Exercises in Nonstandard Static Analysis of Hybrid Systems

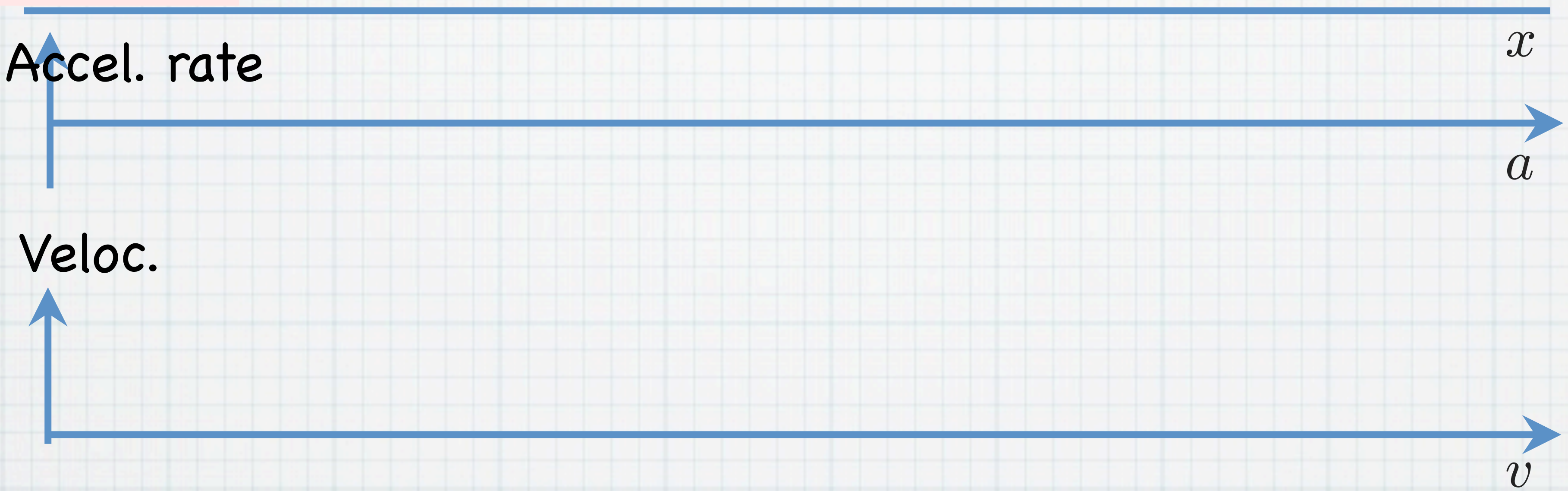
Ichiro Hasuo
University of Tokyo (JP)

Kohei Suenaga
Kyoto University (JP)

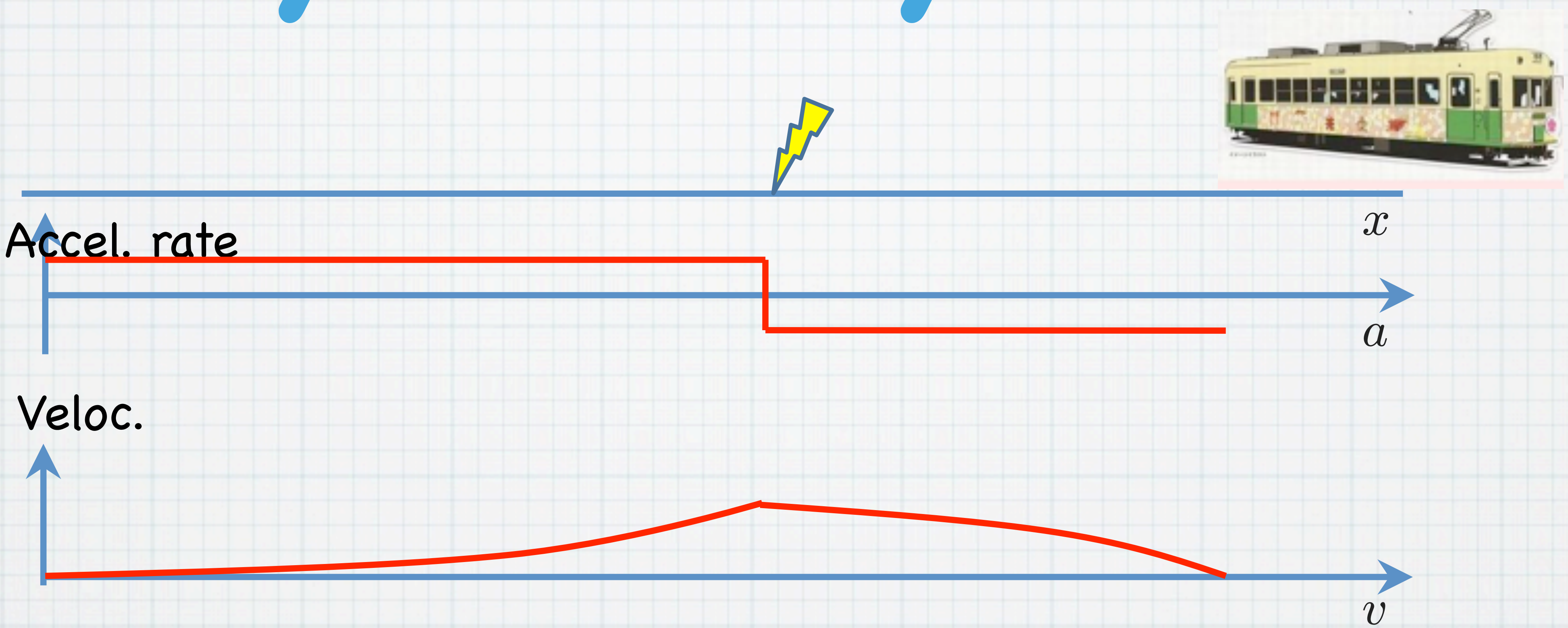
東京大学 THE UNIVERSITY OF TOKYO

京都大学 KYOTO UNIVERSITY

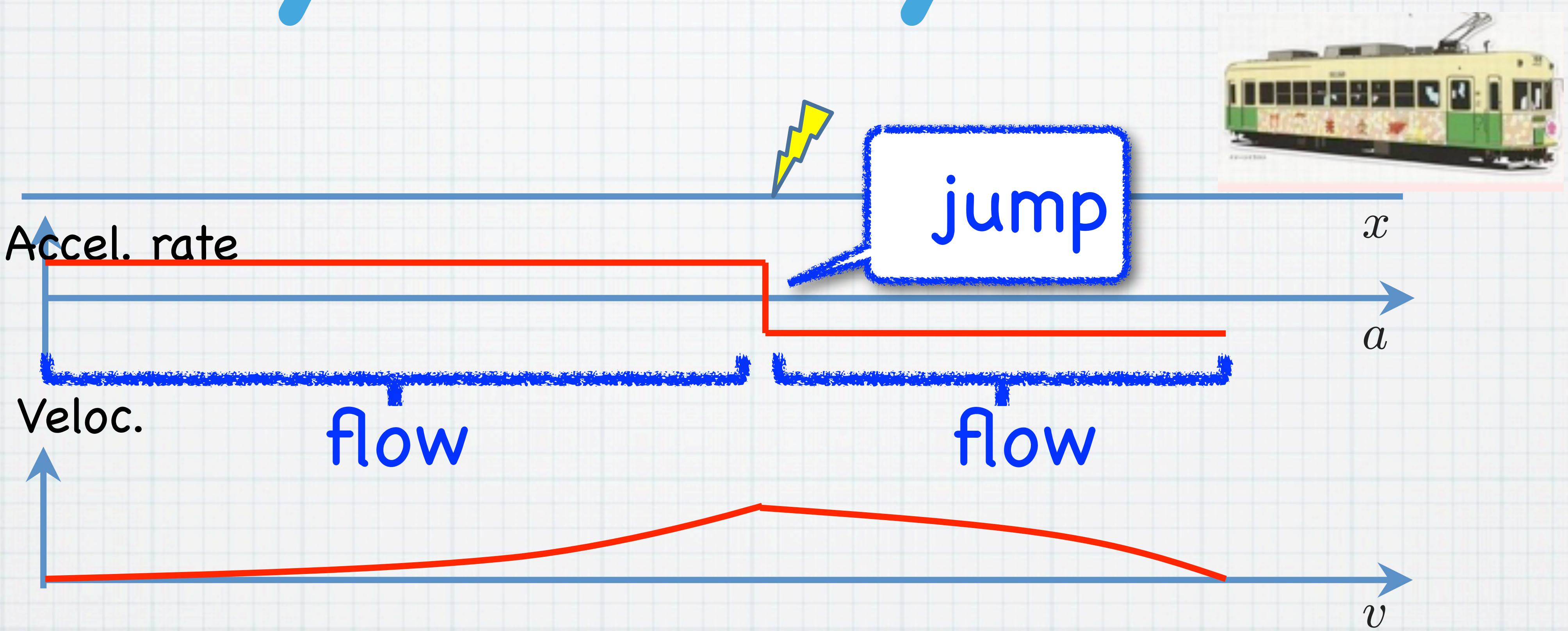
Hybrid System



Hybrid System

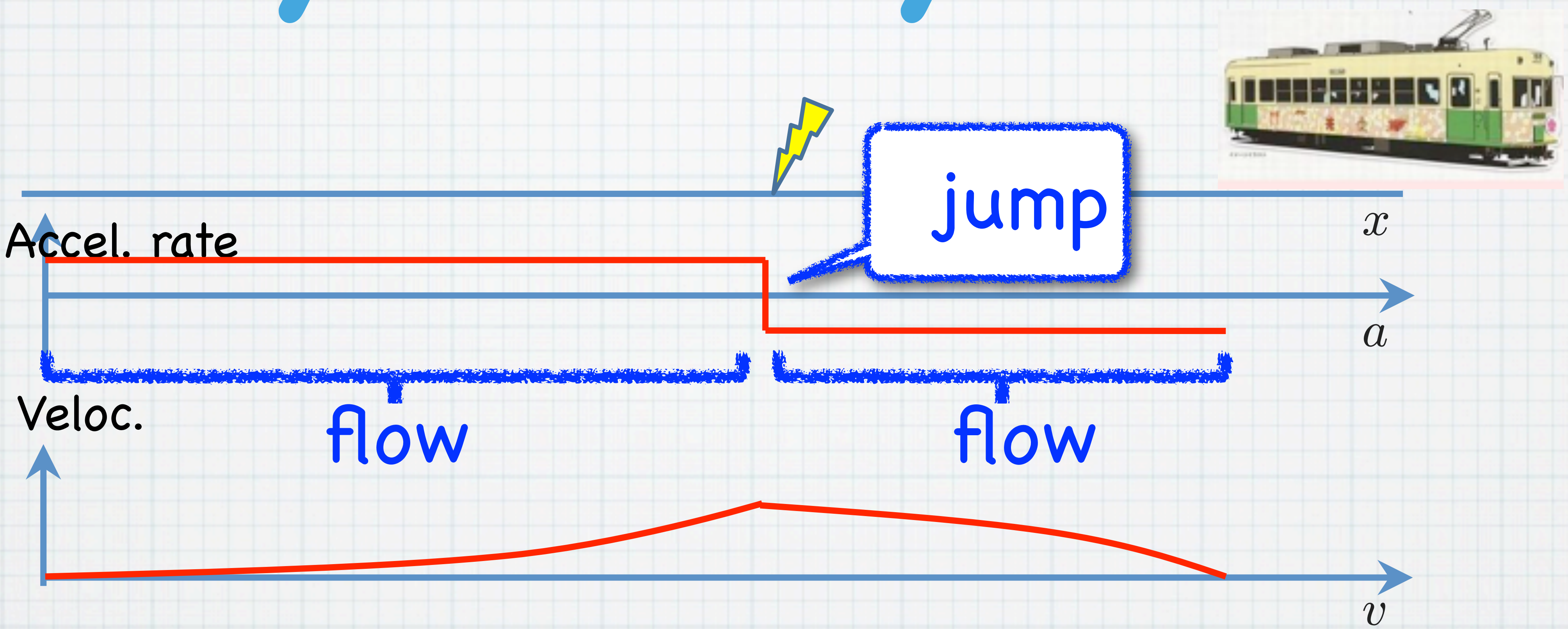


Hybrid System



Hasuo (Tokyo)

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"

Hasuo (Tokyo)

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

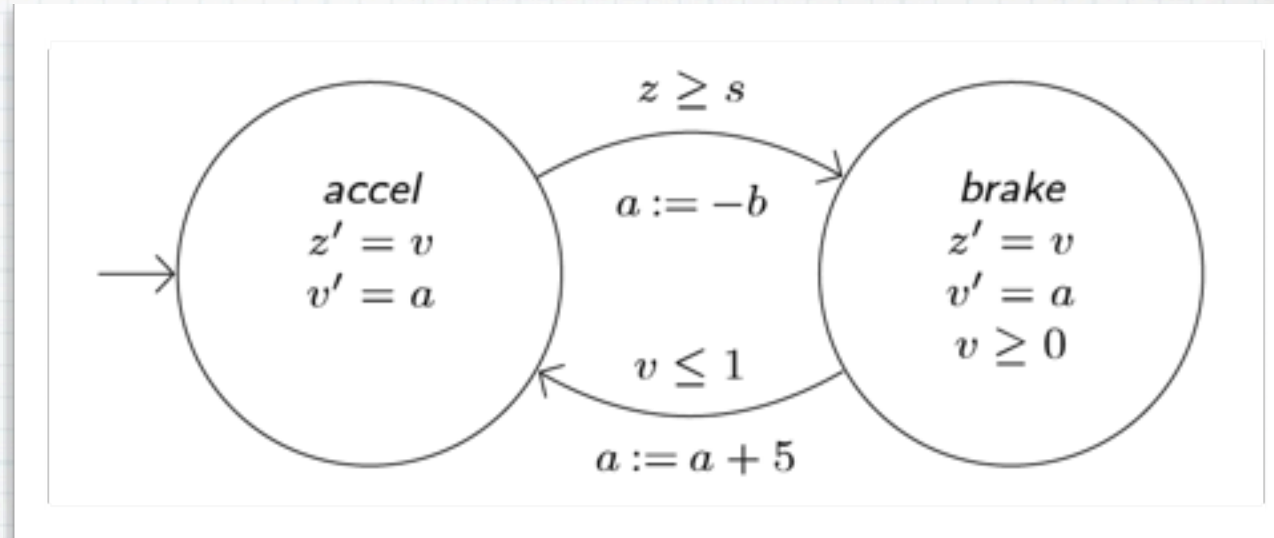
Continuous
"**flow**"

Hybrid!

Control theory
(applied analysis)

Formal Verification Approaches

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



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

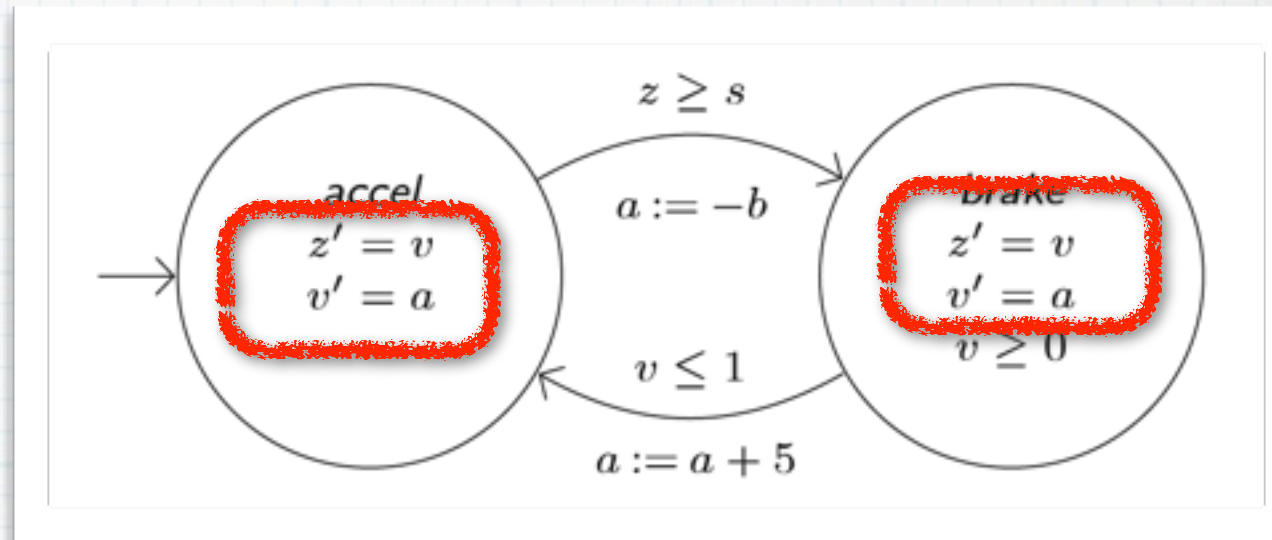
$$[\dot{x} = 1 \text{ while } x \leq 3]\varphi$$

- Differential equations, explicitly
  ➜ distinction jump vs. flow

Formal Verification Approaches

* Hybrid automata

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



* Differential dynamic logic

  [Platzer & others, '07-]

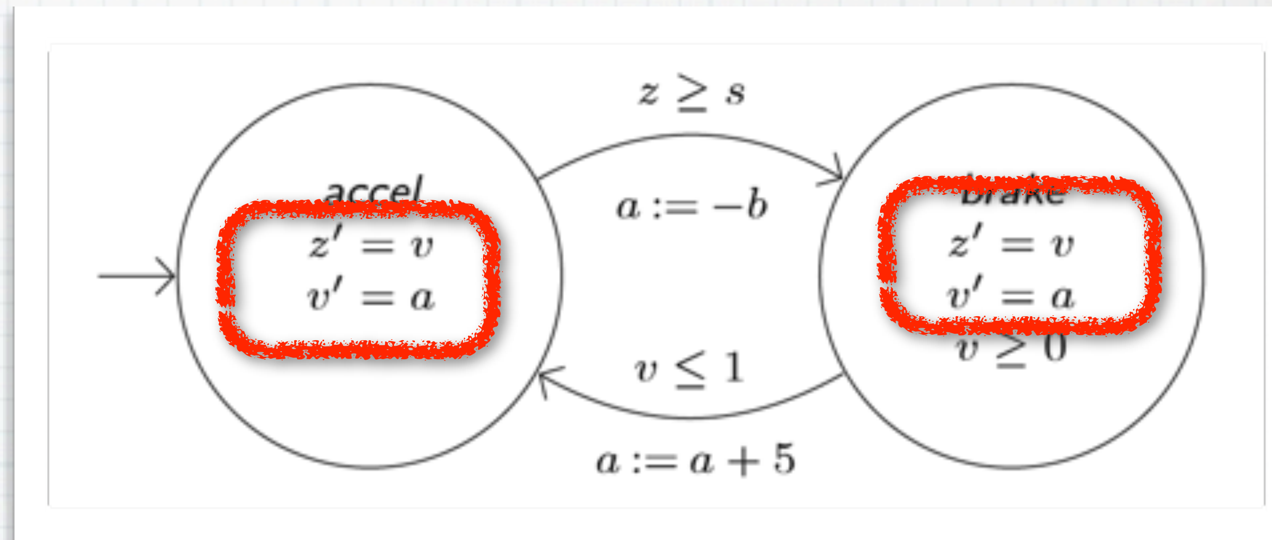
$$[\dot{x} = 1 \text{ while } x \leq 3]\varphi$$

