream of streams**"
  - Precise def.: via *-transfer (modulo an ultrafilter $\mathcal{F}$)

# "Smoothing"

$$\{\text{signals } f : \mathbb{R}_{\geq 0} \to \mathbb{C}\} \underset{\textsf{Smth}}{\overset{\textsf{Smp}}{\rightleftarrows}} \{\text{hyperstreams}\}$$




$$\langle f(0), f(1), f(2), \ldots \rangle$$

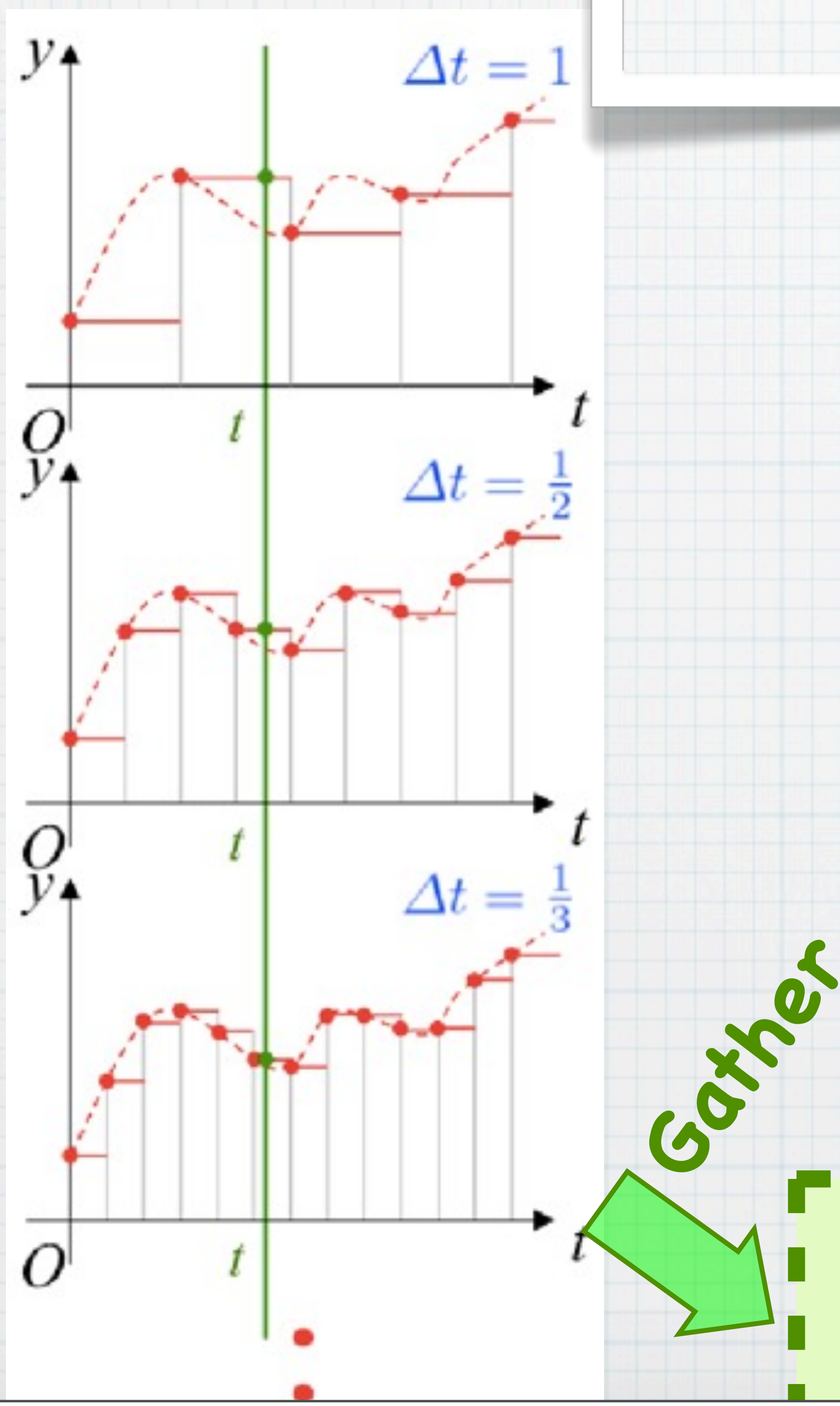
$$\langle f(\tfrac{0}{2}), f(\tfrac{1}{2}), f(\tfrac{2}{2}), \ldots \rangle$$

$$\langle f(\tfrac{0}{3}), f(\tfrac{1}{3}), f(\tfrac{2}{3}), \ldots \rangle$$

$$\vdots$$

# “Smoothing”

$$\{\text{signals } f : \mathbb{R}_{\geq 0} \to \mathbb{C}\} \underset{\textsf{Smth}}{\overset{\textsf{Smp}}{\rightleftarrows}} \{\text{hyperstreams}\}$$