* Differential equations, explicitly

  ➜ distinction jump vs. flow

Formal Verification Approaches

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

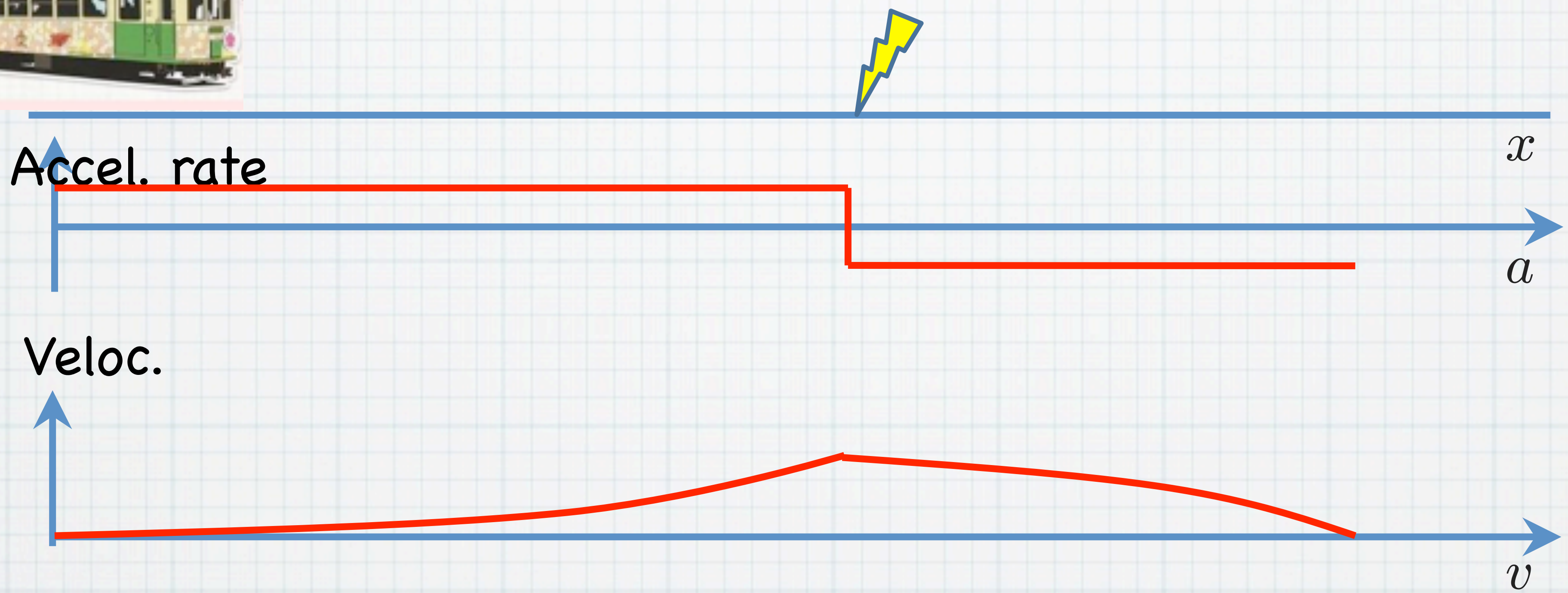


* Differential dynamic logic
  [Platzer & others, '07-]

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

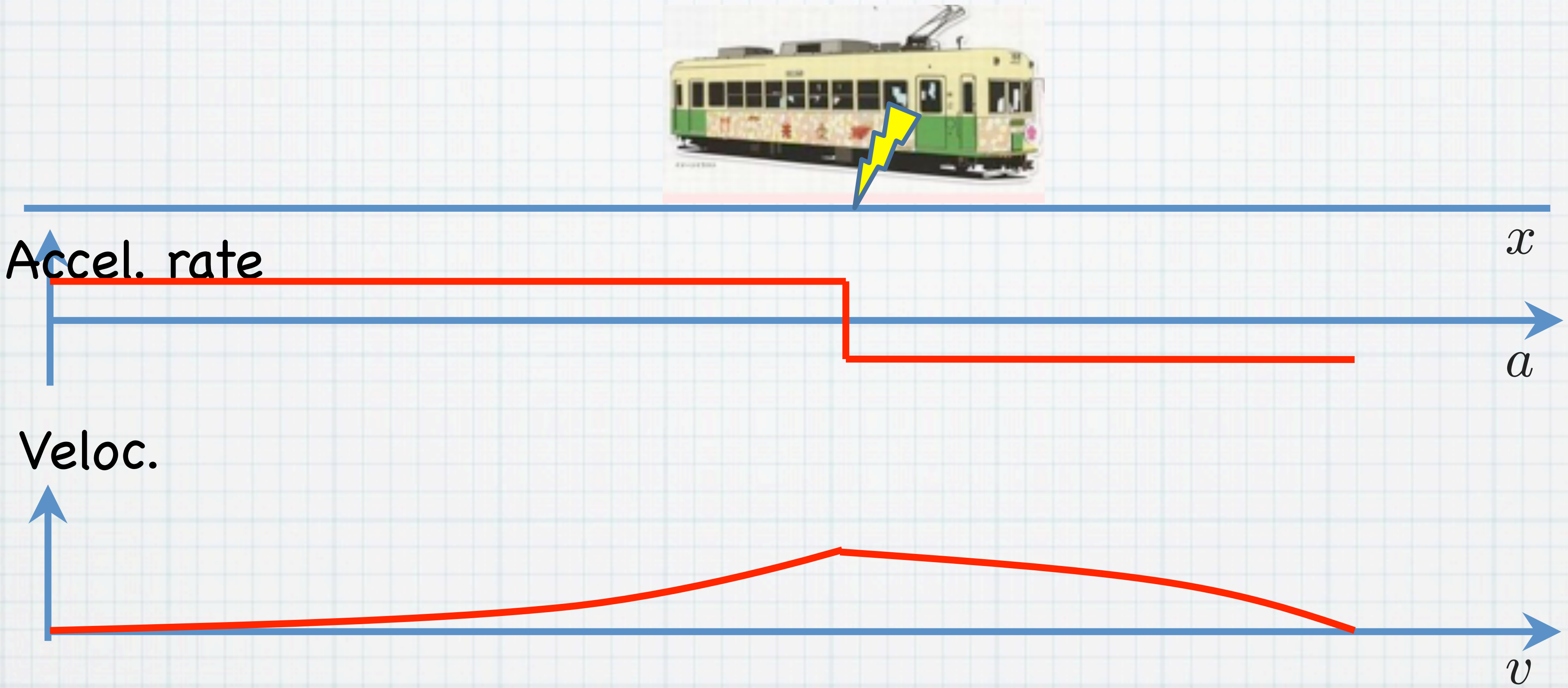
* Differential equations, explicitly
  ➜ distinction jump vs. flow

"Turn Flow into Jump"



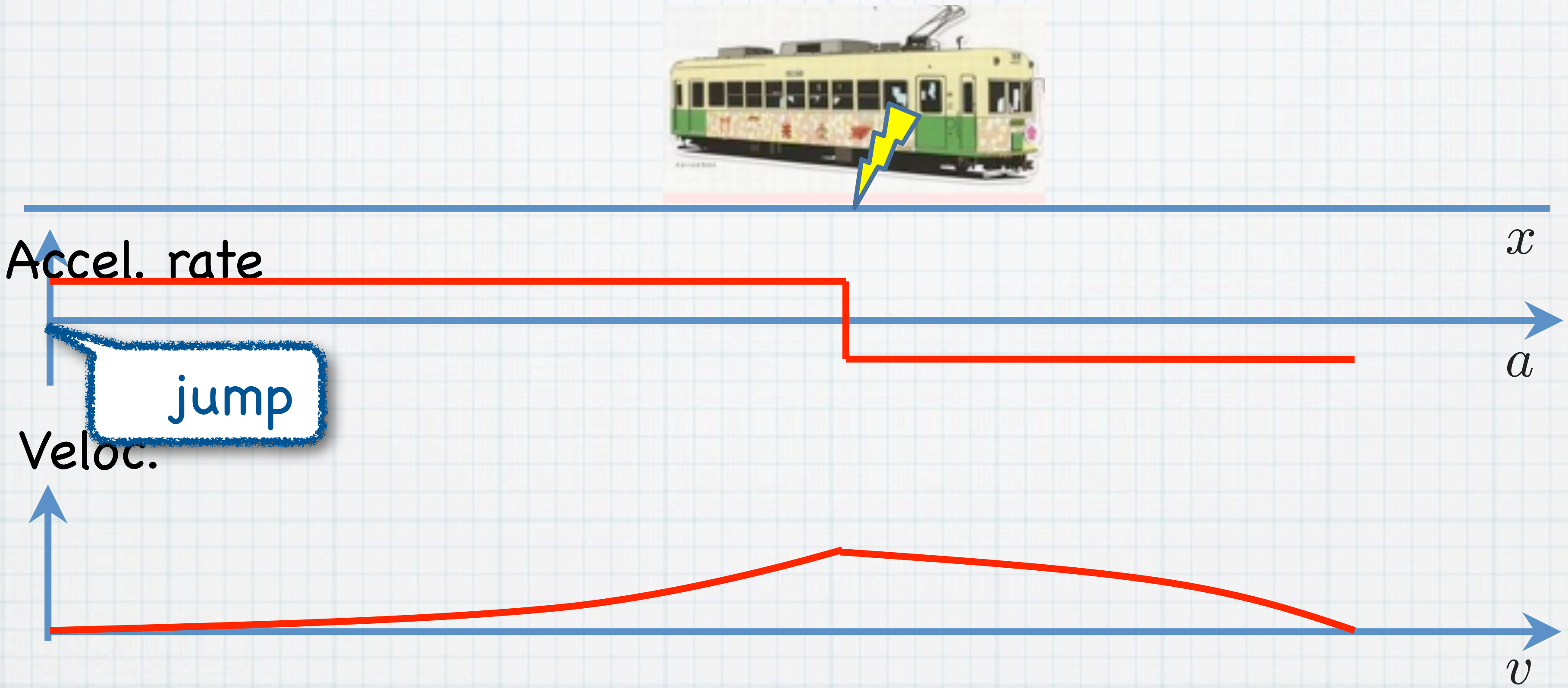
* Flow as infinitely many, infinitely small jumps

"Turn Flow into Jump"



* Flow as infinitely many, infinitely small jumps

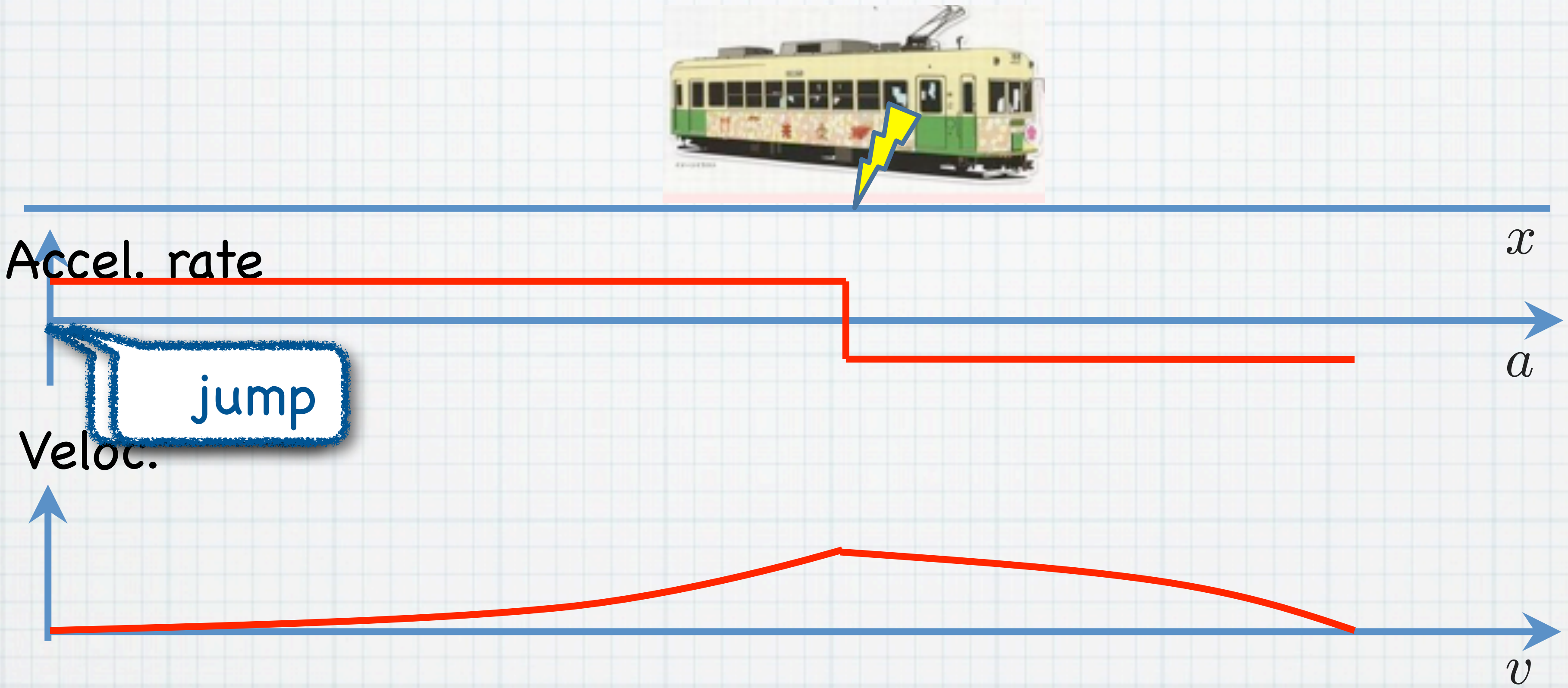
"Turn Flow into Jump"



* Flow as infinitely many, infinitely small jumps

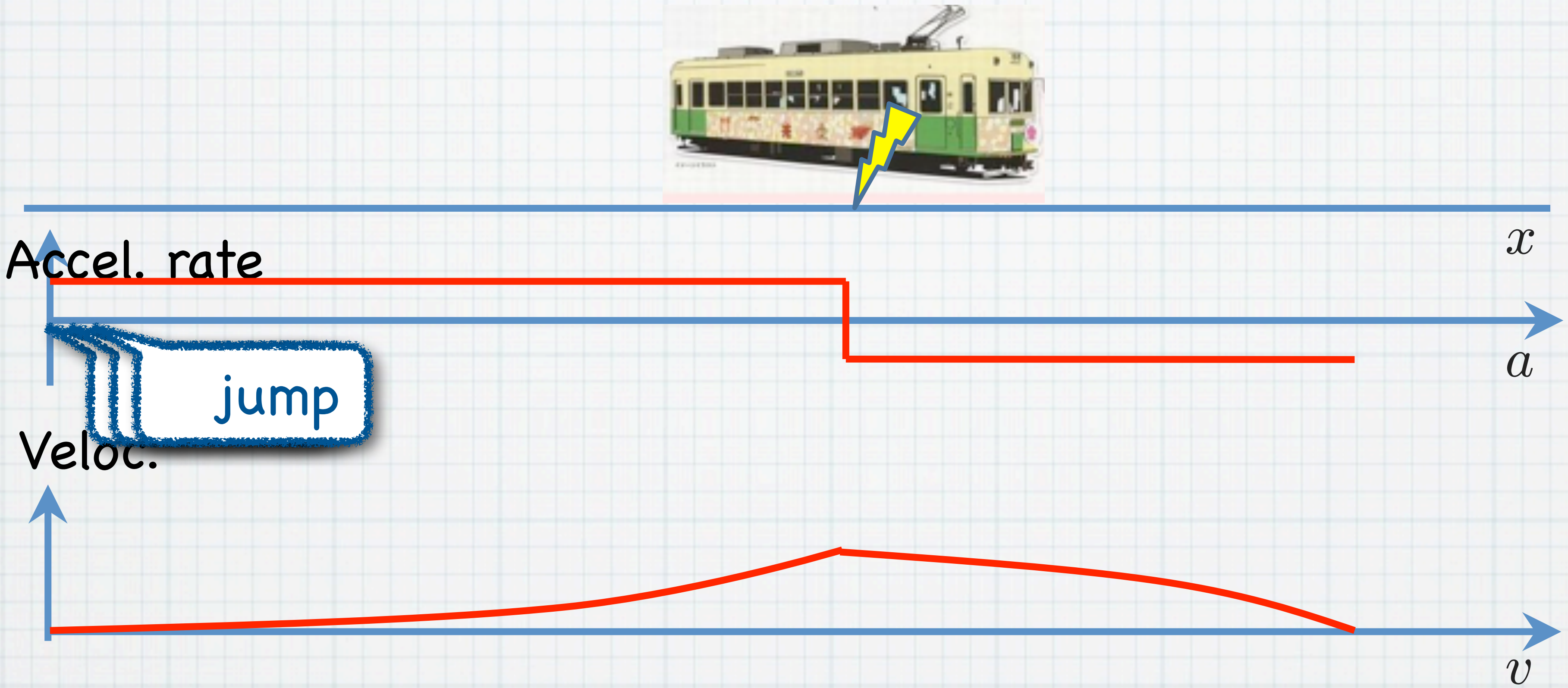
Hasuo (Tokyo)

"Turn Flow into Jump"



* Flow as infinitely many, infinitely small jumps

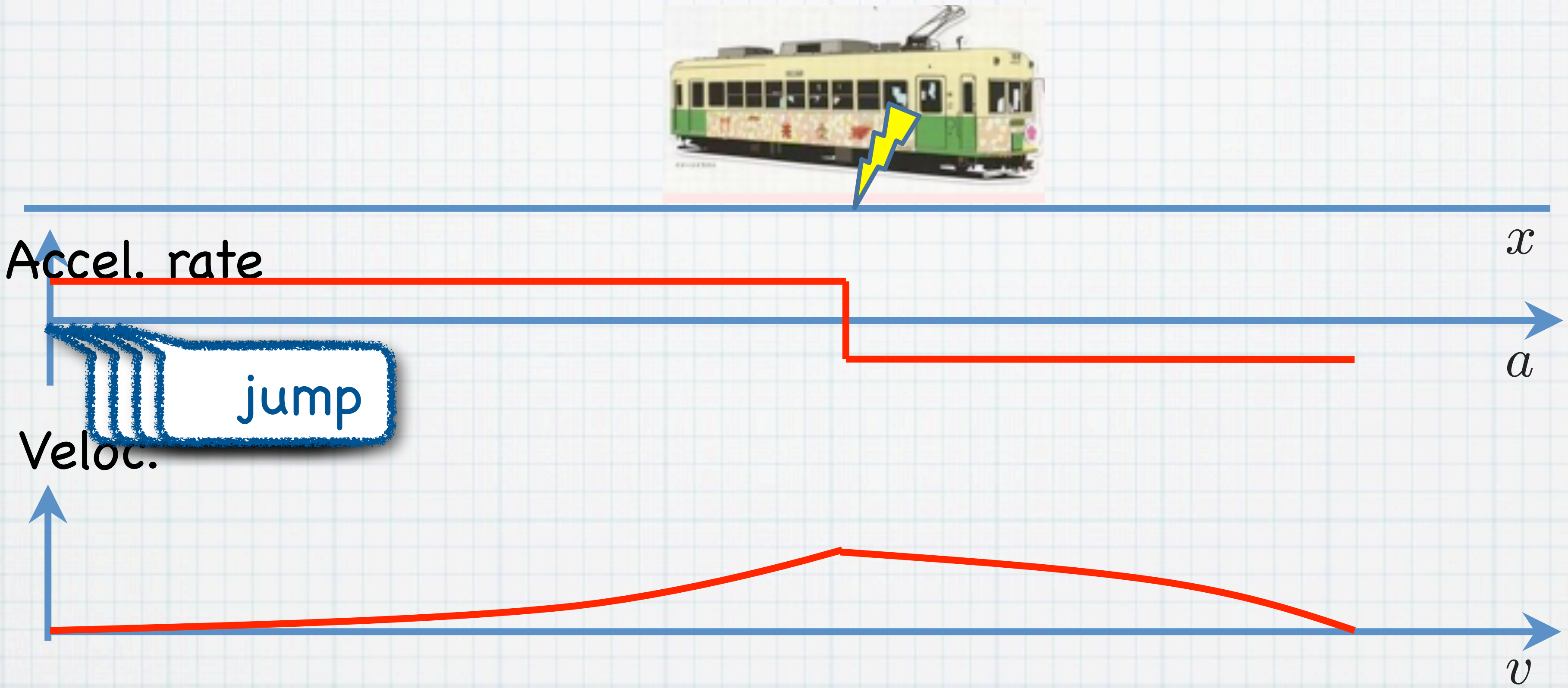
"Turn Flow into Jump"