$$\langle f(0), f(1), f(2), \ldots \rangle$$

$$\langle f(\tfrac{0}{2}), f(\tfrac{1}{2}), f(\tfrac{2}{2}), \ldots \rangle$$

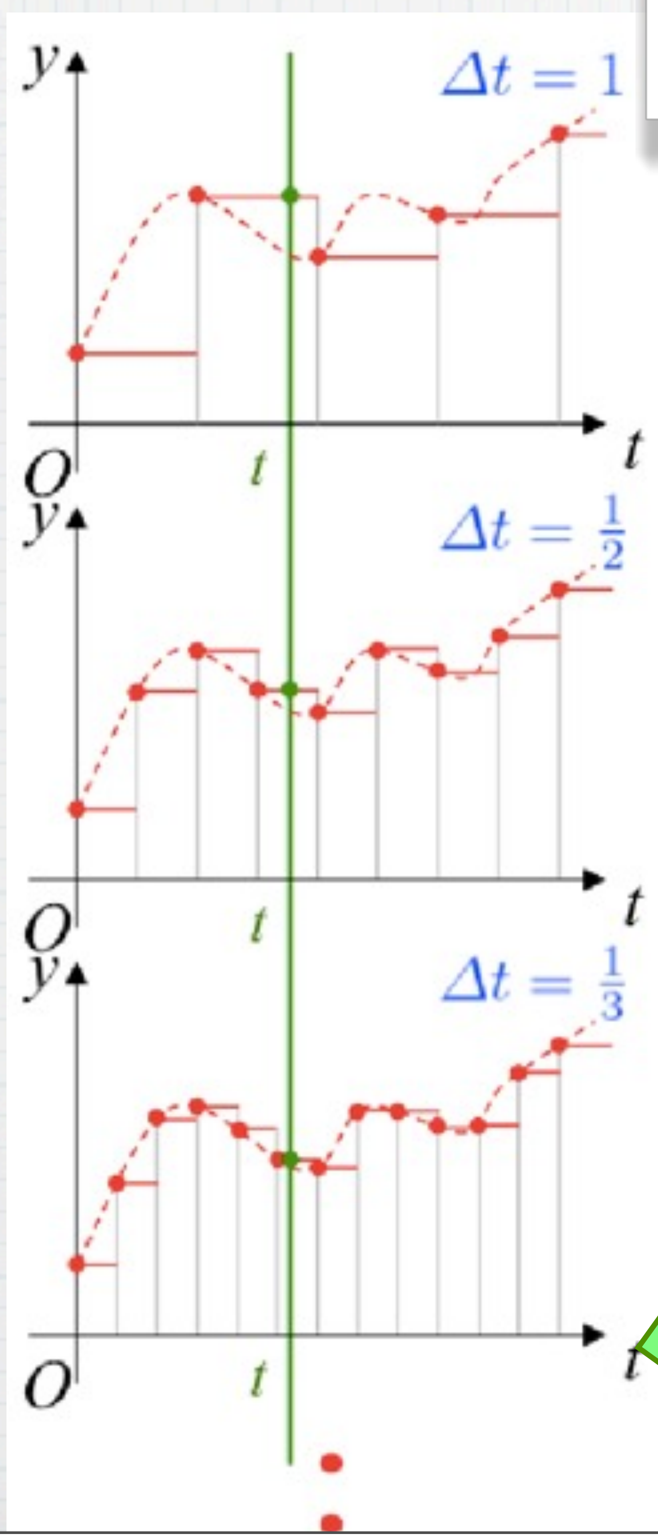
$$\langle f(\tfrac{0}{3}), f(\tfrac{1}{3}), f(\tfrac{2}{3}), \ldots \rangle$$

$$\vdots$$

$$\langle\ f(\lceil t \rceil),\ f(\tfrac{\lceil 2t \rceil}{2}),\ f(\tfrac{\lceil 3t \rceil}{3}),\ \ldots\ \rangle$$

# "Smoothing"

$$\{\text{signals } f : \mathbb{R}_{\geq 0} \to \mathbb{C}\} \underset{\textsf{Smth}}{\overset{\textsf{Smp}}{\rightleftarrows}} \{\text{hyperstreams}\}$$




$$\langle f(0), f(1), f(2), \ldots \rangle$$

$$\langle f(\tfrac{0}{2}), f(\tfrac{1}{2}), f(\tfrac{2}{2}), \ldots \rangle$$

$$\langle f(\tfrac{0}{3}), f(\tfrac{1}{3}), f(\tfrac{2}{3}), \ldots \rangle$$

$$\vdots$$

Gather

$$\langle\ f(\lceil t \rceil),\ f(\tfrac{\lceil 2t \rceil}{2}),\ f(\tfrac{\lceil 3t \rceil}{3}),\ \ldots\ \rangle$$

Limit??

$$f(t)$$

# Hyperstream Sampling

Also in: [Beauxis & Mimram, CSL'11]

* Our class of "signals"

**Def.** $f : \mathbb{R}_{\geq 0} \to \mathbb{C}$ is a *signal* if:

- it is of class $C\grave{a}dl\grave{a}g^{\infty}$; and,
- for any $t \in \mathbb{R}_{\geq 0}$, there is an interval $(t, t + \varepsilon)$ in which $f$ is of class $C^{\infty}$.

# Hyperstream Sampling

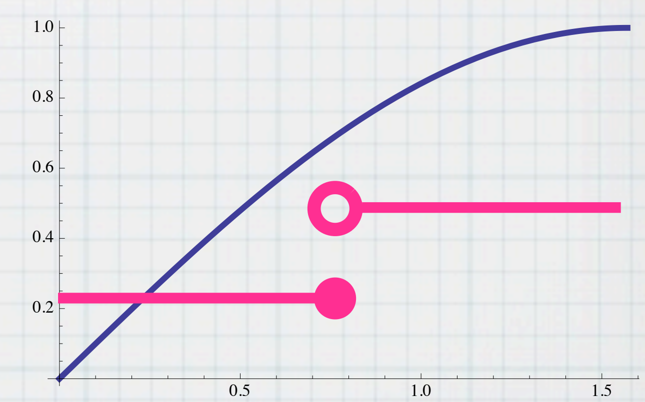
Also in: [Beauxis & Mimram, CSL'11]

* Our class of "signals"

**Def.** $f : \mathbb{R}_{\geq 0} \to \mathbb{C}$ is a *signal* if:

- it is of class $C\grave{a}dl\grave{a}g^{\infty}$; and,
- for any $t \in \mathbb{R}_{\geq 0}$, there is an interval $(t, t+\varepsilon)$ in which $f$ is of class $C^{\infty}$.

* **Càdlàg (Right-Continuous Left-Limit):**
  common notion in stochastic analysis
  (use suggested by M. L. Bujorianu)



# Hyperstream Sampling

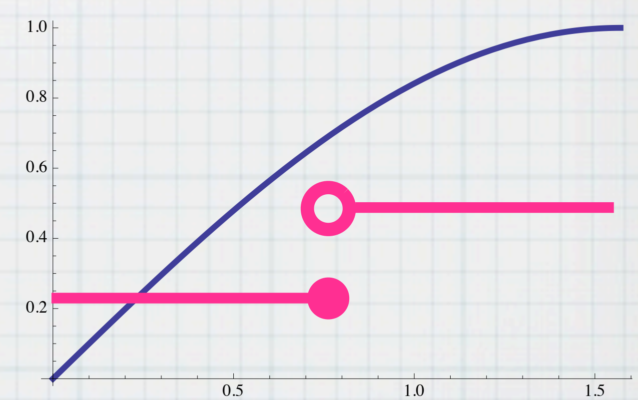
Also in: [Beauxis & Mimram, CSL'11]

* Our class of "signals"

**Def.** $f : \mathbb{R}_{\geq 0} \to \mathbb{C}$ is a *signal* if:

- it is of class $\mathit{Càdlàg}^{\infty}$; and,
- for any $t \in \mathbb{R}_{\geq 0}$, there is an interval $(t, t+\varepsilon)$ in which $f$ is of class $C^{\infty}$.

* **Càdlàg (Right-Continuous Left-Limit):**
common notion in stochastic analysis
(use suggested by M. L. Bujorianu)




* **Càdlàg$^{\infty}$: "Right-Differentiable Left-Limit"**
right derivative is again of class Càdlàg$^{\infty}$