* Flow as infinitely many, infinitely small jumps

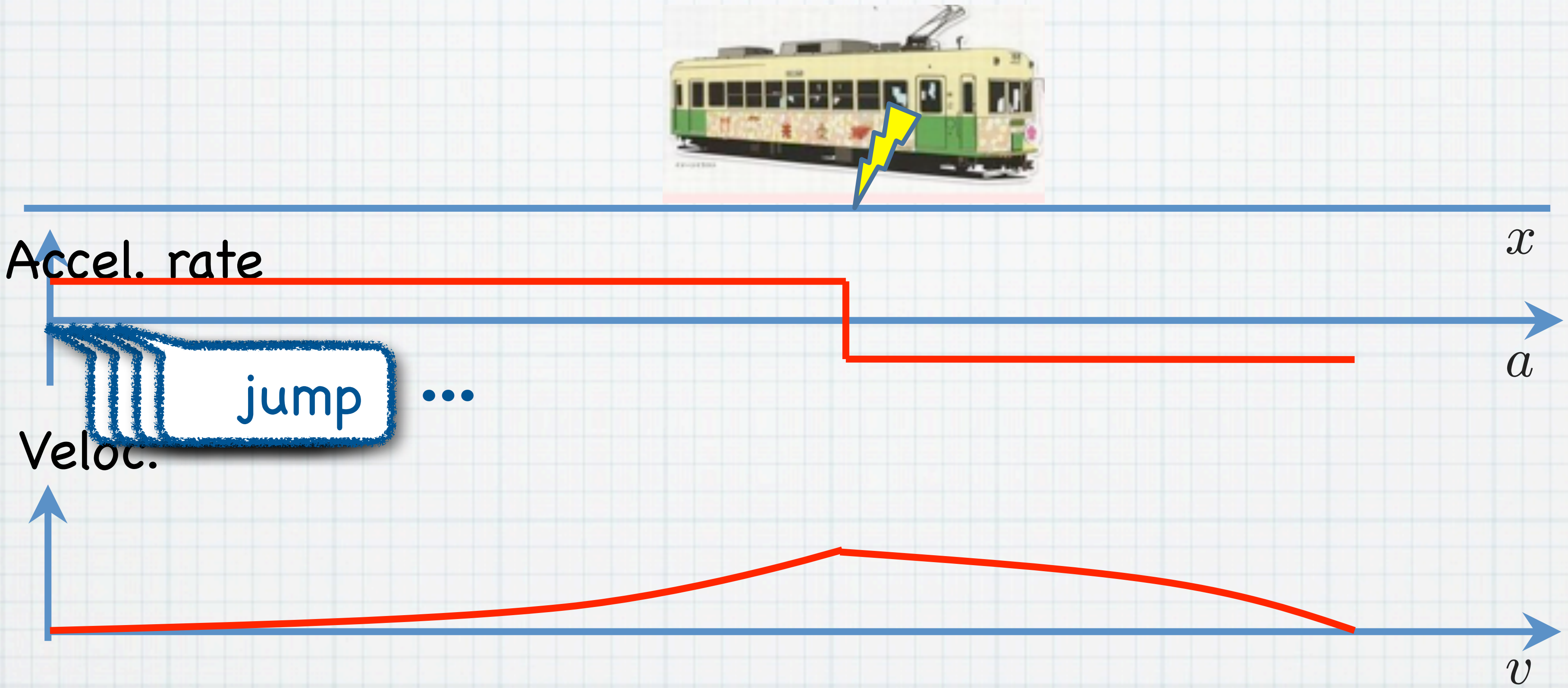
Hasuo (Tokyo)

“Turn Flow into Jump”



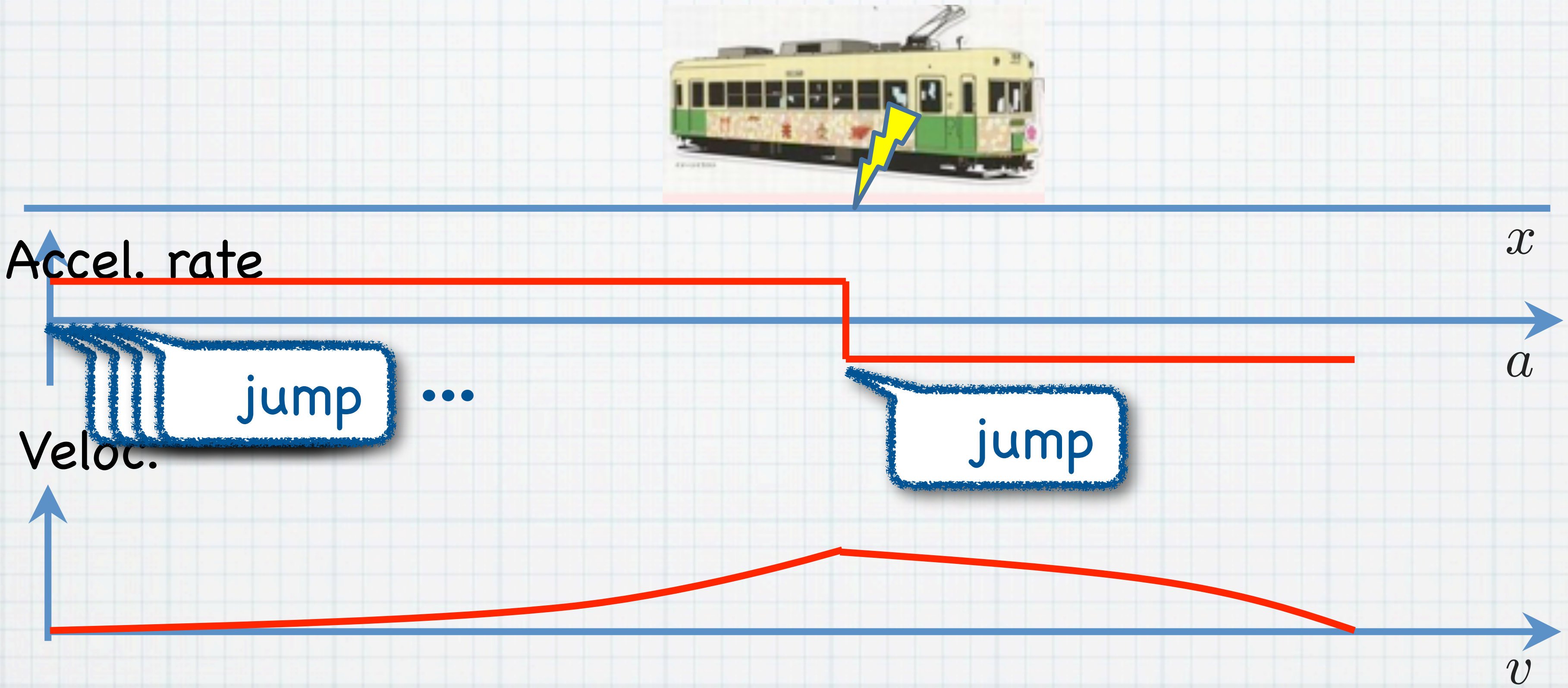
* Flow as infinitely many, infinitely small jumps

"Turn Flow into Jump"



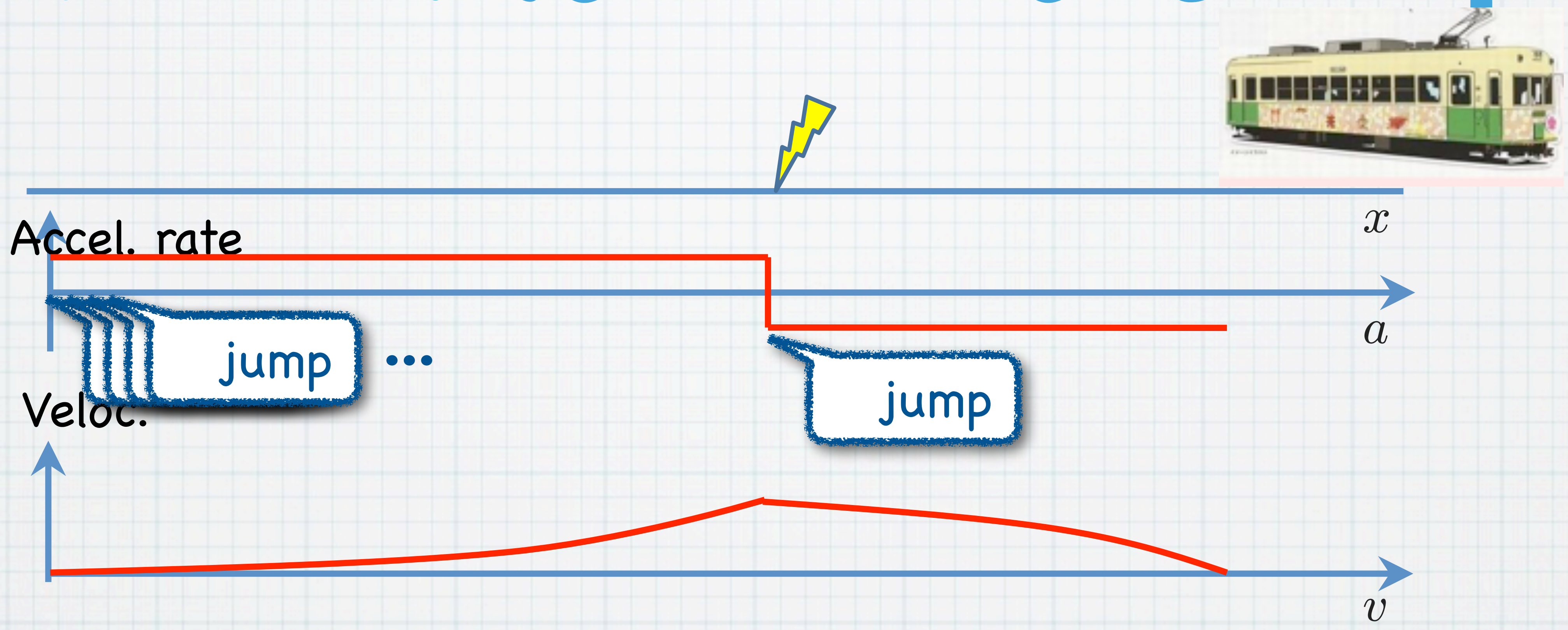
* Flow as infinitely many, infinitely small jumps

"Turn Flow into Jump"



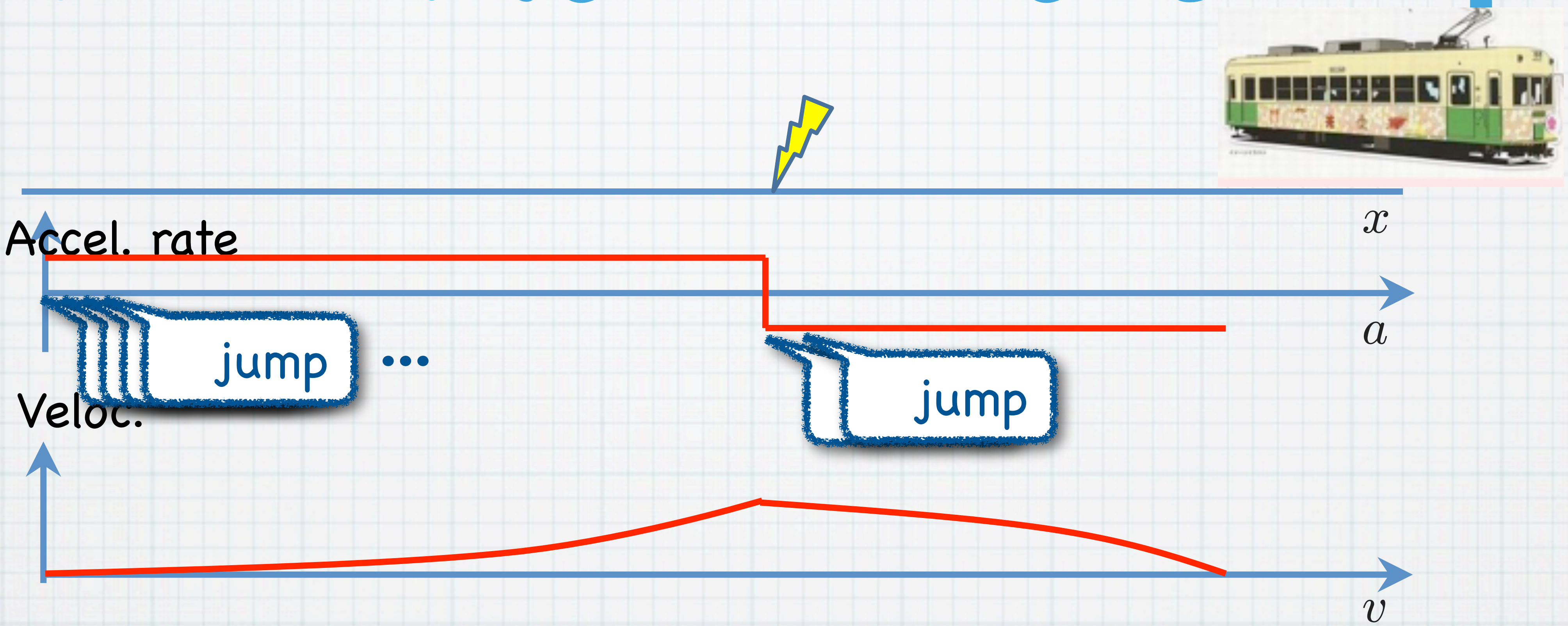
* Flow as infinitely many, infinitely small jumps

"Turn Flow into Jump"



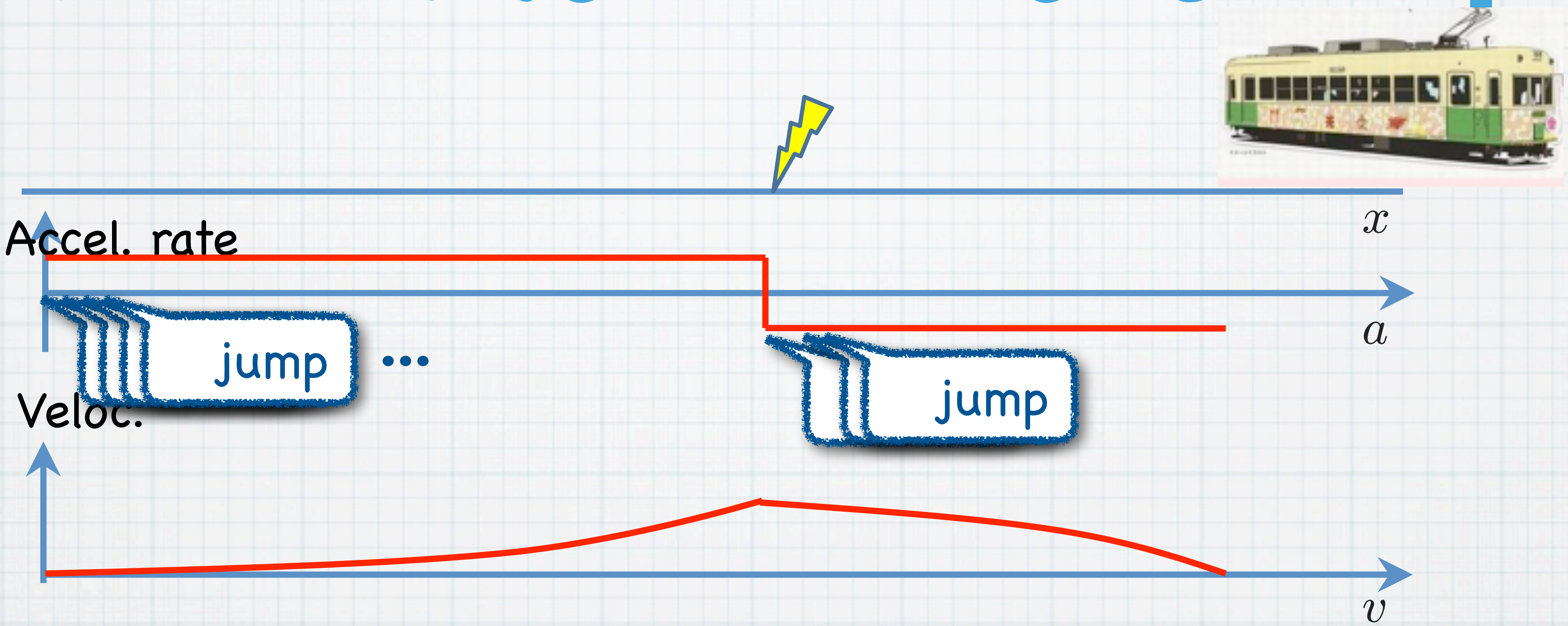
* Flow as infinitely many, infinitely small jumps

"Turn Flow into Jump"



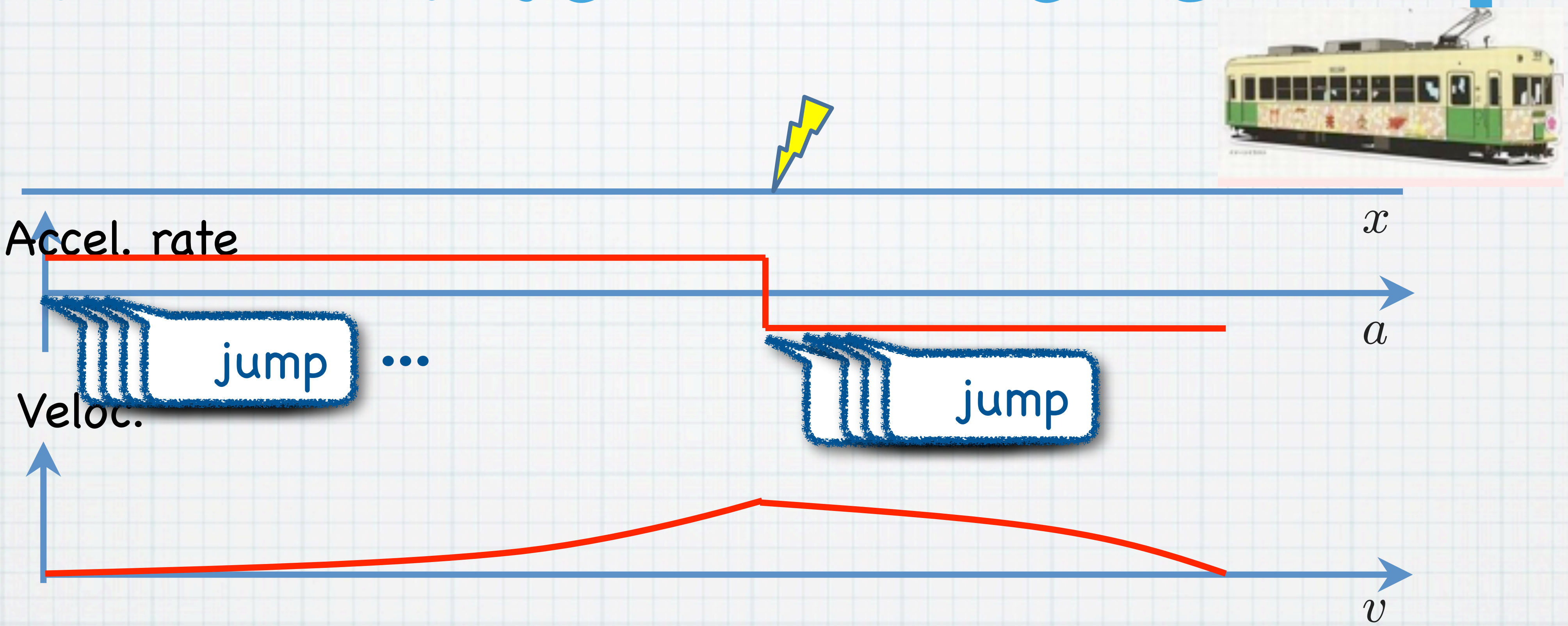
* Flow as infinitely many, infinitely small jumps

"Turn Flow into Jump"



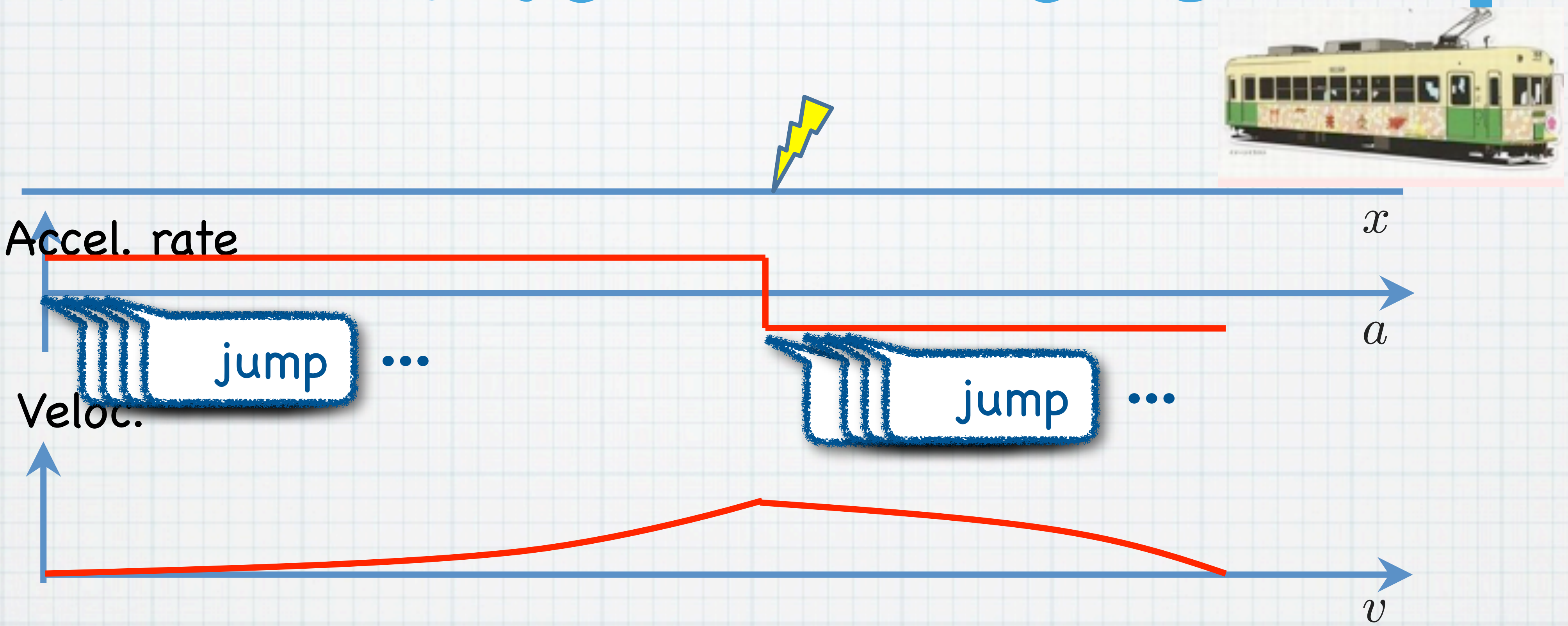
* Flow as infinitely many, infinitely small jumps

"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"

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

"Turn Flow into Jump"

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

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

"Turn Flow into Jump"

```
t := 0 ;
while (t ≤ 1) do {
   t := t + 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 < dt < r
    for **any** positive real r
* t = 1 after the execution?
* Nonstandard analysis!
  [Robinson '60s]

Theoretical Framework [Suenaga&H., ICALP'11]

Theoretical Framework [Suenaga&H., ICALP'11]

The standard textbook [Winskel]

Theoretical Framework [Suenaga&H., ICALP'11]

The standard textbook [Winskel]

While

Programming lang.

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

Theoretical Framework [Suenaga&H., ICALP'11]

The standard textbook [Winskel]

While

Programming lang.

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

Assn

First-order assertion lang.

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

Theoretical Framework

[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

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

Theoretical Framework

[Suenaga&H., ICALP'11]

The standard textbook [Winskel]

While^{dt}

Programming lang.

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

Assn^{dt}

First-order assertion lang.