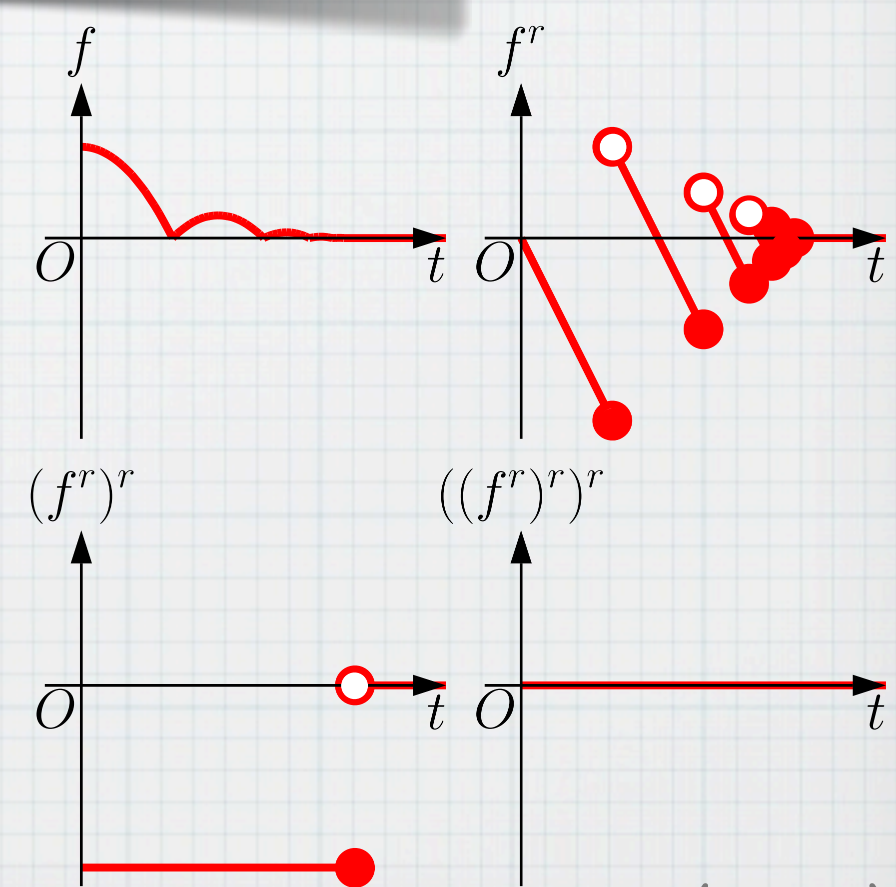
# Hyperstream Sampling

## Also in: [Beauxis & Mimram, CSL'11]

**Def.** $f : \mathbb{R}_{\geq 0} \to \mathbb{C}$ is a *signal* if:

- it is of class *Càdlàg*$^\infty$; and,
- for any $t \in \mathbb{R}_{\geq 0}$, there is an interval $(t, t+\varepsilon)$ in which $f$ is of class $C^\infty$.

* May not be "**the right**" class, but **reasonable**
  * Much more than class $C^\infty$ cf. [Beauxis & Mimram, CSL'11]
  * Accommodates many hybrid dynamics (e.g. a bouncing ball)
  * Closed under **right differentiation** and **Riemann Integration**



# Hyperstream Sampling

## Also in: [Beauxis & Mimram, CSL'11]

$$\{\text{signals } f : \mathbb{R}_{\geq 0} \to \mathbb{C}\} \underset{\textbf{Smth}}{\overset{\textbf{Smp}}{\rightleftarrows}} \{\text{hyperstreams}\}$$

**Thm.** For each signal $f$ and $t \in \mathbb{R}_{\geq 0}$:

- the hyperreal $\mathbf{Smth}(\mathbf{Smp}(f))(t)$ is limited; and
- $\mathrm{sh}\big(\mathbf{Smth}(\mathbf{Smp}(f))(t)\big) = f(t)$.

# Hyperstream Sampling

Also in: [Beauxis & Mimram, CSL'11]

$$\{\text{signals } f : \mathbb{R}_{\geq 0} \to \mathbb{C}\} \underset{\textbf{Smth}}{\overset{\textbf{Smp}}{\rightleftarrows}} \{\text{hyperstreams}\}$$

**Thm.** For each signal $f$ and $t \in \mathbb{R}_{\geq 0}$:

- the hyperreal $\mathbf{Smth}(\mathbf{Smp}(f))(t)$ is limited; and
- $\mathrm{sh}\big(\mathbf{Smth}(\mathbf{Smp}(f))(t)\big) = f(t)$.

**sh**: "shadow"
(an NSA way of taking "limit")

# Hyperstream Sampling

Also in: [Beauxis & Mimram, CSL'11]

$$\{\text{signals } f : \mathbb{R}_{\geq 0} \to \mathbb{C}\} \underset{\textbf{Smth}}{\overset{\textbf{Smp}}{\rightleftarrows}} \{\text{hyperstreams}\}$$

**Thm.** For each signal $f$ and $t \in \mathbb{R}_{\geq 0}$:

- the hyperreal $\mathbf{Smth}(\mathbf{Smp}(f))(t)$ is limited; and
- $\mathrm{sh}\big(\mathbf{Smth}(\mathbf{Smp}(f))(t)\big) = f(t)$.

sh: "shadow"
(an NSA way of taking "limit")

* Because of this,

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \;\texttt{fby}\; (s + c \times \mathrm{dt}); \\ & c = 1 \;\texttt{fby}\; (c - s \times \mathrm{dt}) \end{array}\right] : \prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid t_0 \leq v \times \mathrm{dt} \vee u \leq 1 + \varepsilon\} \,.$$

means "the sine curve never exceeds 1"