```
∃z(x=2*z ∧ y=3*z)
```

Hoare^{dt}

Hoare-style program logic

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

Theoretical Framework

[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

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

Rigorous semantics by nonstandard analysis

Theoretical Framework

[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 {`
`  t:=t+1;`
`  if ...`
`}`
	`∃z(x=2*z ∧ y=3*z)`	$$\frac{\{A \wedge b\}\, c\, \{A\}}{\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{A \wedge \neg b\}}$$

Rigorous semantics by nonstandard analysis

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

Static Analysis

Nonstandard Analysis

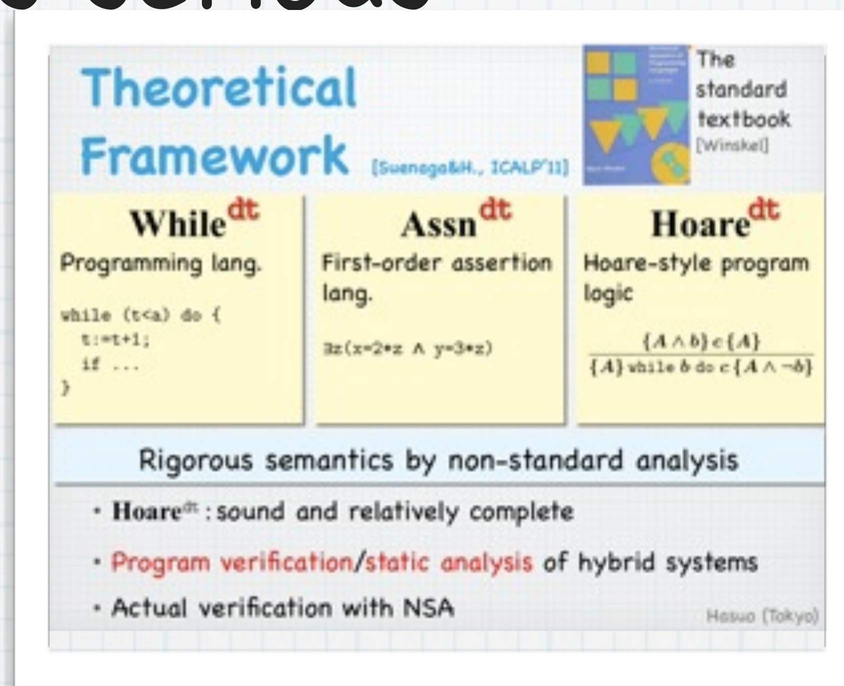
Nonstandard Static Analysis

Hasuo (Tokyo)

Nonstandard Static Analysis

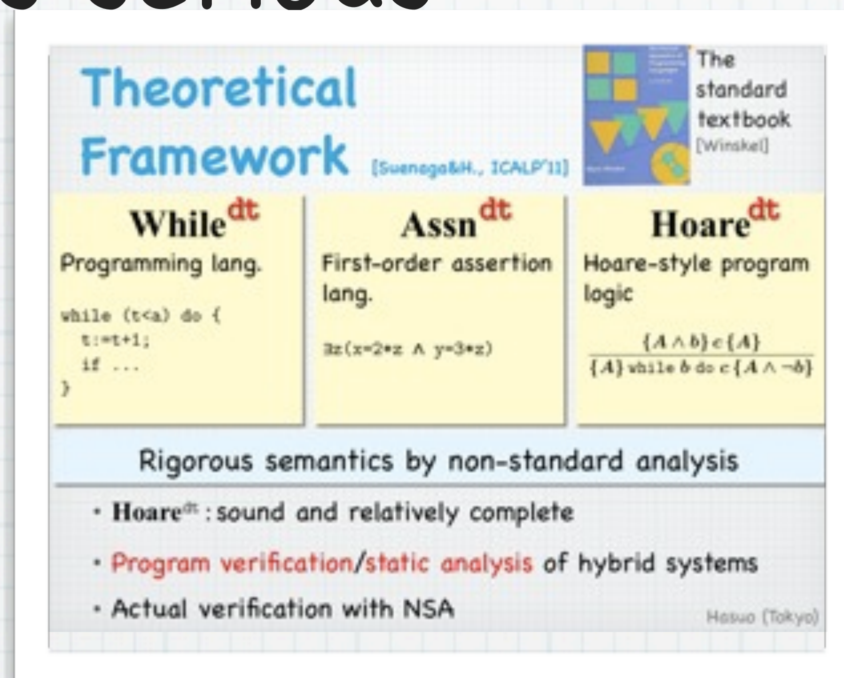
Nonstandard Static Analysis

* Towards serious use of



Nonstandard Static Analysis

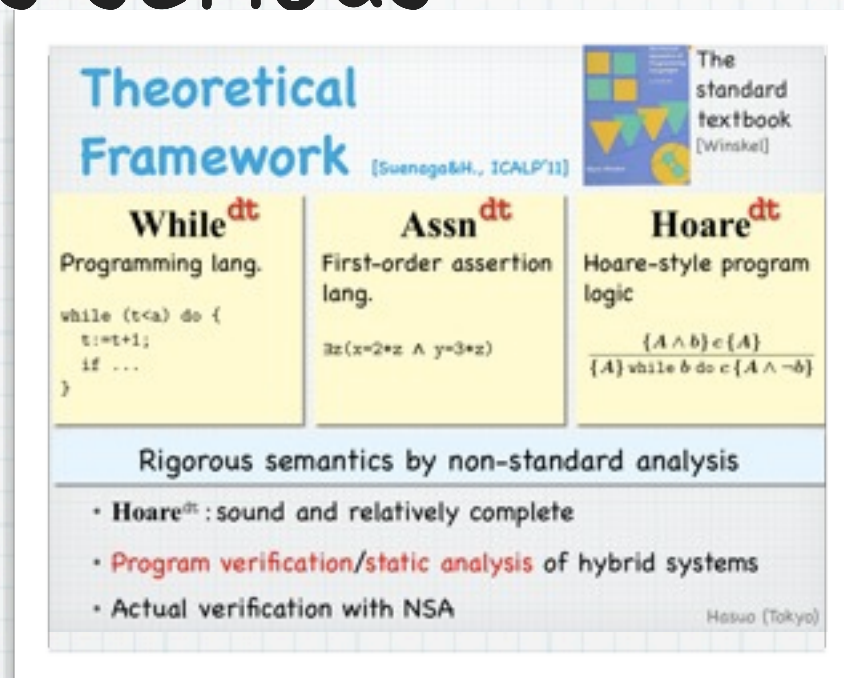
- Towards serious use of



- Static analysis techniques **transferred** to hybrid appl.

Nonstandard Static Analysis

- Towards serious use of

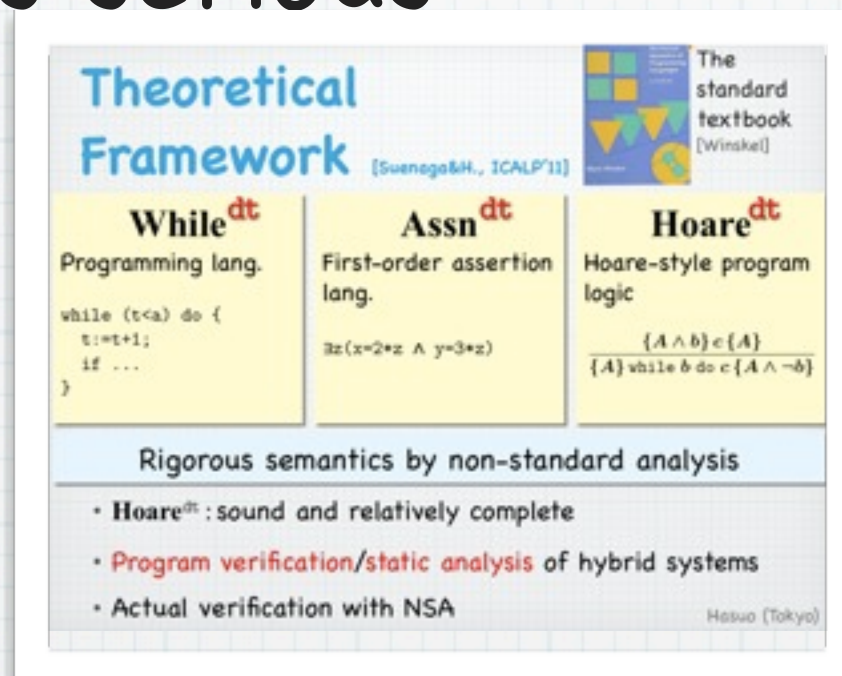


Exactly as they are!

- Static analysis techniques **transferred** to hybrid appl.

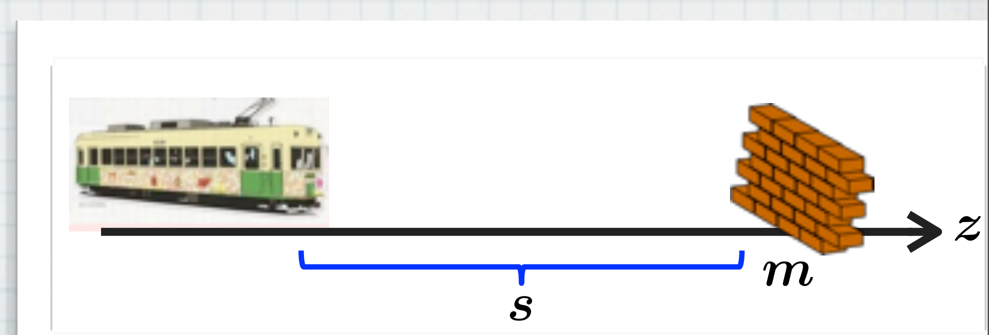
Nonstandard Static Analysis

- Towards serious use of



Exactly as they are!

- Static analysis techniques **transferred** to hybrid appl.
- Leading example: ETCS



Prototype Automatic Prover

Hoare^{dt} Analyzer

Prototype Automatic Prover

P

(**While**^{dt} program) →

B

(postcondition)

Hoare^{dt} Analyzer

Prototype Automatic Prover

P

(**While**^{dt} program) →

B

(postcondition)

Hoare^{dt} Analyzer

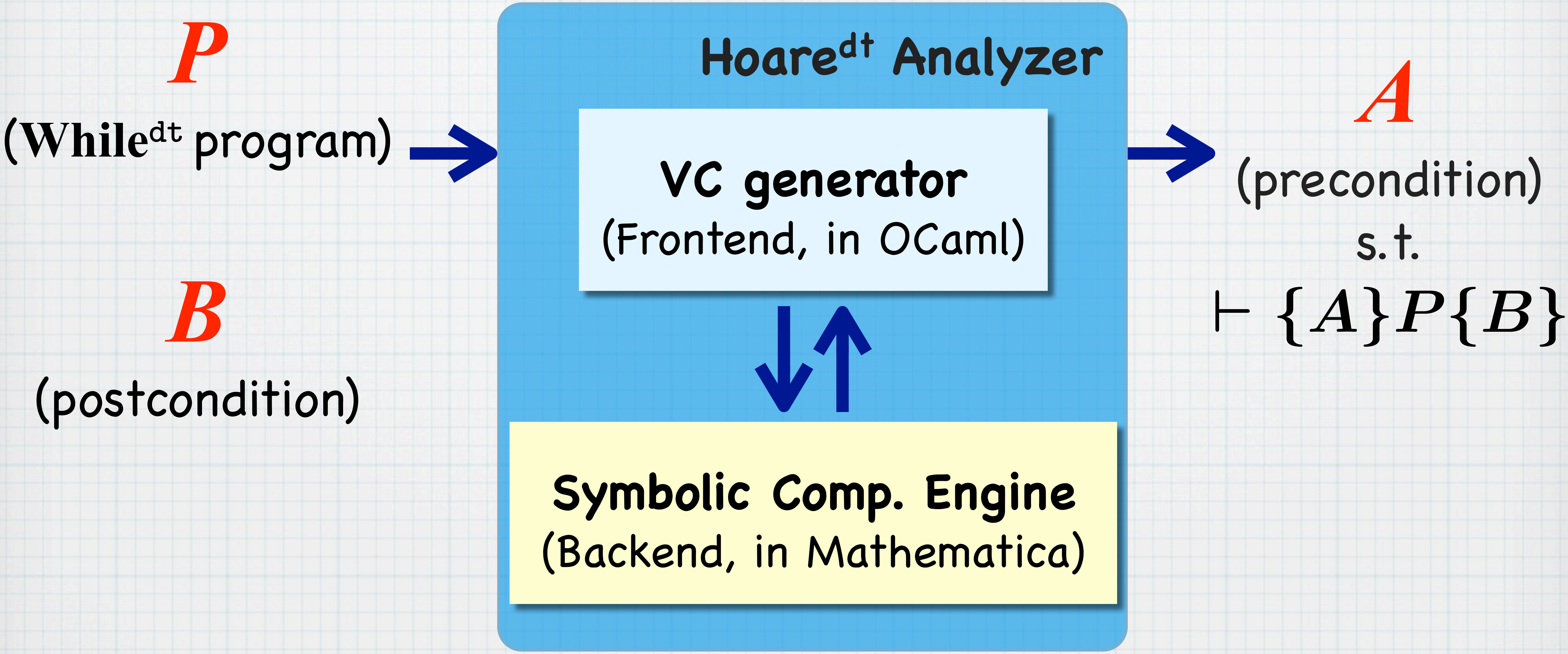
→ A

(precondition)

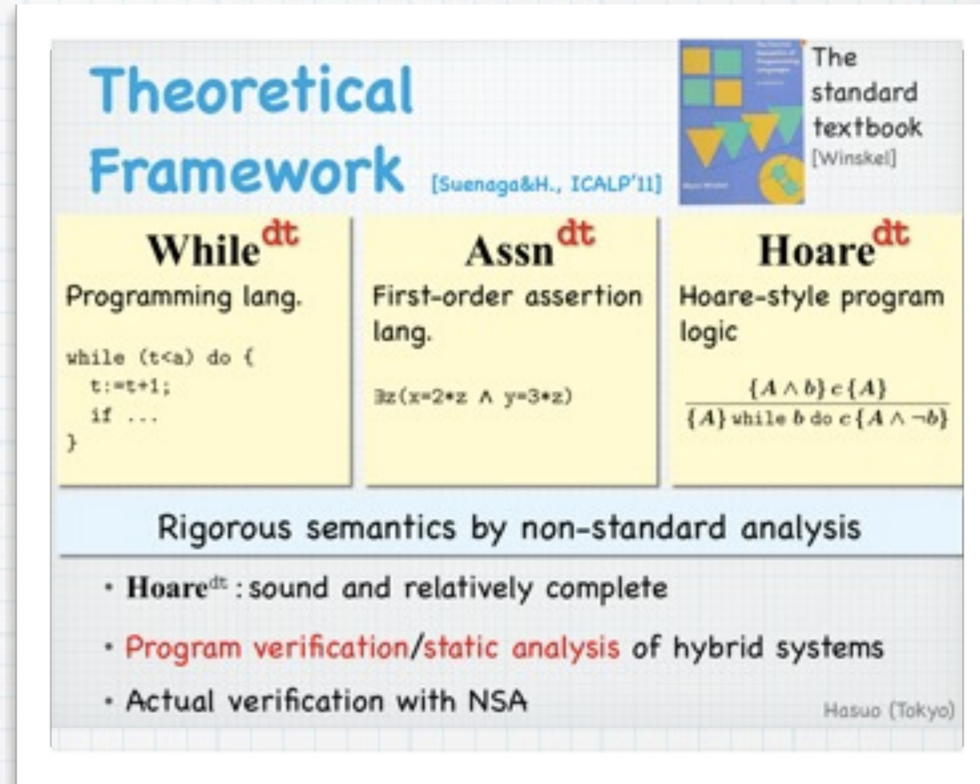
s.t.

$\vdash \{A\}P\{B\}$

Prototype Automatic Prover

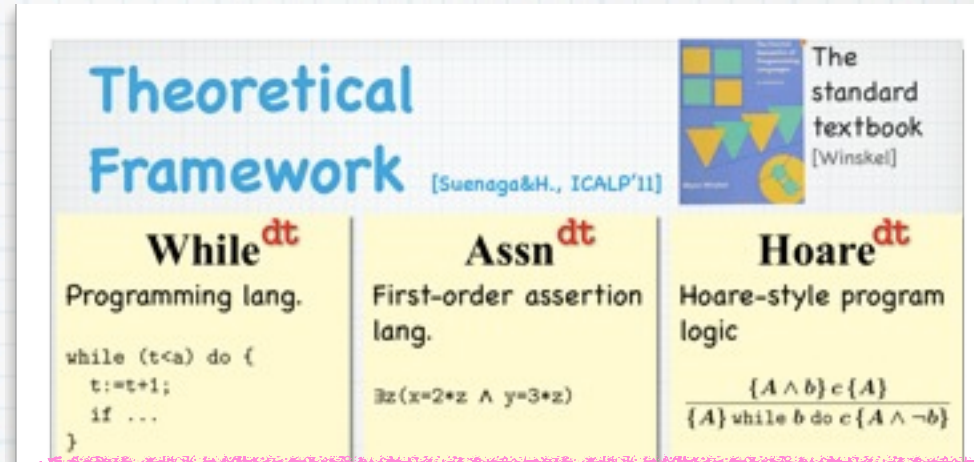


Outline



- **Theoretical foundations**
  - $\mathbf{While}^{\mathrm{dt}}$, $\mathbf{Assn}^{\mathrm{dt}}$, $\mathbf{Hoare}^{\mathrm{dt}}$
  - Rigorous semantics via NSA
  - Transfer principle, "sectionwise lemmas"
- **Static analysis techniques, transferred as they are**
  - Phase split [Sharma,Dillig,Dillig,Aiken; CAV'11] [Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]
  - Differential invariant [Platzer,Clarke; CAV'08]
  - ... and more!

Outline

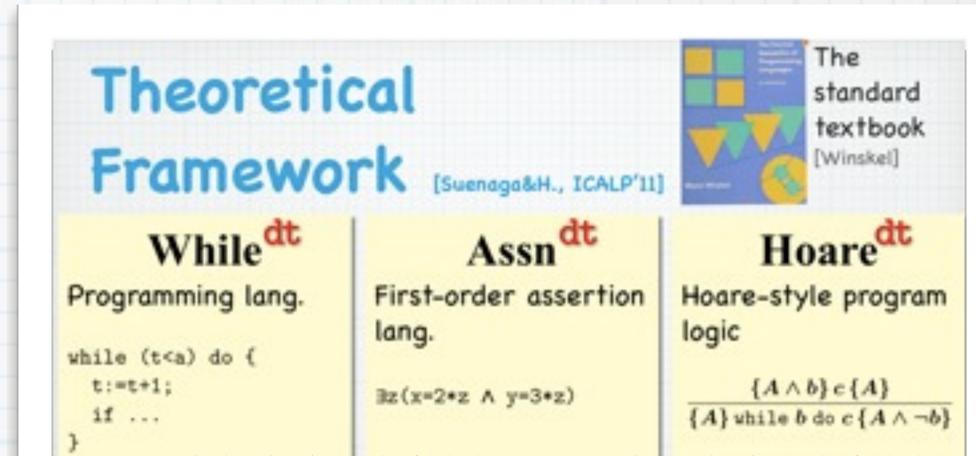


w/ or w/o dt ...

➜ logically "**the same**"

- **Theoretical foundations**
  - **While^{dt}, Assn^{dt}, Hoare^{dt}**
  - Rigorous semantics via NSA
  - Transfer principle, "sectionwise lemmas"
- **Static analysis techniques, transferred as they are**
  - Phase split [Sharma,Dillig,Dillig,Aiken; CAV'11] [Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]
  - Differential invariant [Platzer,Clarke; CAV'08]
  - ... and more!

Outline



- **Theoretical foundations**
  - $\mathbf{While}^{\mathrm{dt}}$, $\mathbf{Assn}^{\mathrm{dt}}$, $\mathbf{Hoare}^{\mathrm{dt}}$
  - Rigorous semantics via NSA
  - Transfer principle, "sectionwise lemmas"

w/ or w/o dt ...

➜ logically "**the same**"

- **Static analysis techniques, transferred as they are**
  - Phase split [Sharma,Dillig,Dillig,Aiken; CAV'11] [Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]
  - Differential invariant [Platzer,Clarke; CAV'08]
  - ... and more!

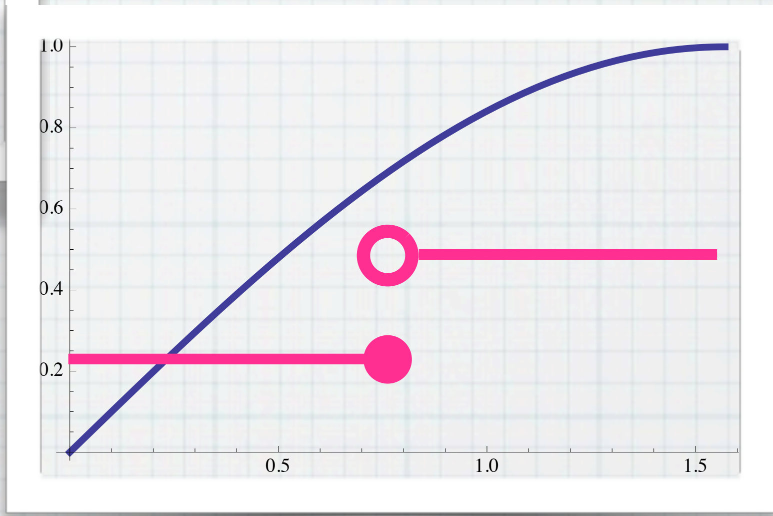
Part I:

Theoretical Foundations

Nonstandard Analysis

- Analysis with an **infinitesimal** ∂, e.g.

$$f \text{ is continuous} \iff \left(\begin{array}{l} |x - x'| \text{ is infinitesimal} \\ \implies |f(x) - f(x')| \text{ is infinitesimal} \end{array} \right)$$



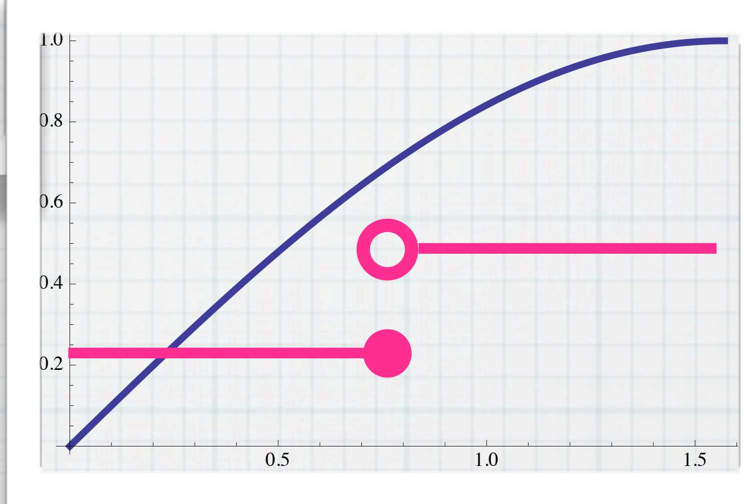
- Done naively ➜ contradiction!

Nonstandard Analysis

* Analysis with an **infinitesimal** ∂, e.g.

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

$$f \text{ is continuous} \iff \left(\begin{array}{l} |x - x'| \text{ is infinitesimal} \\ \implies |f(x) - f(x')| \text{ is infinitesimal} \end{array} \right)$$



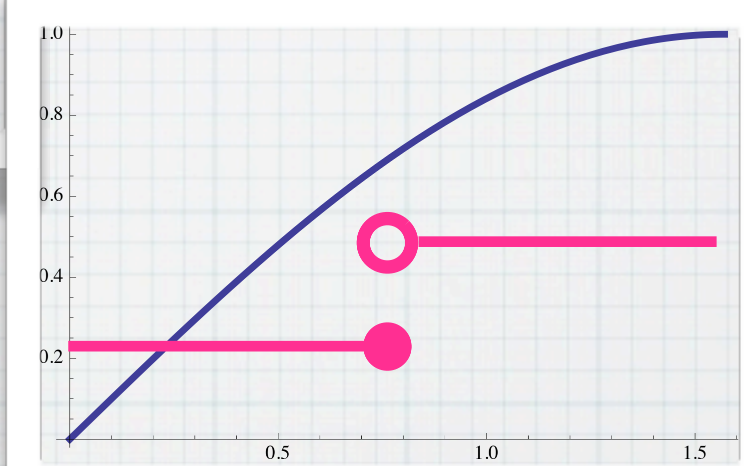
* Done naively ➜ contradiction!

Nonstandard Analysis

* Analysis with an **infinitesimal** ∂, e.g.

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

$$f \text{ is continuous} \iff \left(\begin{array}{l} |x - x'| \text{ is infinitesimal} \\ \implies |f(x) - f(x')| \text{ is infinitesimal} \end{array} \right)$$



* Done naively ➜ contradiction!

Logical foundation via an ultrafilter

[Robinson,1960]

Hyperreals

= Reals + Infinitesimals + ...

Defn.

The set of *hyperreal numbers* is

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

Ignore

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

Ignore

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

Ignore

0th section

1st section

2nd section

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

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 + ...

Defn.
The set of *hyperreal numbers* is

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

0th section

1st section

2nd section

Ignore

- Operations: sectionwise

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

- Reals are hyperreals

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

Hyperreals

= Reals + Infinitesimals + ...

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"

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

"For sufficiently large i"
"Except for finitely many i"

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

* 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

"For sufficiently large i"
"Except for finitely many i"

Hyperreals

= Reals + Infinitesimals + ...

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"

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

"For sufficiently large i"
"Except for finitely many i"

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 + ...

Defn.
The set of *hyperreal numbers* is

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

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

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

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

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.

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

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

$\Longleftrightarrow$ $a_i < b_i$ "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 + ...

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

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

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

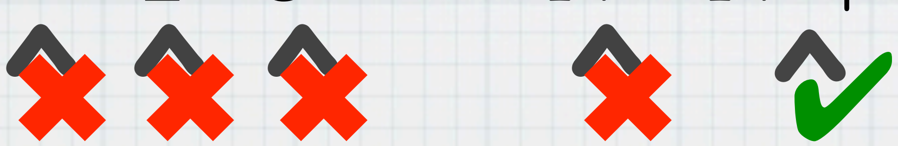
$$ {}^*\mathbb{R} \;:=\; \mathbb{R}^{\mathbb{N}} / \sim_{\mathcal{F}} $$

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

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

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

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



Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

$\iff$ $a_i < b_i$ "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.

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

✗ ✗ ✗ ✗ ✓ ✓

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

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)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

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

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

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

Hype

= Reals + Inf

Defn.
The set of *hyperreal numbers* is

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

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. $\emptyset \notin \mathcal{F}$

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

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.
$^*A$: its $*$-*transform*. Then

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

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.
$^*A$: its $*$-*transform*. Then

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

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

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.
$^*A$: 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"

Theoretical Framework

[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

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

Rigorous semantics by nonstandard analysis

Syntax

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

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

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

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

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

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

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

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

Tokyo)

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

Syntax

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

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

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

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

Tokyo)

While$^{\mathrm{dt}}$

Syntax

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

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$^{\mathrm{dt}}$

While + dt

Syntax

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	$::=$	$\mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2$
Cmd $\ni$	c	$::=$	$\mathtt{skip} \mid x := a \mid c_1 ; c_2$
			$\mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c$

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

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

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

Tokyo)

Syntax

While + dt

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

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

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

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

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

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

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

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

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

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

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

Tokyo)

Syntax

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 $r \in \mathbb{R}$, $\mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}$

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

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

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)

Syntax

While^{dt}

While + dt

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

Assn^{dt}

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

Hoare^{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

While$^{\mathrm{dt}}$

While + dt

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

Assn$^{\mathrm{dt}}$

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

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

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

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

Tokyo)

Syntax

While + dt

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

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

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

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

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

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

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

Tokyo)

Syntax

While^dt — While + dt

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

Assn^dt — Assn, *-transformed

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

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

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

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

Tokyo)

Syntax

While^{dt}

While + dt

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

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

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

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

Assn^{dt}

Assn, *-transformed

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

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)

Syntax

While^{dt}

While + dt

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

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

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

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

Assn^{dt}

Assn, *-transformed

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

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$^{\mathrm{dt}}$

While + dt

Syntax

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

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

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

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

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

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

Hoare$^{\mathrm{dt}}$

Syntax

While$^{\mathtt{dt}}$

While + dt

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 $r \in \mathbb{R}$, $\mathtt{aop} \in \{+, -, \cdot, \wedge, /\}$

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$^{\mathtt{dt}}$

Precisely the same rules

Syntax

While^{dt} — While + dt

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

Assn^{dt} — Assn, *-transformed

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

Hoare^{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

While$^{\mathrm{dt}}$

While + dt

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

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

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

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

$< a_2 \mid$

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

Hoare$^{\mathrm{dt}}$

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)

Denotational Semantics

- Execute sectionwise and bundle up the outcomes!