# Related Work

* Stream processing + NSA
  [Benveniste, Bourke, Caillaud, & Pouzet, LCTES'11; J. Comp. Sys. Sci. '12]
  * Not about deductive verification
* Formal verification in Simulink
  [Bouissou & Chapoutot, LCTES'12] [Chapoutot & Martel, ICESS'09] [Schrammel & Jeannet, HSCC'12] [Tripakis, Sofronis, Caspi, & Curic, ACM Trans. Emb. Comp. Sys.'05]
* Formal verification in stream prosessing
  [Gamati´e and Gonnord, LCTES'11]
* Formal verification of hybrid systems
  [Lee & Zheng, HSCC'05] [Sankaranarayanan, HSCC'10]
  * Model checking [Alur et al., TCS'95]
  * Deductive/theorem proving [Platzer'10] [Platzer, LICS'12]
* NSA for hybrid systems [Bliudze & Krob, Fund. Inform.'09]
* [Beauxis & Mimram, CSL'11]
  * The basic ideas of "hyperdomains" and "hyperstream sampling" are already here

# Conclusions:

# Nonstandard Static Analysis

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$<br>Imperative | **SProc**$^{dt}$<br>hyperstream processing language<br>1st-order functional<br>(like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system**<br>(partial correctness) |

- Future work:
  - Extended examples; a more expressive logic
  - Invariant discovery
  - Verification of Simulink
  - "Hyper Fourier transform"?

# Conclusions:

# Nonstandard Static Analysis

| | [ICALP'11] [CAV'12] | [POPL'13] |
|---|---|---|
| Programing language | **While**$^{dt}$ Imperative | **SProc**$^{dt}$ hyperstream processing language 1st-order functional (like **Lustre**) |
| Program logic | **Hoare**$^{dt}$ | **Type system** (partial correctness) |

* Future work:
  * Extended examples; a more expressive logic
  * Invariant discovery
  * Verification of Simulink
  * "Hyper Fourier transform"?

**Thank you for your attention!**

Ichiro Hasuo (Dept. CS, U Tokyo)
http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro/

# Set Theory in Nonstandard Analysis

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

**Thm. (Transfer Principle)**
$\boldsymbol{A}$: a first-order formula in $\mathscr{L}_{\boldsymbol{X}}$.
$^*\boldsymbol{A}$: its $*$-*transform*. Then

$$\mathbb{R} \models \boldsymbol{A} \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*\boldsymbol{A} \;.$$

# Set Theory in Nonstandard Analysis

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathrm{N}} / \sim_{\mathcal{F}}$$

**Thm. (Transfer Principle)**
$\boldsymbol{A}$: a first-order formula in $\mathscr{L}_{\mathbf{X}}$.
$^*\boldsymbol{A}$: its $*$-*transform*. Then

$$\mathbb{R} \models \boldsymbol{A} \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*\boldsymbol{A}\;.$$

# Set Theory in Nonstandard Analysis

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

**Thm. (Transfer Principle)**
$\boldsymbol{A}$: a first-order formula in $\mathscr{L}_{\mathbf{X}}$.
$^*\boldsymbol{A}$: its $*$-*transform*. Then

$$\mathbb{R} \models \boldsymbol{A} \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*\boldsymbol{A}\ .$$

* $\mathscr{L}_X$ : (almost) the lang. of set theory

  * <u>**Terms**</u>: variables and constants $a \in \begin{bmatrix} \boldsymbol{X} \\ \cup\, \mathcal{P}(\boldsymbol{X}) \\ \cup\, \mathcal{P}(\boldsymbol{X} \cup \mathcal{P}(\boldsymbol{X})) \\ \cup\, \cdots \end{bmatrix}$
  * <u>**Predicates**</u>: $=$ and $\in$
  * <u>**Connectives**</u>: $\wedge$, $\vee$, $\neg$, $\Rightarrow$
  * <u>**Quantifiers**</u>: always bounded $\quad \forall x \in s \quad$ ($s$ is a term)

# Set Theory in Nonstandard Analysis

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

**Thm. (Transfer Principle)**
$\boldsymbol{A}$: a first-order formula in $\mathscr{L}_{\mathbf{X}}$.
$^*\boldsymbol{A}$: its $*$-*transform*. Then

$$\mathbb{R} \models \boldsymbol{A} \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*\boldsymbol{A} \;.$$

# Set Theory in Nonstandard Analysis

**Defn.**
The set of *hyperreal numbers* is

$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

**Thm. (Transfer Principle)**
$\boldsymbol{A}$: a first-order formula in $\mathscr{L}_{\mathbf{X}}$.
${}^*\boldsymbol{A}$: its $*$-*transform*. Then

$$ \mathbb{R} \models \boldsymbol{A} \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*\boldsymbol{A} \;. $$