```
t := 0;
while (t < 1)
  t := t + dt;
```

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := 0;
while (t < 1)
  t := t + dt;
```

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := (0,0,0,...);
while (t < (1,1,1,...))
```

$$\texttt{t := t +} \left(1, \frac{1}{2}, \frac{1}{3}, \dots\right) \ ;$$

Denotational Semantics

- Execute sectionwise and bundle up the outcomes!

0th section	1st section	2nd section	
`t := 0;` `while (t < 1)` $t := t + 1;$	`t := 0;` `while (t < 1)` $t := t + \frac{1}{2};$	`t := 0;` `while (t < 1)` $t := t + \frac{1}{3};$	...

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

0th section	1st section	2nd section	

`t := 0;`
`while (t < 1)`
`t := t + 1 ;`
	`t := 0;`
	`while (t < 1)`
	$t := t + \frac{1}{2}$;
		`t := 0;`
		`while (t < 1)`
		$t := t + \frac{1}{3}$;
			…

t = 1	t = 1	t = 1	…

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

$$\texttt{t := } (0,0,0,\ldots)\texttt{;}$$
$$\texttt{while (t < } (1,1,1,\ldots)\texttt{)}$$
$$\texttt{t := t + } \left(1, \frac{1}{2}, \frac{1}{3}, \ldots\right) \texttt{;}$$

$$\texttt{t = } (1,1,1,\ldots)$$

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := 0;
while (t < 1)
  t := t + dt;
```

```
t = 1
```

Denotational Semantics

$$[\![x]\!]\sigma := \sigma(x) \qquad [\![\mathtt{c}_r]\!]\sigma := r \quad \text{for each } r \in \mathbb{R}$$

$$[\![a_1 \ \mathtt{aop}\ a_2]\!]\sigma := [\![a_1]\!]\sigma \ \mathtt{aop}\ [\![a_2]\!]\sigma$$

$$[\![\mathtt{dt}]\!]\sigma := \omega^{-1} = \left[\,(1, \tfrac{1}{2}, \tfrac{1}{3}, \dots)\,\right] \qquad [\![\infty]\!]\sigma := \omega = \left[\,(1, 2, 3, \dots)\,\right]$$

$$[\![\mathtt{true}]\!]\sigma := \mathtt{tt} \qquad [\![\mathtt{false}]\!]\sigma := \mathtt{ff}$$

$$[\![b_1 \wedge b_2]\!]\sigma := [\![b_1]\!]\sigma \wedge [\![b_2]\!]\sigma \qquad [\![\neg b]\!]\sigma := \neg([\![b]\!]\sigma)$$

$$[\![a_1 < a_2]\!]\sigma := [\![a_1]\!]\sigma < [\![a_2]\!]\sigma$$

$$[\![\mathtt{skip}]\!]\sigma := \sigma \qquad [\![x := a]\!]\sigma := \sigma\left[\,x \mapsto [\![a]\!]\sigma\,\right] \qquad [\![c_1; c_2]\!]\sigma := [\![c_2]\!]\left(\,[\![c_1]\!]\sigma\,\right)$$

$$[\![\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2]\!]\sigma := \begin{cases} [\![c_1]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathtt{tt} \\ [\![c_2]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathtt{ff} \end{cases}$$

$$[\![\mathtt{while}\ b\ \mathtt{do}\ c]\!]\sigma := \Big(\,[\![\,(\mathtt{while}\ b\ \mathtt{do}\ c)|_i\,]\!](\sigma|_i)\,\Big)_{i \in \mathbb{N}}$$

Hasuo (Tokyo)

Denotational Semantics

$$[\![x]\!]\sigma := \sigma(x) \qquad [\![\mathtt{c}_r]\!]\sigma := r \quad \text{for each } r \in \mathbb{R}$$

$$[\![a_1 \ \mathtt{aop}\ a_2]\!]\sigma := [\![a_1]\!]\sigma \ \mathtt{aop}\ [\![a_2]\!]\sigma$$

$$[\![\mathtt{dt}]\!]\sigma := \omega^{-1} = \left[\,(1, \tfrac{1}{2}, \tfrac{1}{3}, \dots)\,\right] \qquad [\![\infty]\!]\sigma := \omega = \left[\,(1, 2, 3, \dots)\,\right]$$

$$[\![\mathtt{true}]\!]\sigma := \mathrm{tt} \qquad [\![\mathtt{false}]\!]\sigma := \mathrm{ff}$$

$$[\![b_1 \wedge b_2]\!]\sigma := [\![b_1]\!]\sigma \wedge [\![b_2]\!]\sigma \qquad [\![\neg b]\!]\sigma := \neg([\![b]\!]\sigma)$$

$$[\![a_1 < a_2]\!]\sigma := [\![a_1]\!]\sigma < [\![a_2]\!]\sigma$$

$$[\![\mathtt{skip}]\!]\sigma := \sigma \qquad [\![x := a]\!]\sigma := \sigma\left[\,x \mapsto [\![a]\!]\sigma\,\right] \qquad [\![c_1; c_2]\!]\sigma := [\![c_2]\!]\left(\,[\![c_1]\!]\sigma\,\right)$$

$$[\![\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2]\!]\sigma := \begin{cases} [\![c_1]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathrm{tt} \\ [\![c_2]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathrm{ff} \end{cases}$$

$$[\![\mathtt{while}\ b\ \mathtt{do}\ c]\!]\sigma := \Big(\,[\![\,(\mathtt{while}\ b\ \mathtt{do}\ c)|_i\,]\!](\sigma|_i)\,\Big)_{i \in \mathbb{N}}$$

Sectionwise definition

Hasuo (Tokyo)

Denotational Semantics

$$
\begin{aligned}
[\![x]\!]\sigma &:= \sigma(x) \\
[\![a_1 \ \mathtt{aop}\ a_2]\!]\sigma &:= [\![a_1]\!]\sigma \ \mathtt{aop}\ [\![a_2]\!]\sigma \\
[\![\mathtt{dt}]\!]\sigma &:= \omega^{-1} = \left[\, (1, \tfrac{1}{2}, \tfrac{1}{3}, \dots)\, \right]
\end{aligned}
$$

$$
\begin{aligned}
[\![\mathtt{true}]\!]\sigma &:= \mathrm{tt} \\
[\![b_1 \wedge b_2]\!]\sigma &:= [\![b_1]\!]\sigma \wedge [\![b_2]\!]\sigma \\
[\![a_1 < a_2]\!]\sigma &:= [\![a_1]\!]\sigma < [\![a_2]\!]\sigma
\end{aligned}
$$

$[\![\mathtt{fal}$

$[\![\neg b$

Def.

The ***i*-*th section*** of a $\mathrm{W{\scriptsize HILE}}^{\mathrm{dt}}$ expression e is

$$
e|_i \quad :\equiv \quad e\left[\, \frac{1}{i+1} \,/\, \mathtt{dt} \,\right] .
$$

$$
[\![\mathtt{skip}]\!]\sigma := \sigma \qquad [\![x := a]\!]\sigma := \sigma\left[\, x \mapsto [\![a]\!]\sigma \,\right] \qquad [\![c_1; c_2]\!]\sigma := [\![c_2]\!]\left([\![c_1]\!]\sigma \right)
$$

$$
[\![\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2]\!]\sigma := \begin{cases} [\![c_1]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathrm{tt} \\ [\![c_2]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathrm{ff} \end{cases}
$$

$$
[\![\mathtt{while}\ b\ \mathtt{do}\ c]\!]\sigma := \Big([\![\, (\mathtt{while}\ b\ \mathtt{do}\ c)|_i \,]\!](\sigma|_i) \Big)_{i \in \mathbb{N}}
$$

Sectionwise definition

Denota

$$\left[\begin{array}{l} t := 0\,; \\ \texttt{while}\ (t \le 1)\ \texttt{do} \\ \quad t := t + \mathtt{dt} \end{array}\right] \overset{i\text{-th section}}{\longmapsto} \left[\begin{array}{l} t := 0\,; \\ \texttt{while}\ (t \le 1)\ \texttt{do} \\ \quad t := t + \frac{1}{i+1} \end{array}\right]$$

Def.
The *i-th section* of a WHILE$^{\mathrm{dt}}$ expression e is

$$e|_i \;:\equiv\; e\left[\,\frac{1}{i+1}\,/\,\mathtt{dt}\,\right]\,.$$

$$[\![x]\!]\sigma \;:=\; \sigma(x)$$
$$[\![a_1\ \mathtt{aop}\ a_2]\!]\sigma \;:=\; [\![a_1]\!]\sigma\ \mathtt{aop}\ [\![a_2]\!]\sigma$$
$$[\![\mathtt{dt}]\!]\sigma \;:=\; \omega^{-1} = \left[\,(1, \tfrac{1}{2}, \tfrac{1}{3}, \dots)\,\right]$$

$$[\![\mathtt{true}]\!]\sigma \;:=\; \mathrm{tt} \qquad [\![\mathtt{fal}$$
$$[\![b_1 \wedge b_2]\!]\sigma \;:=\; [\![b_1]\!]\sigma \wedge [\![b_2]\!]\sigma \qquad [\![\neg b$$
$$[\![a_1 < a_2]\!]\sigma \;:=\; [\![a_1]\!]\sigma < [\![a_2]\!]\sigma$$

$$[\![\mathtt{skip}]\!]\sigma \;:=\; \sigma \qquad [\![x := a]\!]\sigma \;:=\; \sigma\left[\,x \mapsto [\![a]\!]\sigma\,\right] \qquad [\![c_1; c_2]\!]\sigma \;:=\; [\![c_2]\!]\left([\![c_1]\!]\sigma\right)$$

$$[\![\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2]\!]\sigma := \begin{cases} [\![c_1]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathrm{tt} \\ [\![c_2]\!]\sigma & \text{if } [\![b]\!]\sigma = \mathrm{ff} \end{cases}$$

$$[\![\mathtt{while}\ b\ \mathtt{do}\ c]\!]\sigma := \Big(\, [\![\, (\mathtt{while}\ b\ \mathtt{do}\ c)|_i \,]\!](\sigma|_i) \,\Big)_{i \in \mathbb{N}}$$

Sectionwise definition

Hasuo (Tokyo)

“Sectionwise Lemmas”

Sectionwise Execution Lemma. For any expr. $\boldsymbol{e}$ and $\boldsymbol{i} \in \mathbb{N}$,

$$[\![\boldsymbol{e}]\!]\boldsymbol{\sigma} = \left[\,\left(\,[\![\boldsymbol{e}|_{\boldsymbol{i}}]\!](\boldsymbol{\sigma}|_{\boldsymbol{i}})\,\right)_{\boldsymbol{i}\in\mathbb{N}}\,\right] \ .$$

Sectionwise Satisfaction Lemma. For any hyperstate $\boldsymbol{\sigma}$ and an $\textsc{Assn}^{\mathrm{dt}}$ formula $\boldsymbol{\varphi}$:

$$\boldsymbol{\sigma} \models \boldsymbol{\varphi} \quad \Longleftrightarrow \quad \boldsymbol{\sigma}|_{\boldsymbol{i}} \models \boldsymbol{\varphi}|_{\boldsymbol{i}} \quad \text{for almost every } \boldsymbol{i}.$$

“Sectionwise Lemmas”

Sectionwise Execution Lemma. For any expr. $\boldsymbol{e}$ and $\boldsymbol{i} \in \mathbb{N}$,

$$[\![e]\!] \boldsymbol{\sigma} = \Big[\Big([\![e|_i]\!] (\boldsymbol{\sigma}|_i) \Big)_{\boldsymbol{i} \in \mathbb{N}} \Big] \ .$$

“Whole semantics is sectionwise”

Sectionwise Satisfaction Lemma. For any hyperstate $\boldsymbol{\sigma}$ and an $\mathrm{Assn}^{\mathrm{dt}}$ formula $\boldsymbol{\varphi}$:

$$\boldsymbol{\sigma} \models \boldsymbol{\varphi} \quad \Longleftrightarrow \quad \boldsymbol{\sigma}|_i \models \boldsymbol{\varphi}|_i \quad \text{for almost every } \boldsymbol{i}.$$

Hasuo (Tokyo)

"Sectionwise Lemmas"

Sectionwise Execution Lemma. For any expr. e and $i \in \mathbb{N}$,

$$[\![e]\!] \sigma = \left[\left([\![e|_i]\!] (\sigma|_i) \right)_{i \in \mathbb{N}} \right] .$$

"Whole semantics is sectionwise"

Sectionwise Satisfaction Lemma. For any hyperstate σ and an $\textsc{Assn}^{\mathrm{dt}}$ formula φ:

$$\sigma \models \varphi \quad \iff \quad \sigma|_i \models \varphi|_i \quad \text{for almost every } i.$$

"Łos' Theorem"

"Sectionwise Lemmas"

Lem. (Sectionwise validity of Hoare triples)

$$\models \{A\}c\{B\} \quad \iff$$

$$\models \{A|_i\}\, c|_i\, \{B|_i\} \text{ for almost every } i.$$

"Sectionwise Lemmas"

Lem. (Sectionwise validity of Hoare triples)

$$\models \{A\}c\{B\} \iff \models \{A|_i\}\, c|_i\, \{B|_i\} \text{ for almost every } i.$$

Interface for **transferring** static analysis techniques

Q. Is a $\mathbf{While}^{\mathrm{dt}}$ program executable?

* A. Not exactly.
  * A **modeling** language
    * Not numerical approx., but **exact** modeling

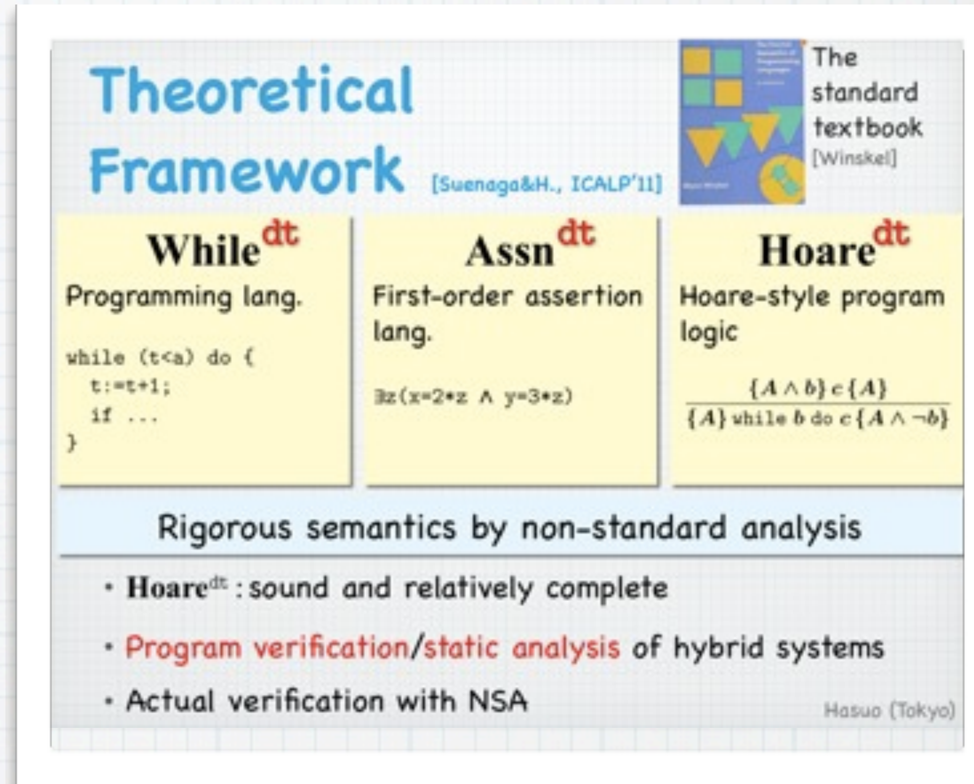
Q. Is a $\mathbf{While}^{\mathrm{dt}}$ program executable?

* A. Not exactly.
  * A **modeling** language
    * Not numerical approx., but **exact** modeling

Q. Is a $\mathbf{While}^{\mathrm{dt}}$ program executable?

- A. Not exactly.
  - A **modeling** language
    - Not numerical approx., but **exact** modeling
  - Static analysis ➜ **no need to execute!**
    - Mathematical semantics suffices

Outline



Suenaga & H., ICALP'11

- **Theoretical foundations**
  - **While^{dt}, Assn^{dt}, Hoare^{dt}**
  - Rigorous semantics via NSA
  - Transfer principle, "sectionwise lemmas"

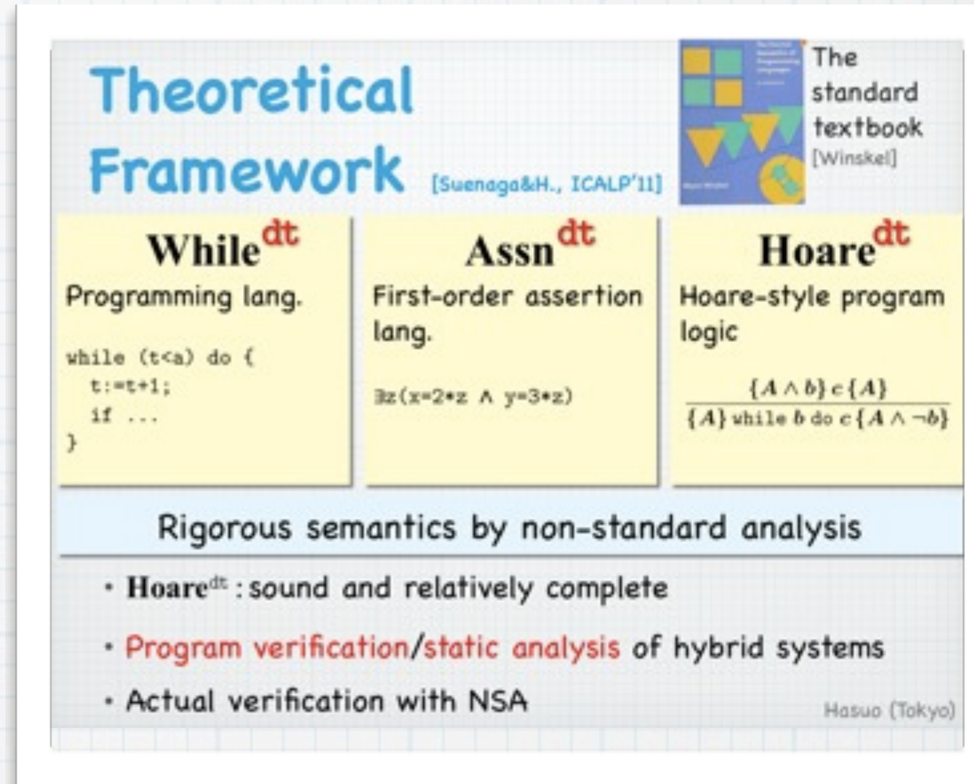
H. & Suenaga, CAV'12

- **Static analysis techniques, transferred as they are**
  - Phase split [Sharma,Dillig,Dillig,Aiken; CAV'11] [Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]
  - Differential invariant [Platzer,Clarke; CAV'08]
  - ... and more!

Outline

Suenaga & H., ICALP'11

* **Theoretical foundations**
  * **While**^{dt}, **Assn**^{dt}, **Hoare**^{dt}
  * Rigorous semantics via NSA
  * Transfer principle, "sectionwise lemmas"



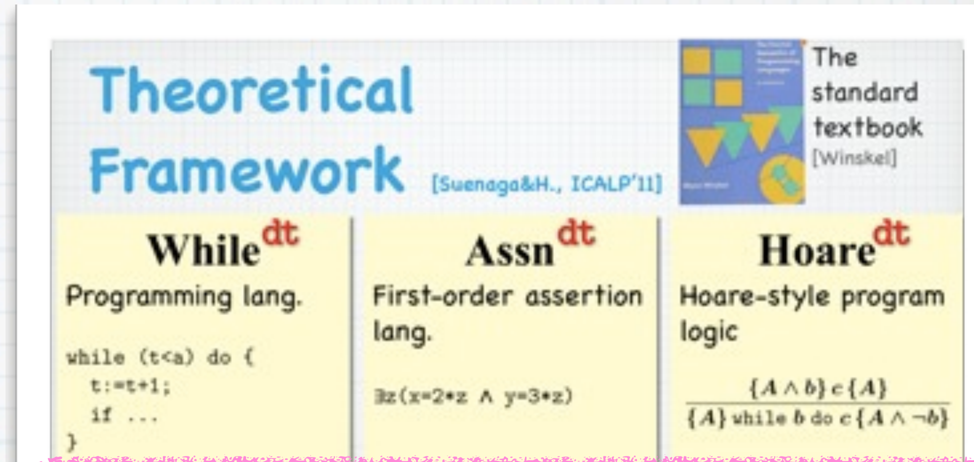
Done

H. & Suenaga, CAV'12

* **Static analysis techniques, transferred as they are**
  * Phase split [Sharma,Dillig,Dillig,Aiken; CAV'11] [Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]
  * Differential invariant [Platzer,Clarke; CAV'08]
  * ... and more!

Outline

Suenaga & H., ICALP'11



- **Theoretical foundations**
  - **While**^{dt}**, Assn**^{dt}**, Hoare**^{dt}
  - Rigorous semantics via NSA
  - Transfer principle, "sectionwise lemmas"

w/ or w/o dt ...

➜ logically "**the same**"

Done

H. & Suenaga, CAV'12

- **Static analysis techniques, transferred as they are**
  - Phase split [Sharma,Dillig,Dillig,Aiken; CAV'11] [Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]
  - Differential invariant [Platzer,Clarke; CAV'08]
  - ... and more!

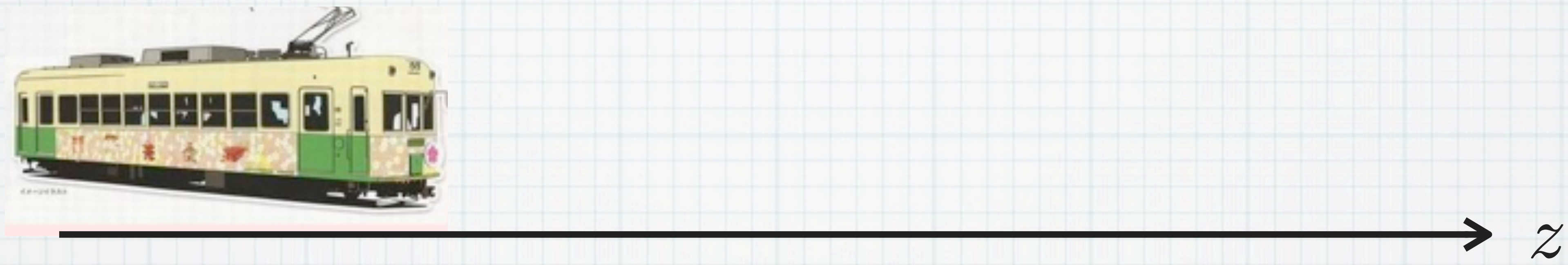
Part II:

Exercises in Nonstandard Static Analysis

Exercise 1.1

$$
\begin{array}{l}
\texttt{while}\ t < \varepsilon\ \texttt{do}\ \{ \\
\qquad t := t + \mathrm{dt}\,; \\
\qquad v := v + a \cdot \mathrm{dt}\,; \\
\qquad z := z + v \cdot \mathrm{dt} \\
\}
\end{array}
$$

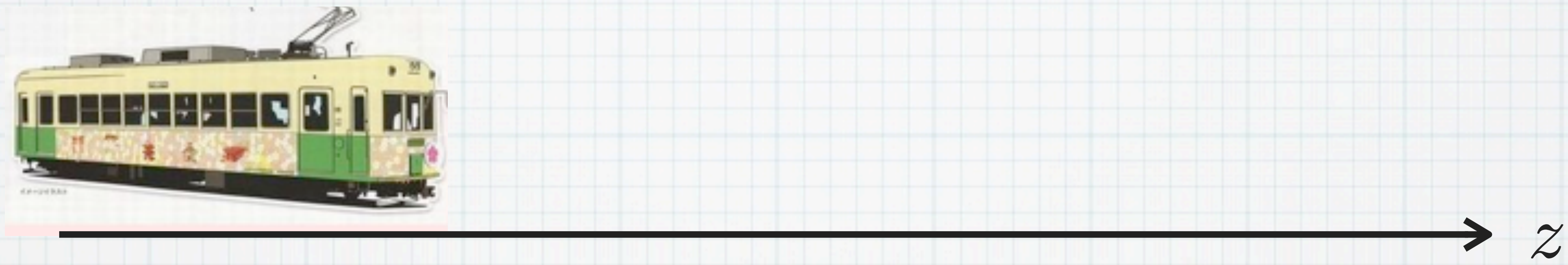
Exercise 1.1




```
while t < ε do {
    t := t + dt ;
    v := v + a · dt ;
    z := z + v · dt
}
```

```
while v > 0 do {
    t := 0 ;
    if m − z < s then a := −b else a := a₀ ;
    while t < ε do {
        t := t + dt ;
        v := v + a · dt ;
        z := z + v · dt
    }}
```

Exercise 1.1



$\texttt{while } v > 0 \texttt{ do } \{$

$\quad t := 0\,;$

$\quad \texttt{if } m - z < s \texttt{ then } a := -b \texttt{ else } a := a_0\,;$

$\quad \texttt{while } t < \varepsilon \texttt{ do } \{$

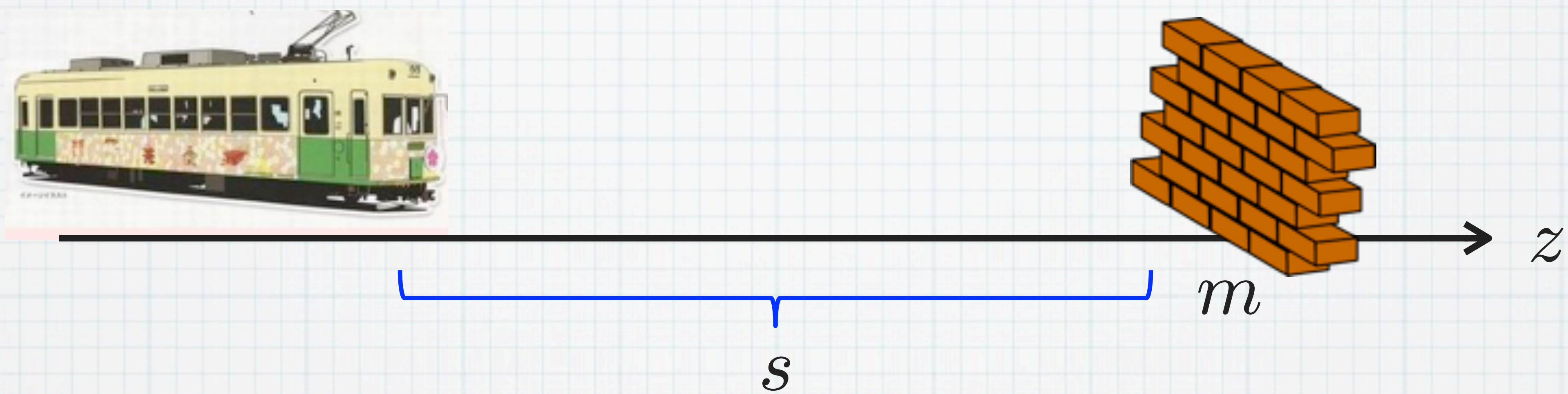
$\qquad t := t + \mathrm{dt}\,;$

$\qquad v := v + a \cdot \mathrm{dt}\,;$

$\qquad z := z + v \cdot \mathrm{dt}$

$\quad \}\}$

Exercise 1.1



$\texttt{while}\ v > 0\ \texttt{do}\ \{$

$\quad t := 0\,;$

$\quad \texttt{if}\ m - z < s\ \texttt{then}\ a := -b\ \texttt{else}\ a := a_0\,;$

$\quad \texttt{while}\ t < \varepsilon\ \texttt{do}\ \{$

$\qquad t := t + \mathrm{dt}\,;$

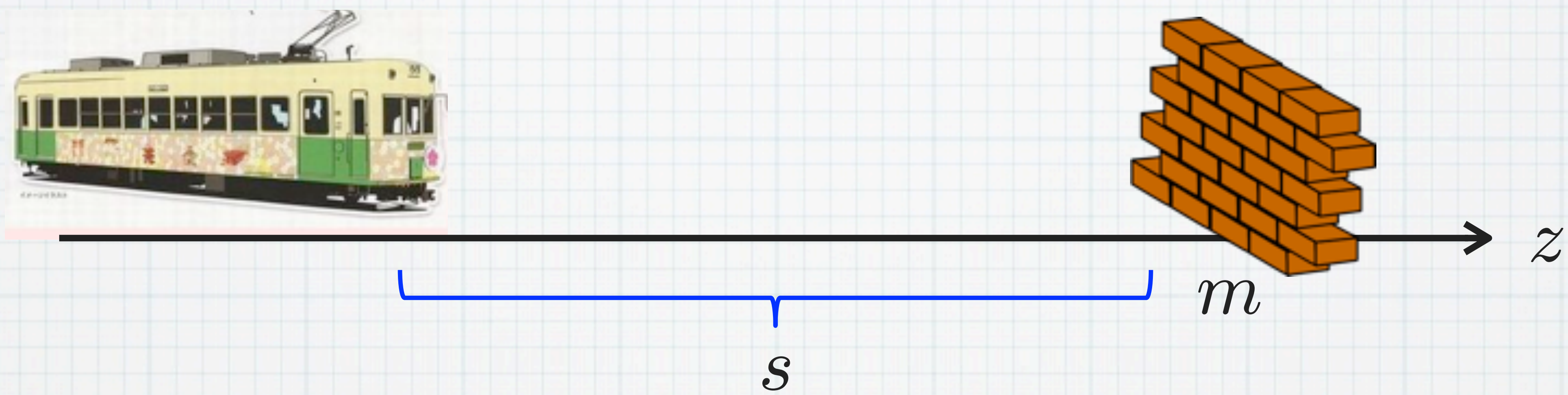
$\qquad v := v + a \cdot \mathrm{dt}\,;$

$\qquad z := z + v \cdot \mathrm{dt}$

$\quad \}\}$

Exercise 1.1

(Tiny) fragment of
Euro. Train Ctrl. Sys. (ETCS)

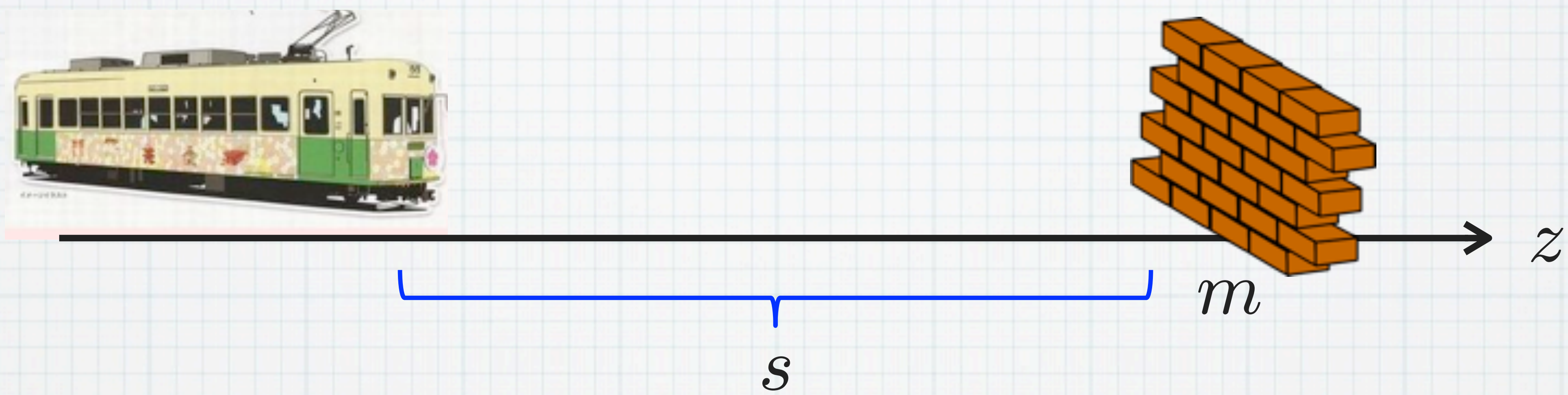



```
while v > 0 do {
    t := 0 ;
    if m − z < s then a := −b else a := a₀ ;
    while t < ε do {
        t := t + dt ;
        v := v + a · dt ;
        z := z + v · dt
}}
```