- $\mathscr{L}_X$ : (almost) the lang. of set theory
  - Rich enough to do "usual mathematics"

# Set Theory in Nonstandard Analysis

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

**Thm. (Transfer Principle)**
$\boldsymbol{A}$: a first-order formula in $\mathscr{L}_{\mathbf{X}}$.
$^*\boldsymbol{A}$: its $*$-*transform.* Then

$$\mathbb{R} \models \boldsymbol{A} \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*\boldsymbol{A} \;.$$

- $\mathscr{L}_X$ : (almost) the lang. of set theory
  - Rich enough to do "usual mathematics"
    - E.g.: "is a cpo"; "is a continuous function"

$$\begin{aligned}
&\mathsf{CPO}_{a,r} :\equiv \; \mathsf{Poset}_{a,r} \wedge \mathsf{HasBot}_{a,r} \wedge \\
&\quad \forall s \in (\mathbb{N} \to a).\,\big(\, \mathsf{AscCn}_{a,r}(s) \Rightarrow \exists p \in a.\, \mathsf{Sup}_{a,r}(p, s)\,\big) \;, \\
&\mathsf{Conti}_{a_1,r_1,a_2,r_2}(f) :\equiv \; \forall s \in (\mathbb{N} \to a_1).\, \forall p \in a_1. \\
&\quad \big(\, \mathsf{AscCn}_{a_1,r_1}(s) \wedge \mathsf{Sup}_{a_1,r_1}(p, s) \Rightarrow \mathsf{Sup}_{a_2,r_2}(f(p), f \circ s)\,\big) \;.
\end{aligned}$$

$\mathsf{BinRel}_{a,r} :\equiv\ r \subseteq a \times a \qquad \mathsf{Refl}_{a,r} :\equiv\ \forall x \in a.\,(x,x) \in r$
$\mathsf{Trans}_{a,r} :\equiv\ \forall x,y,z \in a.\,(\,(x,y) \in r \wedge (y,z) \in r \Rightarrow (x,z) \in r\,)$
$\mathsf{AntiSym}_{a,r} :\equiv\ \forall x,y \in a.\,(\,(x,y) \in r \wedge (y,x) \in r \Rightarrow x = y\,)$
$\mathsf{Poset}_{a,r} :\equiv\ \mathsf{BinRel}_{a,r} \wedge \mathsf{Refl}_{a,r} \wedge \mathsf{Trans}_{a,r} \wedge \mathsf{AntiSym}_{a,r}$
$\mathsf{HasBot}_{a,r} :\equiv\ \exists x \in a.\,\forall y \in a.\,(x,y) \in r$
$\mathsf{AscCn}_{a,r}(s) :\equiv\ \forall x,x' \in \mathbb{N}.\,(x \le x' \Rightarrow (s(x),s(x')) \in r)$
$\mathsf{UpBd}_{a,r}(b,s) :\equiv\ \forall x \in \mathbb{N}.\,((s(x),b) \in r)$
$\mathsf{Sup}_{a,r}(p,s) :\equiv\ \mathsf{UpBd}_{a,r}(p,s) \wedge \forall b \in a.\,(\mathsf{UpBd}_{a,r}(b,s) \Rightarrow (p,b) \in r)$

# ...ry in Analysis

**Defn.**
The set of *hyperreal numbers* is

$$^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}}$$

**Thm. (Transfer Principle)**
$A$: a first-order formula in $\mathscr{L}_X$.
$^*A$: its $*$-*transform*. Then

$$\mathbb{R} \models A \quad\Longleftrightarrow\quad {}^*\mathbb{R} \models {}^*A\ .$$

- $\mathscr{L}_X$: (almost) the lang. of set theory
  - Rich enough to do "usual mathematics"
    - E.g.: "is a cpo", "is a continuous function"

$$\begin{aligned}
&\mathsf{CPO}_{a,r} :\equiv\ \mathsf{Poset}_{a,r} \wedge \mathsf{HasBot}_{a,r} \wedge \\
&\quad \forall s \in (\mathbb{N} \to a).\,(\,\mathsf{AscCn}_{a,r}(s) \Rightarrow \exists p \in a.\,\mathsf{Sup}_{a,r}(p,s)\,)\ , \\
&\mathsf{Conti}_{a_1,r_1,a_2,r_2}(f) :\equiv\ \forall s \in (\mathbb{N} \to a_1).\,\forall p \in a_1. \\
&\quad (\,\mathsf{AscCn}_{a_1,r_1}(s) \wedge \mathsf{Sup}_{a_1,r_1}(p,s) \Rightarrow \mathsf{Sup}_{a_2,r_2}(f(p), f \circ s)\,)\ .
\end{aligned}$$