ETCS_0

Exercise 1.1

(Tiny) fragment of
Euro. Train Ctrl. Sys. (ETCS)



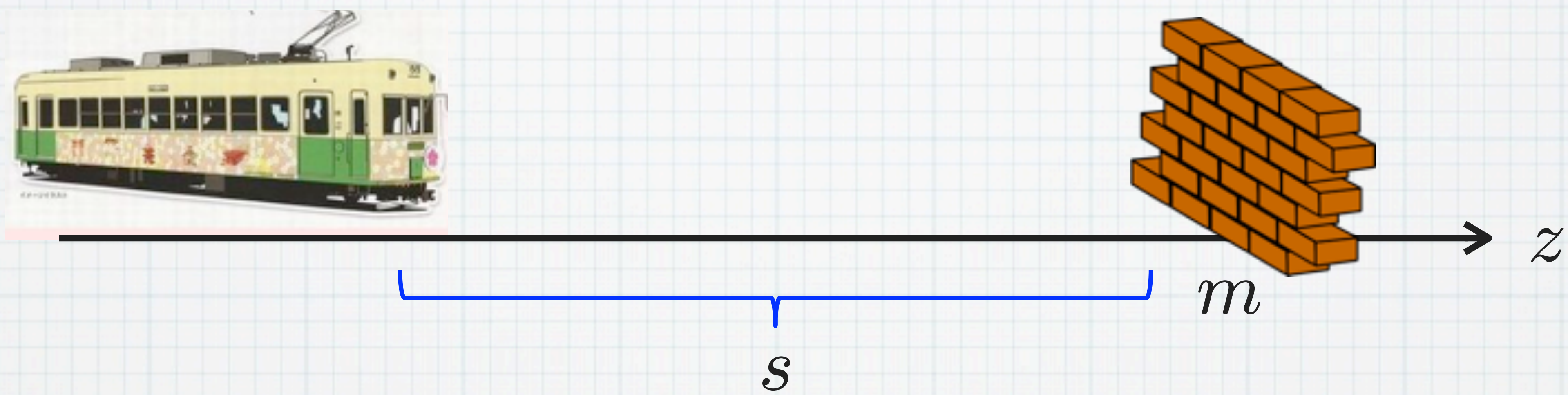
$$
\begin{array}{l}
\texttt{while } v > 0 \texttt{ do } \{ \\
\quad t := 0\,; \\
\quad \texttt{if } m - z < s \texttt{ then } a := -b \texttt{ else } a := a_0\,; \\
\quad \texttt{while } t < \varepsilon \texttt{ do } \{ \\
\qquad t := t + \mathrm{dt}\,; \\
\qquad v := v + a \cdot \mathrm{dt}\,; \\
\qquad z := z + v \cdot \mathrm{dt} \\
\quad \}\}
\end{array}
$$

ETCS_0

Q. Find A s.t. $\models \{A\}\,\mathrm{ETCS}_0\,\{z < m\}$

Exercise 1.1

(Tiny) fragment of
Euro. Train Ctrl. Sys. (ETCS)




```
while v > 0 do {
    t := 0 ;
    if m − z < s then a := −b else a := a₀ ;
    while t < ε do {
        t := t + dt ;
        v := v + a · dt ;
        z := z + v · dt
}}
```

ETCS_0

s: big enough
b: big enough
a_0: small enough
…

Q. Find A s.t. $\models \{A\}\,\mathrm{ETCS}_0\,\{z < m\}$

```
while (v > 0) {
  if m - z < s
    then a := -b
    else a := a0;
  t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0) {
  if m - z < s
    then a := -b
    else a := a0;
  t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0) {
  if m - z < s
    then a := -b
    else a := a0;
  t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0 && m - z >= s) {
  a := a0;    t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;    t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0) {
  if m - z < s
    then a := -b
    else a := a0;
  t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0 && m - z >= s) {
  a := a0;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

accel.

brake

```
while (v > 0) {
  if m - z < s
    then a := -b
    else a := a0;
  t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0 && m - z >= s) {
  a := a0;    t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;    t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

accel.

brake

Startegy 1 **"Phase split"**

[Sharma,Dillig,Dillig,Aiken; CAV'11]
[Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09] [Gopan,Reps; SAS'07]

Phase Split

(Standard Ver., for While & Hoare)

[Sharma,Dillig,Dillig,Aiken; CAV'11]

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \;::=\; \mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid h; c \mid c; h \mid \mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c \mid \mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2} :\equiv \mathtt{skip}$$
$$\overline{h; c} :\equiv \overline{h} \qquad \overline{c; h} :\equiv c; \overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c} :\equiv \mathtt{assert}\ b\,; \overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h} :\equiv \mathtt{assert}\ \neg b\,; \overline{h}$$

Lem.
If a Boolean expression $b_s \in \mathbf{BExp}$ satisfies

$$\models \{b_s\}\,\overline{h}\,\{b_c\}\;,\quad \models \{\neg b_s\}\,\overline{h}\,\{\neg b_c\}\;,\quad \text{and} \quad \models \{b_g \wedge b_s\}\,h[b_c]\,\{\neg b_g \vee b_s\}\;,$$

then we have

$$[\![\, \mathtt{while}\ b_g\ \mathtt{do}\ h[b_c] \,]\!] = \left[\!\!\left[\, \begin{array}{l} \mathtt{while}\ (b_g \wedge \neg b_s)\ \mathtt{do}\ h[\mathtt{false}]\,; \\ \mathtt{while}\ (b_g \wedge b_s)\ \mathtt{do}\ h[\mathtt{true}] \end{array} \,\right]\!\!\right] .$$

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \;::=\; \texttt{if}\;[\_]\;\texttt{then}\;c_1\;\texttt{else}\;c_2 \mid h;c \mid c;h \mid \texttt{if}\;b\;\texttt{then}\;h\;\texttt{else}\;c \mid \texttt{if}\;b\;\texttt{then}\;c\;\texttt{else}\;h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\texttt{if}\;[\_]\;\texttt{then}\;c_1\;\texttt{else}\;c_2} :\equiv \texttt{skip}$$
$$\overline{h;c} :\equiv \overline{h} \qquad \overline{c;h} :\equiv c;\overline{h}$$
$$\overline{\texttt{if}\;b\;\texttt{then}\;h\;\texttt{else}\;c} :\equiv \texttt{assert}\;b\;;\overline{h}$$
$$\overline{\texttt{if}\;b\;\texttt{then}\;c\;\texttt{else}\;h} :\equiv \texttt{assert}\;\neg b\;;\overline{h}$$

Phase Split

(Standard Ver., for While & Hoare)

[Sharma,Dillig,Dillig,Aiken; CAV'11]

$\texttt{while}\;b_{\mathrm{g}}\;\texttt{do}\;\boxed{\ldots(\texttt{if}\ldots)\ldots}$

into $\left[\begin{array}{l}\texttt{while}\;b_{\mathrm{g}}\wedge\neg b_{\mathrm{s}}\;\texttt{do}\;\boxed{\ldots}\;;\\ \texttt{while}\;b_{\mathrm{g}}\wedge b_{\mathrm{s}}\;\texttt{do}\;\boxed{\ldots}\end{array}\right]$

Lem.
If a Boolean expression $b_{\mathrm{s}} \in \mathbf{BExp}$ satisfies

$$\models \{b_{\mathrm{s}}\}\,\overline{h}\,\{b_{\mathrm{c}}\}\;,\quad \models \{\neg b_{\mathrm{s}}\}\,\overline{h}\,\{\neg b_{\mathrm{c}}\}\;,\quad\text{and}\quad \models \{b_{\mathrm{g}}\wedge b_{\mathrm{s}}\}\,h[b_{\mathrm{c}}]\,\{\neg b_{\mathrm{g}}\vee b_{\mathrm{s}}\}\;,$$

then we have

$$[\![\,\texttt{while}\;b_{\mathrm{g}}\;\texttt{do}\;h[b_{\mathrm{c}}]\,]\!] = \left[\!\!\left[\begin{array}{l}\texttt{while}\;(b_{\mathrm{g}}\wedge\neg b_{\mathrm{s}})\;\texttt{do}\;h[\texttt{false}]\;;\\ \texttt{while}\;(b_{\mathrm{g}}\wedge b_{\mathrm{s}})\;\texttt{do}\;h[\texttt{true}]\end{array}\right]\!\!\right].$$

Phase Split

(Nonstandard Ver., for $\mathbf{While}^{\mathrm{dt}}$ & $\mathbf{Hoare}^{\mathrm{dt}}$)

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \ ::= \ \mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid h; c \mid c; h \mid \mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c \mid \mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2} :\equiv \mathtt{skip}$$
$$\overline{h; c} :\equiv \overline{h} \qquad \overline{c; h} :\equiv c; \overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c} :\equiv \mathtt{assert}\ b\ ; \overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h} :\equiv \mathtt{assert}\ \neg b\ ; \overline{h}$$

Lem.
If a Boolean expression $b_{\mathrm{s}} \in \mathbf{BExp}$ satisfies

$$\models \{b_{\mathrm{s}}\}\, \overline{h}\, \{b_{\mathrm{c}}\} \ , \quad \models \{\neg b_{\mathrm{s}}\}\, \overline{h}\, \{\neg b_{\mathrm{c}}\} \ , \quad \text{and} \quad \models \{b_{\mathrm{g}} \wedge b_{\mathrm{s}}\}\, h[b_{\mathrm{c}}]\, \{\neg b_{\mathrm{g}} \vee b_{\mathrm{s}}\} \ ,$$

then we have

$$[\![\, \mathtt{while}\ b_{\mathrm{g}}\ \mathtt{do}\ h[b_{\mathrm{c}}] \,]\!] = \left[\!\!\left[\begin{array}{l} \mathtt{while}\ (b_{\mathrm{g}} \wedge \neg b_{\mathrm{s}})\ \mathtt{do}\ h[\mathtt{false}]\ ; \\ \mathtt{while}\ (b_{\mathrm{g}} \wedge b_{\mathrm{s}})\ \mathtt{do}\ h[\mathtt{true}] \end{array}\right]\!\!\right] .$$

Hasuo (Tokyo)

Phase Split
(Nonstandard Ver., for $\mathbf{While}^{\mathrm{dt}}$ & $\mathbf{Hoare}^{\mathrm{dt}}$)

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \ ::= \ \texttt{if}\ [\_]\ \texttt{then}\ c_1\ \texttt{else}\ c_2 \mid h; c \mid c; h \mid \texttt{if}\ b\ \texttt{then}\ h\ \texttt{else}\ c \mid \texttt{if}\ b\ \texttt{then}\ c\ \texttt{else}\ h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\texttt{if}\ [\_]\ \texttt{then}\ c_1\ \texttt{else}\ c_2} :\equiv \texttt{skip}$$
$$\overline{h; c} :\equiv \overline{h} \qquad \overline{c; h} :\equiv c; \overline{h}$$
$$\overline{\texttt{if}\ b\ \texttt{then}\ h\ \texttt{else}\ c} :\equiv \texttt{assert}\ b\ ; \overline{h}$$
$$\overline{\texttt{if}\ b\ \texttt{then}\ c\ \texttt{else}\ h} :\equiv \texttt{assert}\ \neg b\ ; \overline{h}$$

Lem.
If a Boolean expression $b_{\mathrm{s}} \in \mathbf{BExp}$ satisfies

$$\models \{b_{\mathrm{s}}\}\ \overline{h}\ \{b_{\mathrm{c}}\}\ , \quad \models \{\neg b_{\mathrm{s}}\}\ \overline{h}\ \{\neg b_{\mathrm{c}}\}\ , \quad \text{and} \quad \models \{b_{\mathrm{g}} \wedge b_{\mathrm{s}}\}\ h[b_{\mathrm{c}}]\ \{\neg b_{\mathrm{g}} \vee b_{\mathrm{s}}\}\ ,$$

then we have

$$[\![\ \texttt{while}\ b_{\mathrm{g}}\ \texttt{do}\ h[b_{\mathrm{c}}]\ ]\!] = \left[\!\!\left[\ \begin{array}{l} \texttt{while}\ (b_{\mathrm{g}} \wedge \neg b_{\mathrm{s}})\ \texttt{do}\ h[\texttt{false}]\ ; \\ \texttt{while}\ (b_{\mathrm{g}} \wedge b_{\mathrm{s}})\ \texttt{do}\ h[\texttt{true}] \end{array}\ \right]\!\!\right] .$$

Phase Split

(Nonstandard Ver., for $\mathbf{While}^{\mathrm{dt}}$ & $\mathbf{Hoare}^{\mathrm{dt}}$)

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \ ::= \ \mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid h; c \mid c; h \mid \mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c \mid \mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2} :\equiv \mathtt{skip}$$
$$\overline{h; c} :\equiv \overline{h} \qquad \overline{c; h} :\equiv c; \overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c} :\equiv \mathtt{assert}\ b\ ; \overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h} :\equiv \mathtt{assert}\ \neg b\ ; \overline{h}$$

Lem.
If a Boolean expression $b_{\mathrm{s}} \in \mathbf{BExp}$ satisfies

$$\models \{b_{\mathrm{s}}\}\,\overline{h}\,\{b_{\mathrm{c}}\}\ , \quad \models \{\neg b_{\mathrm{s}}\}\,\overline{h}\,\{\neg b_{\mathrm{c}}\}\ , \quad \text{and} \quad \models \{b_{\mathrm{g}} \wedge b_{\mathrm{s}}\}\, h[b_{\mathrm{c}}]\, \{\neg b_{\mathrm{g}} \vee b_{\mathrm{s}}\}\ ,$$

then we have

$$[\![\ \mathtt{while}\ b_{\mathrm{g}}\ \mathtt{do}\ h[b_{\mathrm{c}}]\ ]\!] = \left[\!\!\left[\ \begin{array}{l} \mathtt{while}\ (b_{\mathrm{g}} \wedge \neg b_{\mathrm{s}})\ \mathtt{do}\ h[\mathtt{false}]\ ; \\ \mathtt{while}\ (b_{\mathrm{g}} \wedge b_{\mathrm{s}})\ \mathtt{do}\ h[\mathtt{true}] \end{array}\ \right]\!\!\right] .$$

Proof.

$$\models \{b_{\mathrm{s}}\}\,\overline{h}\,\{b_{\mathrm{c}}\}$$
$$\models \{\neg b_{\mathrm{s}}\}\,\overline{h}\,\{\neg b_{\mathrm{c}}\}$$
$$\models \{b_{\mathrm{g}} \wedge b_{\mathrm{s}}\}\, h[b_{\mathrm{c}}]\ \{\neg b_{\mathrm{g}} \vee b_{\mathrm{s}}\}$$

Hasuo (Tokyo)

Phase Split

(Nonstandard Ver., for While^{dt} & Hoare^{dt})

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \ ::= \ \texttt{if}\ [\_]\ \texttt{then}\ c_1\ \texttt{else}\ c_2 \mid h; c \mid c; h \mid \texttt{if}\ b\ \texttt{then}\ h\ \texttt{else}\ c \mid \texttt{if}\ b\ \texttt{then}\ c\ \texttt{else}\ h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\texttt{if}\ [\_]\ \texttt{then}\ c_1\ \texttt{else}\ c_2} :\equiv \texttt{skip}$$
$$\overline{h; c} :\equiv \overline{h} \qquad \overline{c; h} :\equiv c; \overline{h}$$
$$\overline{\texttt{if}\ b\ \texttt{then}\ h\ \texttt{else}\ c} :\equiv \texttt{assert}\ b\ ; \overline{h}$$
$$\overline{\texttt{if}\ b\ \texttt{then}\ c\ \texttt{else}\ h} :\equiv \texttt{assert}\ \neg b\ ; \overline{h}$$

Lem.
If a Boolean expression $b_s \in \mathbf{BExp}$ satisfies

$$\models \{b_s\}\, \overline{h}\, \{b_c\}\ , \quad \models \{\neg b_s\}\, \overline{h}\, \{\neg b_c\}\ , \quad \text{and} \quad \models \{b_g \wedge b_s\}\, h[b_c]\, \{\neg b_g \vee b_s\}\ ,$$

then we have

$$[\![\ \texttt{while}\ b_g\ \texttt{do}\ h[b_c]\ ]\!] = \left[\!\!\left[\ \begin{array}{l} \texttt{while}\ (b_g \wedge \neg b_s)\ \texttt{do}\ h[\texttt{false}]\ ; \\ \texttt{while}\ (b_g \wedge b_s)\ \texttt{do}\ h[\texttt{true}] \end{array}\ \right]\!\!\right] .$$

Proof.

$$\models \{b_s\}\, \overline{h}\, \{b_c\}$$
$$\models \{\neg b_s\}\, \overline{h}\, \{\neg b_c\}$$
$$\models \{b_g \wedge b_s\}\, h[b_c]\, \{\neg b_g \vee b_s\}$$

$\Leftrightarrow$ sectionwise

$$\models \{b_s|_i\}\, \overline{h}|_i\, \{b_c|_i\}$$
$$\models \{\neg b_s|_i\}\, \overline{h}|_i\, \{\neg b_c|_i\}$$
$$\models \{b_g|_i \wedge b_s|_i\}\, h|_i[b_c|_i]\, \{\neg b_g|_i \vee b_s|_i\}$$

⋮

(for almost all i)

Hasuo (Tokyo)

Phase Split

(Nonstandard Ver., for $\mathbf{While}^{\mathrm{dt}}$ & $\mathbf{Hoare}^{\mathrm{dt}}$)

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \ ::= \ \mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid h;c \mid c;h \mid \mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c \mid \mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\mathtt{if}\ [\_]\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2} :\equiv \mathtt{skip}$$
$$\overline{h;c} :\equiv \overline{h} \qquad \overline{c;h} :\equiv c;\overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ h\ \mathtt{else}\ c} :\equiv \mathtt{assert}\ b\ ;\overline{h}$$
$$\overline{\mathtt{if}\ b\ \mathtt{then}\ c\ \mathtt{else}\ h} :\equiv \mathtt{assert}\ \neg b\ ;\overline{h}$$

Lem.
If a Boolean expression $b_{\mathrm{s}} \in \mathbf{BExp}$ satisfies

$$\models \{b_{\mathrm{s}}\}\,\overline{h}\,\{b_{\mathrm{c}}\}\ , \quad \models \{\neg b_{\mathrm{s}}\}\,\overline{h}\,\{\neg b_{\mathrm{c}}\}\ , \quad \text{and} \quad \models \{b_{\mathrm{g}} \wedge b_{\mathrm{s}}\}\,h[b_{\mathrm{c}}]\,\{\neg b_{\mathrm{g}} \vee b_{\mathrm{s}}\}\ ,$$

then we have

$$[\![\, \mathtt{while}\ b_{\mathrm{g}}\ \mathtt{do}\ h[b_{\mathrm{c}}] \,]\!] = [\![\, \begin{array}{l} \mathtt{while}\ (b_{\mathrm{g}} \wedge \neg b_{\mathrm{s}})\ \mathtt{do}\ h[\mathtt{false}]\ ; \\ \mathtt{while}\ (b_{\mathrm{g}} \wedge b_{\mathrm{s}})\ \mathtt{do}\ h[\mathtt{true}] \end{array} \,]\!]\ .$$

Proof.

$$\models \{b_{\mathrm{s}}\}\,\overline{h}\,\{b_{\mathrm{c}}\}$$
$$\models \{\neg b_{\mathrm{s}}\}\,\overline{h}\,\{\neg b_{\mathrm{c}}\}$$
$$\models \{b_{\mathrm{g}} \wedge b_{\mathrm{s}}\}\,h[b_{\mathrm{c}}]\,\{\neg b_{\mathrm{g}} \vee b_{\mathrm{s}}\}$$

$\Leftrightarrow$ sectionwise

$$\models \{b_{\mathrm{s}}|_i\}\,\overline{h|_i}\,\{b_{\mathrm{c}}|_i\}$$
$$\models \{\neg b_{\mathrm{s}}|_i\}\,\overline{h|_i}\,\{\neg b_{\mathrm{c}}|_i\}$$
$$\models \{b_{\mathrm{g}}|_i \wedge b_{\mathrm{s}}|_i\}\,h|_i[b_{\mathrm{c}}|_i]\,\{\neg b_{\mathrm{g}}|_i \vee b_{\mathrm{s}}|_i\}$$

⋮

(for almost all i)

$\Rightarrow$ std. ver.

$$[\![\, \mathtt{while}\ b_{\mathrm{g}}|_i\ \mathtt{do}\ h|_i[b_{\mathrm{c}}|_i] \,]\!] = [\![\, \begin{array}{l} \mathtt{while}\ (b_{\mathrm{g}}|_i \wedge \neg b_{\mathrm{s}}|_i)\ \mathtt{do}\ h|_i[\mathtt{false}]\ ; \\ \mathtt{while}\ (b_{\mathrm{g}}|_i \wedge b_{\mathrm{s}}|_i)\ \mathtt{do}\ h|_i[\mathtt{true}] \end{array} \,]\!]$$

⋮

Phase Split
(Nonstandard Ver., for While^{dt} & Hoare^{dt})

Defn.
The set of *holed commands* $\mathbf{Cmd}_{[\_]}$ is:

$$\mathbf{Cmd}_{[\_]} \ni h \ ::= \ \texttt{if } [\_] \texttt{ then } c_1 \texttt{ else } c_2 \mid h; c \mid c; h \mid \texttt{if } b \texttt{ then } h \texttt{ else } c \mid \texttt{if } b \texttt{ then } c \texttt{ else } h$$

For each holed command h, its *pre-hole fragment* $\overline{h}$ is:

$$\overline{\texttt{if } [\_] \texttt{ then } c_1 \texttt{ else } c_2} :\equiv \texttt{skip}$$
$$\overline{h; c} :\equiv \overline{h} \qquad \overline{c; h} :\equiv c; \overline{h}$$
$$\overline{\texttt{if } b \texttt{ then } h \texttt{ else } c} :\equiv \texttt{assert } b \,; \overline{h}$$
$$\overline{\texttt{if } b \texttt{ then } c \texttt{ else } h} :\equiv \texttt{assert } \neg b \,; \overline{h}$$

Lem.
If a Boolean expression $b_s \in \mathbf{BExp}$ satisfies

$$\models \{b_s\}\, \overline{h}\, \{b_c\} \ , \quad \models \{\neg b_s\}\, \overline{h}\, \{\neg b_c\} \ , \quad \text{and} \quad \models \{b_g \wedge b_s\}\, h[b_c]\, \{\neg b_g \vee b_s\} \ ,$$

then we have

$$[\![\texttt{while } b_g \texttt{ do } h[b_c]]\!] = \left[\!\!\left[\begin{array}{l} \texttt{while } (b_g \wedge \neg b_s) \texttt{ do } h[\texttt{false}] \,; \\ \texttt{while } (b_g \wedge b_s) \texttt{ do } h[\texttt{true}] \end{array} \right]\!\!\right] .$$

Proof.

$\models \{b_s\}\, \overline{h}\, \{b_c\}$
$\models \{\neg b_s\}\, \overline{h}\, \{\neg b_c\}$
$\models \{b_g \wedge b_s\}\, h[b_c]\, \{\neg b_g \vee b_s\}$

$\Leftrightarrow$ sectionwise

$\models \{b_s|_i\}\, \overline{h|_i}\, \{b_c|_i\}$
$\models \{\neg b_s|_i\}\, \overline{h|_i}\, \{\neg b_c|_i\}$
$\models \{b_g|_i \wedge b_s|_i\}\, h|_i[b_c|_i]\, \{\neg b_g|_i \vee b_s|_i\}$

⋮

(for almost all i)

$\Rrightarrow$ ... std. ver.

$$[\![\texttt{while } b_g|_i \texttt{ do } h|_i[b_c|_i]]\!] = \left[\!\!\left[\begin{array}{l} \texttt{while } (b_g|_i \wedge \neg b_s|_i) \texttt{ do } h|_i[\texttt{false}] \,; \\ \texttt{while } (b_g|_i \wedge b_s|_i) \texttt{ do } h|_i[\texttt{true}] \end{array} \right]\!\!\right]$$

⋮

sectionwise $\Updownarrow$

$$[\![\texttt{while } b_g \texttt{ do } h[b_c]]\!] = \left[\!\!\left[\begin{array}{l} \texttt{while } (b_g \wedge \neg b_s) \texttt{ do } h[\texttt{false}] \,; \\ \texttt{while } (b_g \wedge b_s) \texttt{ do } h[\texttt{true}] \end{array} \right]\!\!\right]$$

Transferring Static Analysis Strategies

$\models \{b_s\}\, \overline{h}\, \{b_c\}$
$\models \{\neg b_s\}\, \overline{h}\, \{\neg b_c\}$
$\models \{b_g \wedge b_s\}\, h[b_c]\, \{\neg b_g \vee b_s\}$

⇔ sectionwise

$\models \{b_s|_i\}\, \overline{h}|_i\, \{b_c|_i\}$
$\models \{\neg b_s|_i\}\, \overline{h}|_i\, \{\neg b_c|_i\}$
$\models \{b_g|_i \wedge b_s|_i\}\, h|_i[b_c|_i]\, \{\neg b_g|_i \vee b_s|_i\}$

⋮

(for almost all i)

⇒ std. ver.

$[\![\ \texttt{while}\ b_g|_i\ \texttt{do}\ h|_i[b_c|_i]\ ]\!] = \left[\!\!\left[\ \begin{array}{l}\texttt{while}\ (b_g|_i \wedge \neg b_s|_i)\ \texttt{do}\ h|_i[\texttt{false}]\ ;\\ \texttt{while}\ (b_g|_i \wedge b_s|_i)\ \texttt{do}\ h|_i[\texttt{true}]\end{array}\ \right]\!\!\right]$

⋮

sectionwise ⇕

$[\![\ \texttt{while}\ b_g\ \texttt{do}\ h[b_c]\ ]\!] = \left[\!\!\left[\ \begin{array}{l}\texttt{while}\ (b_g \wedge \neg b_s)\ \texttt{do}\ h[\texttt{false}]\ ;\\ \texttt{while}\ (b_g \wedge b_s)\ \texttt{do}\ h[\texttt{true}]\end{array}\ \right]\!\!\right]$

* Doesn't matter what "std. ver." is
* → **modular method** for transfer

```
while (v > 0) {
  if m - z < s
    then a := -b
    else a := a0;
  t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0 && m - z >= s) {
  a := a0;    t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;    t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

accel.

brake

Startegy 1 "Phase split"

[Sharma,Dillig,Dillig,Aiken; CAV'11]
[Balakrishnan,Sankaranarayanan,Ivancic,Gupta; EMSOFT'09]
[Gopan,Reps; SAS'07]

```
while (v > 0 && m - z >= s) {
  a := a0;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

```
while (v > 0 && m - z >= s) {
  a := a0;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }

{z < m}
```

```
while (v > 0 && m - z >= s) {
  a := a0;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }

{z < m}
```

Startegies 2,3

"Superfluous guard elim." "Time elapse"

```
while (v > 0 && m - z >= s) {
  a := a0;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }};
while (v > 0 && m - z < s) {
  a := -b;   t := 0;
  while (t < eps && v > 0) {
    z := z + v * dt;
    v := v - b * dt;
    t := t + dt }}

{z < m}
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
while (v > 0) {
  a := -b;
```

Startegy 4

"Differential invariant"

[Platzer,Clarke; CAV'08]

Startegies 2,3

"Superfluous guard elim." **"Time elapse"**

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }

{z < m}
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }

{z < m}
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
```

$$\{ (v > 0 \lor m > z) \land (b^2 \mathrm{dt}^2 + 4b\mathrm{dt}v + 8bz + 4v^2 < 8bm \lor b\mathrm{dt}v + 2bz + v^2 \leq 2bm) \}$$

```
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }

{z < m}
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
```

$$\{ \; (v > 0 \lor m > z) \land (b^2 \mathrm{dt}^2 + 4b\mathrm{dt}v + 8bz + 4v^2 < 8bm \lor b\mathrm{dt}v + 2bz + v^2 \leq 2bm) \; \}$$

```
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

Startegy 5

"QE Invariant"

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;    t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
```

$$\left\{ \begin{array}{l} (v > 0 \lor m > z) \land \\ \left(b^2 \mathrm{dt}^2 + 4b\mathrm{dt}v + 8bz + 4v^2 < 8bm \right. \\ \quad \left. \lor\, b\mathrm{dt}v + 2bz + v^2 \leq 2bm\right) \end{array} \right\}$$

```
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
```

$$\{ \; (v > 0 \lor m > z) \land \left(b^2\mathrm{dt}^2 + 4b\mathrm{dt}v + 8bz + 4v^2 < 8bm \lor b\mathrm{dt}v + 2bz + v^2 \leq 2bm\right) \; \}$$

```
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

+ some fwd. propagation

```
{ ... (long fml. with dt) }
while (m - z >= s) {
  a := a0;   t := 0;
  while (t < eps) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }}
{ ... }
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

```
if (v > 0)
  then
    while (m - z >= s) {
      a := a0;   t := 0;
      while (t < eps) {
        z := z + v * dt;
        v := v + a0 * dt;
        t := t + dt }}
  else skip;
```

$$\{ \; (v > 0 \lor m > z) \land (b^2 \mathrm{dt}^2 + 4b\mathrm{dt}v + 8bz + 4v^2 < 8bm \lor b\mathrm{dt}v + 2bz + v^2 \leq 2bm) \; \}$$

```
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

+ some fwd. propagation

```
{ ... (long fml. with dt) }
while (m - z >= s) {
  a := a0;   t := 0;
  while (t < eps) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }}
{ ... }
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

Startegy 6

"Iteration count"

```
{ ... (long fml. with dt) }
while (m - z >= s) {
  a := a0;   t := 0;
  while (t < eps) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }}
{ ... }
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

```
{ ... (long fml. with dt) }
while (m - z >= s) {
  a := a0;   t := 0;
  while (t < eps) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }}
{ ... }
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

long fml. w/o `dt`, whose core is

$$a_0\left(2\varepsilon\sqrt{2a_0(m-s-z_0)+v_0^2}+b\epsilon^2+2m-2s-2z_0\right)$$
$$+2b\varepsilon\sqrt{2a_0(m-s-z_0)+v_0^2}+a_0^2\epsilon^2+v_0^2<2bs$$

```
{ ... (long fml. with dt) }
while (m - z >= s) {
  a := a0;   t := 0;
  while (t < eps) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }}
{ ... }
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

long fml. w/o `dt`, whose core is

$$a_0\left(2\varepsilon\sqrt{2a_0(m-s-z_0)+v_0^2}+b\epsilon^2+2m-2s-2z_0\right)$$
$$+2b\varepsilon\sqrt{2a_0(m-s-z_0)+v_0^2}+a_0^2\epsilon^2+v_0^2<2bs$$

Startegy 7

"Cast to shadow"

(Eliminates `dt`, strengthens the precond.)

```
{ ... (long fml. with dt) }
while (m - z >= s) {
  a := a0;   t := 0;
  while (t < eps) {
    z := z + v * dt;
    v := v + a0 * dt;
    t := t + dt }}
{ ... }
while (v > 0) {
  a := -b;
  z := z + v * dt;
  v := v - b * dt }
```

long fml. w/o dt, whose core is

$$a_0\left(2\varepsilon\sqrt{2a_0(m-s-z_0)+v_0^2}+b\epsilon^2+2m-2s-2z_0\right)$$
$$+2b\varepsilon\sqrt{2a_0(m-s-z_0)+v_0^2}+a_0^2\epsilon^2+v_0^2<2bs$$

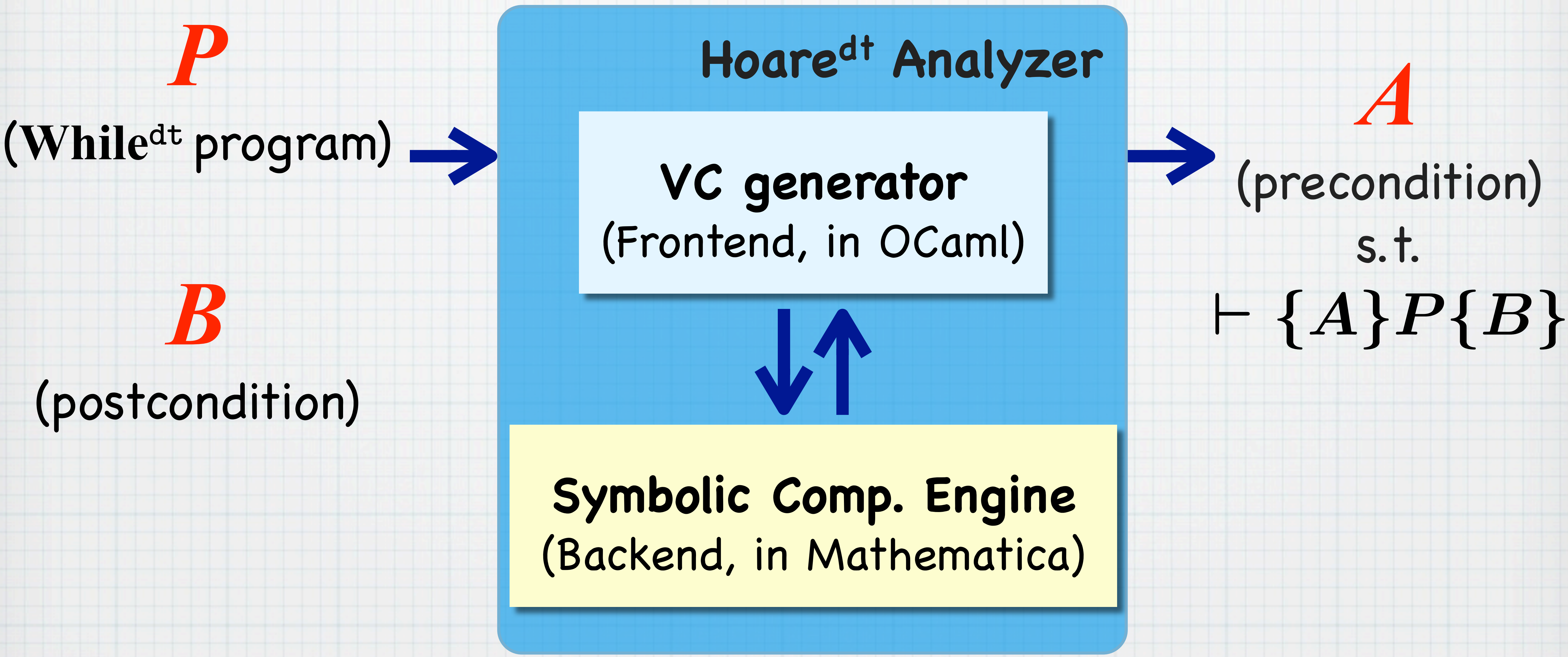
the final outcome

Startegy 7

"Cast to shadow"

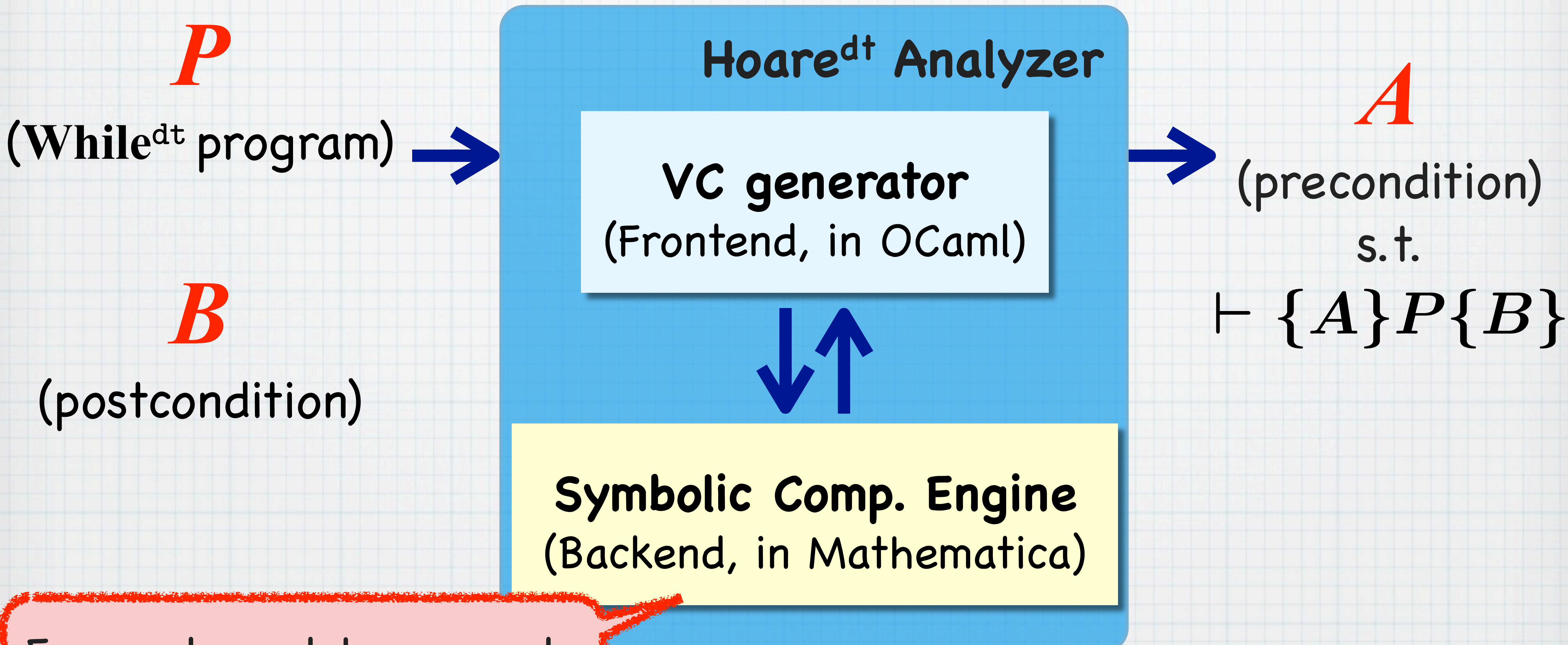
(Eliminates dt, strengthens the precond.)

Prototype Automatic Prover



- Fujitsu HX600 with Quad Core AMD Opteron 2.3GHz CPU, 32GB memory. Mathematica 7.0 for Linux x86 (64-bit)
  - ETCS: 40.96 sec.
  - Bouncing ball: runs with one manual insertion of invariants

Prototype Automatic Prover



For reals, not hyperreals
➜ justified by the **transfer principle**

Core AMD Opteron 2.3GHz CPU, 32GB memory.
x86 (64-bit)

th one manual insertion of invariants

Prototype Automatic Prover

P

(While^{dt} program) →

B

(postcondition)

Hoare^{dt} Analyzer

VC generator
(Frontend, in OCaml)

↓↑

Symbolic Comp. Engine
(Backend, in Mathematica)

→ A

(precondition)

s.t.

$\vdash \{A\}P\{B\}$

Totally symbolic
(crucial for transfer)

For reals, not hyperreals
➜ justified by the **transfer principle**

Core AMD Opteron 2.3GHz CPU, 32GB memory.

x86 (64-bit)

th one manual insertion of invariants

Related Work

- **Deductive verification** of hybrid sys. [Platzer, '10] [Platzer, LICS'12]
  - Automatic prover KeYmaera
- **Static analysis techniques**
  - A **LOT** in CAV, SAS, VMCAI, ...
  - Applied to hybrid systems (w/ diff. eq.)
    [Rodriguez-Carbonell, Tiwari; HSCC'05] [Sankaranarayanan; HSCC'10]
    [Sankaranarayanan, Sipma, Manna; Formal Methods Sys. Design '08]
- Use of **NSA for hybrid systems**
  [Benveniste, Bourke, Caillaud, Pouzet; J. Comput. Syst. Sci. '12]
  [Bliudze, Krob; Fundam. Inform. '09] [Gamboa, Kaufmann; J. Autom. Reason. '01]
- **Continuous techniques applied to discrete appl.**
  [Chaudhuri, Gulwani, Lublinerman, NavidPour; FSE '11]
  - Not contending! Combination?

Conclusions

$\textbf{While}^{\text{dt}}$

Programming lang.

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

$\textbf{Assn}^{\text{dt}}$

First-order assertion lang.

∃z(x=2*z ∧ y=3*z)

$\textbf{Hoare}^{\text{dt}}$

Hoare-style program logic

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

Rigorous semantics by non-standard analysis

Conclusions

Nonstandard Static Analysis

$\mathbf{While}^{dt}$

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

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

Rigorous semantics by non-standard analysis

Conclusions

Nonstandard Static Analysis

While^{dt}

Programming lang.

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

Assn^{dt}

First-order assertion lang.

∃z(x=2*z ∧ y=3*z)

Hoare^{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 non-standard analysis

- Tool's effectivity. More heuristics?
- (Any discrete frmwk.)^{dt} ?
- Simulink as stream processing?
- With (explicit) differential equations?

Conclusions

Nonstandard Static Analysis

$\mathbf{While}^{dt}$

Programming lang.

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

$\mathbf{Assn}^{dt}$

First-order assertion lang.

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

$\mathbf{Hoare}^{dt}$

Hoare-style program logic

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

Rigorous semantics by non-standard analysis

* Tool's effectivity. More heuristics?
* $(\text{Any discrete frmwk.})^{dt}$?
* Simulink as stream processing?
* With (explicit) differential equations?

Thank you for your attention!

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