# $\mathbf{SProc}^{\mathrm{dt}}$: Type System Example

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = \mathbf{0}\ \texttt{fby}\ (s + c \times \texttt{dt}); \\ & c = \mathbf{1}\ \texttt{fby}\ (c - s \times \texttt{dt}) \end{array}\right] \quad :$$

$$\prod_{v \in {}^*\mathbb{N}} \{u \in {}^*\mathbb{C} \mid t_0 \leq v \times \texttt{dt} \vee u \leq 1 + \varepsilon\} \ .$$

# SProc$^{\mathrm{dt}}$: Type System Example

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = \mathbf{0}\ \texttt{fby}\ (s + c \times \texttt{dt}); \\ & c = \mathbf{1}\ \texttt{fby}\ (c - s \times \texttt{dt}) \end{array}\right] \quad : \quad \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid t_0 \leq v \times \texttt{dt} \vee u \leq 1 + \varepsilon \} \ .$$

In derivation, the key step is:

$$\dfrac{\begin{array}{l} \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash \mathbf{0}\ \texttt{fby}^1\ (s + c \times \texttt{dt}) : \tau_{s\text{-inv}} \\ \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash \mathbf{1}\ \texttt{fby}^1\ (c - s \times \texttt{dt}) : \tau_{c\text{-inv}} \\ \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash s : \tau_{s\text{-inv}} \end{array}}{\Delta_0; \left\{\begin{array}{l} s : \tau_{s\text{-inv}} \\ c : \tau_{c\text{-inv}} \end{array}\right\} \vdash \left[\begin{array}{l} \texttt{node Sine() returns } (s) \\ \texttt{where } s = \mathbf{0}\ \texttt{fby}^1\ (s + c \times \texttt{dt}); \\ \qquad\quad c = \mathbf{1}\ \texttt{fby}^1\ (c - s \times \texttt{dt}) \end{array}\right] : \ \tau_{s\text{-inv}}} \ (\textsc{Node})$$

# SProc$^{dt}$: Type System Example

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = \mathbf{0}\ \texttt{fby}\ (s + c \times \mathtt{dt}); \\ & c = \mathbf{1}\ \texttt{fby}\ (c - s \times \mathtt{dt}) \end{array}\right] \quad : \quad \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid t_0 \leq v \times \mathtt{dt} \vee u \leq 1 + \varepsilon \} \ .$$

fixed pt. induction

In derivation, the key step is:

$$\frac{\begin{array}{l} \Delta_0; \{ s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}} \} \vdash \mathbf{0}\ \texttt{fby}^1\ (s + c \times \mathtt{dt}) : \tau_{s\text{-inv}} \\ \Delta_0; \{ s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}} \} \vdash \mathbf{1}\ \texttt{fby}^1\ (c - s \times \mathtt{dt}) : \tau_{c\text{-inv}} \\ \Delta_0; \{ s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}} \} \vdash s : \tau_{s\text{-inv}} \end{array}}{\Delta_0; \left\{\begin{array}{l} s : \tau_{s\text{-inv}} \\ c : \tau_{c\text{-inv}} \end{array}\right\} \vdash \left[\begin{array}{l} \texttt{node Sine() returns } (s) \\ \texttt{where } s = \mathbf{0}\ \texttt{fby}^1\ (s + c \times \mathtt{dt}); \\ \qquad\quad c = \mathbf{1}\ \texttt{fby}^1\ (c - s \times \mathtt{dt}) \end{array}\right] \ : \ \tau_{s\text{-inv}}} \ (\textsc{Node})$$

# $\mathbf{SProc}^{\mathrm{dt}}$: Type System Example

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \texttt{ fby } (s + c \times \mathrm{dt}); \\ & c = 1 \texttt{ fby } (c - s \times \mathrm{dt}) \end{array}\right] \quad : \quad \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid t_0 \leq v \times \mathrm{dt} \vee u \leq 1 + \varepsilon \} \ .$$

In derivation, the key step is:

$$\dfrac{\begin{array}{c} \Delta_0; \{ s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}} \} \vdash 0 \texttt{ fby}^1 \ (s + c \times \mathrm{dt}) : \tau_{s\text{-inv}} \\ \Delta_0; \{ s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}} \} \vdash 1 \texttt{ fby}^1 \ (c - s \times \mathrm{dt}) : \tau_{c\text{-inv}} \\ \Delta_0; \{ s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}} \} \vdash s : \tau_{s\text{-inv}} \end{array}}{\Delta_0; \left\{\begin{array}{l} s : \tau_{s\text{-inv}} \\ c : \tau_{c\text{-inv}} \end{array}\right\} \vdash \left[\begin{array}{l} \texttt{node Sine() returns } (s) \\ \texttt{where } s = 0 \texttt{ fby}^1 \ (s + c \times \mathrm{dt}); \\ \qquad c = 1 \texttt{ fby}^1 \ (c - s \times \mathrm{dt}) \end{array}\right] : \ \tau_{s\text{-inv}}} \ (\textsc{Node})$$

fixed pt. induction

Hasuo (Tokyo)

# SProc$^{\mathrm{dt}}$: Type System Example

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \texttt{ fby } (s + c \times \mathrm{dt}); \\ & c = 1 \texttt{ fby } (c - s \times \mathrm{dt}) \end{array}\right] \quad : \quad \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid t_0 \le v \times \mathrm{dt} \vee u \le 1 + \varepsilon \} \ .$$

fixed pt. induction

In derivation, the key step is:

$$\dfrac{\begin{array}{l} \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash 0 \texttt{ fby}^1 \ (s + c \times \mathrm{dt}) : \tau_{s\text{-inv}} \\ \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash 1 \texttt{ fby}^1 \ (c - s \times \mathrm{dt}) : \tau_{c\text{-inv}} \\ \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash s : \tau_{s\text{-inv}} \end{array}}{\Delta_0; \left\{\begin{array}{l} s : \tau_{s\text{-inv}} \\ c : \tau_{c\text{-inv}} \end{array}\right\} \vdash \left[\begin{array}{l} \texttt{node Sine() returns } (s) \\ \texttt{where } s = 0 \texttt{ fby}^1 \ (s + c \times \mathrm{dt}); \\ \qquad c = 1 \texttt{ fby}^1 \ (c - s \times \mathrm{dt}) \end{array}\right] \ : \ \tau_{s\text{-inv}}} \ (\textsc{Node})$$

Where the "invariants" are

$$\tau_{s\text{-inv}} \equiv \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid u = \tfrac{1}{2} \mathrm{i}(1 - \mathrm{i} \cdot \mathrm{dt})^v - \tfrac{1}{2} \mathrm{i}(1 + \mathrm{i} \cdot \mathrm{dt})^v \}$$

$$\tau_{c\text{-inv}} \equiv \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid u = \tfrac{1}{2}(1 - \mathrm{i} \cdot \mathrm{dt})^v + \tfrac{1}{2}(1 + \mathrm{i} \cdot \mathrm{dt})^v \}$$

# $\mathbf{SProc}^{\mathrm{dt}}$: Type System Example

$$\vdash \left[\begin{array}{ll} \texttt{node} & \texttt{Sine() returns } (s) \\ \texttt{where} & s = 0 \texttt{ fby } (s + c \times \mathrm{dt}); \\ & c = 1 \texttt{ fby } (c - s \times \mathrm{dt}) \end{array}\right] \; : \; \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid t_0 \leq v \times \mathrm{dt} \vee u \leq 1 + \varepsilon \} \;.$$

fixed pt. induction

In derivation, the key step is:

$$\dfrac{\begin{array}{l} \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash 0 \texttt{ fby}^1 \; (s + c \times \mathrm{dt}) : \tau_{s\text{-inv}} \\ \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash 1 \texttt{ fby}^1 \; (c - s \times \mathrm{dt}) : \tau_{c\text{-inv}} \\ \Delta_0; \{s : \tau_{s\text{-inv}}, c : \tau_{c\text{-inv}}\} \vdash s : \tau_{s\text{-inv}} \end{array}}{\Delta_0; \left\{\begin{array}{l} s : \tau_{s\text{-inv}} \\ c : \tau_{c\text{-inv}} \end{array}\right\} \vdash \left[\begin{array}{l} \texttt{node Sine() returns } (s) \\ \texttt{where } s = 0 \texttt{ fby}^1 \; (s + c \times \mathrm{dt}); \\ \qquad\; c = 1 \texttt{ fby}^1 \; (c - s \times \mathrm{dt}) \end{array}\right] : \tau_{s\text{-inv}}} \; (\textsc{Node})$$

Where the "invariants" are

$$\tau_{s\text{-inv}} \equiv \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid u = \tfrac{1}{2}\mathrm{i}(1 - \mathrm{i} \cdot \mathrm{dt})^v - \tfrac{1}{2}\mathrm{i}(1 + \mathrm{i} \cdot \mathrm{dt})^v \}$$

$$\tau_{c\text{-inv}} \equiv \prod_{v \in {}^*\mathbb{N}} \{ u \in {}^*\mathbb{C} \mid u = \tfrac{1}{2}(1 - \mathrm{i} \cdot \mathrm{dt})^v + \tfrac{1}{2}(1 + \mathrm{i} \cdot \mathrm{dt})^v \}$$

By solving recurrence relations

Tokyo)