

Infinately
small values

Programming with **Infinitesimals**

A WHILE-Language for
Hybrid System Modeling

(To appear in ICALP 2011)

Kohei Suenaga
JSPS Research Fellow (PD)
Kyoto University

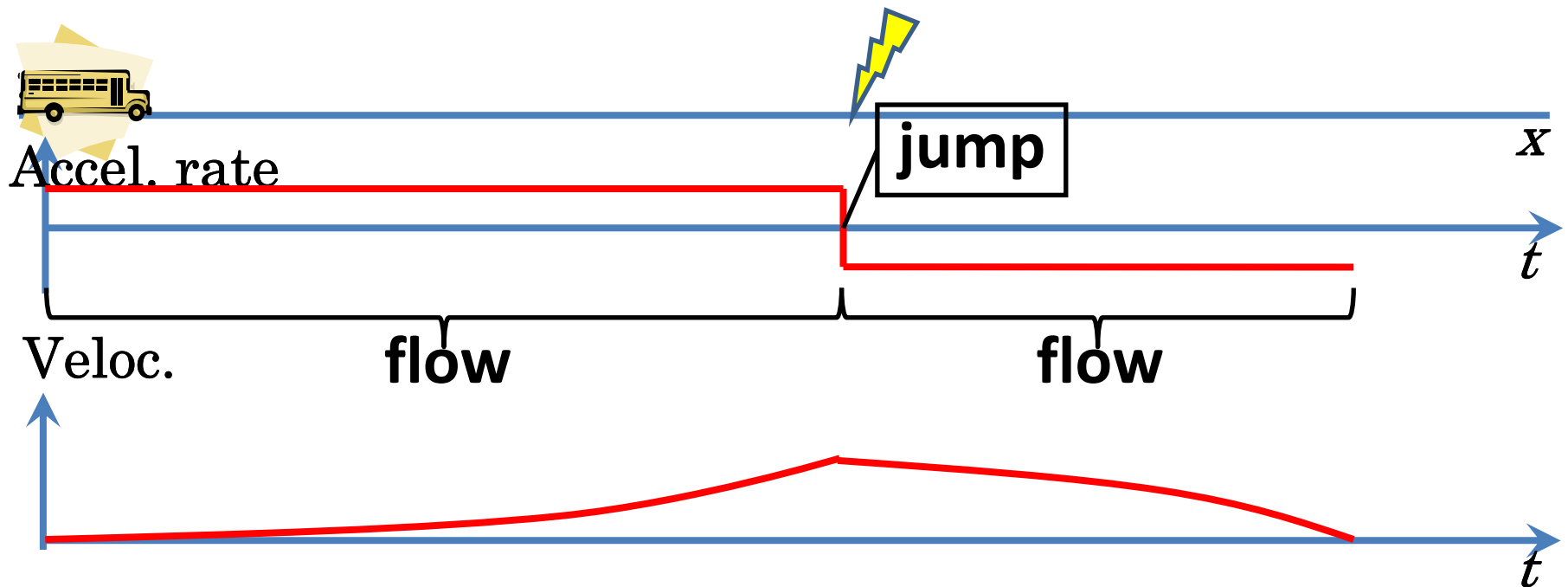
Ichiro Hasuo
University of Tokyo

Hybrid systems

Flow + Jump

Jump is familiar

Flow is not



This work

Turns flow into jump

This work

Turns flow into jump

With infinitesimals

This work

Turns flow into jump

With infinitesimals



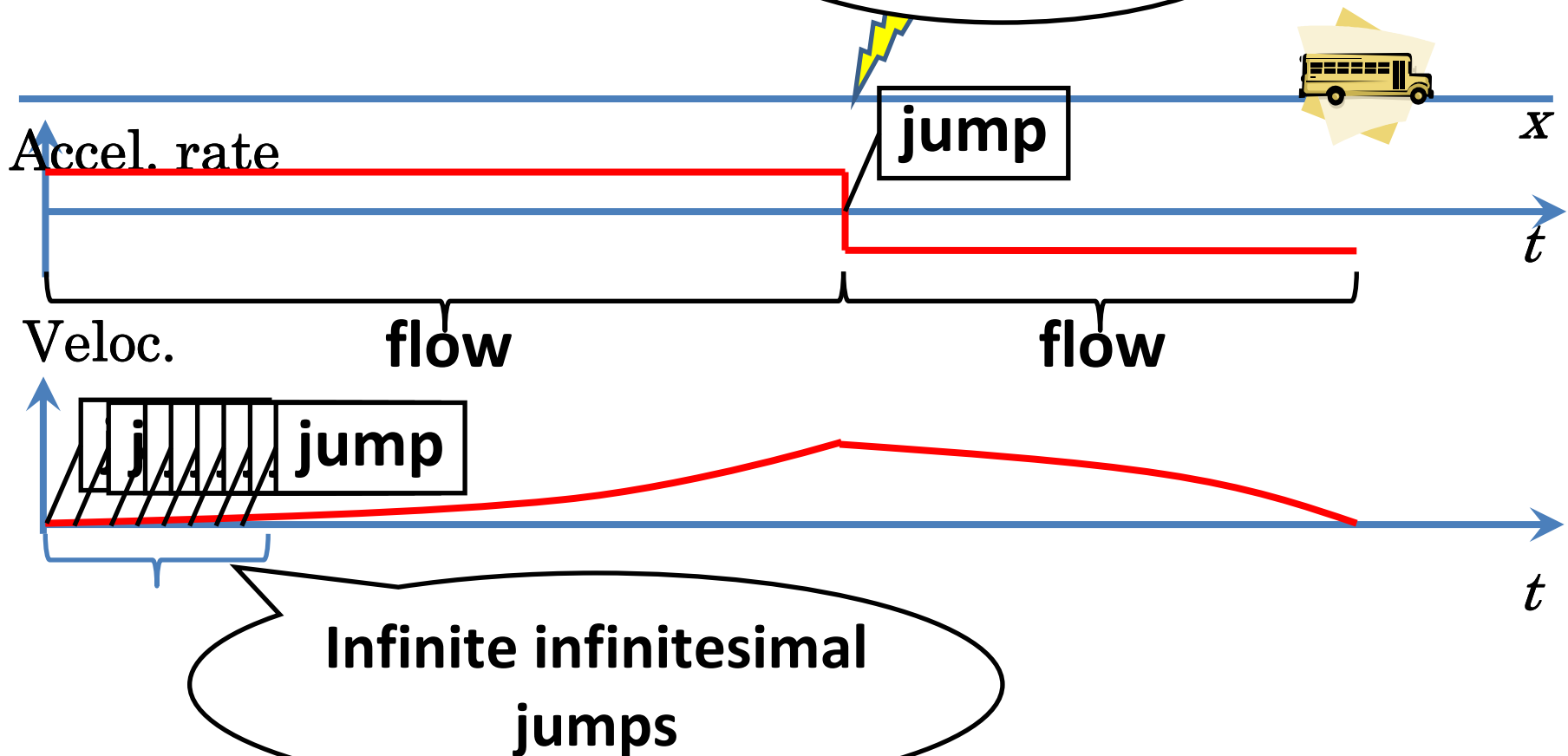
Infinitely
small values

This work

Turns flow into jump

With infinitesimals

Infinitely small values



Concretely

Programming language **WHILE^{dt}**

WHILE + dt

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

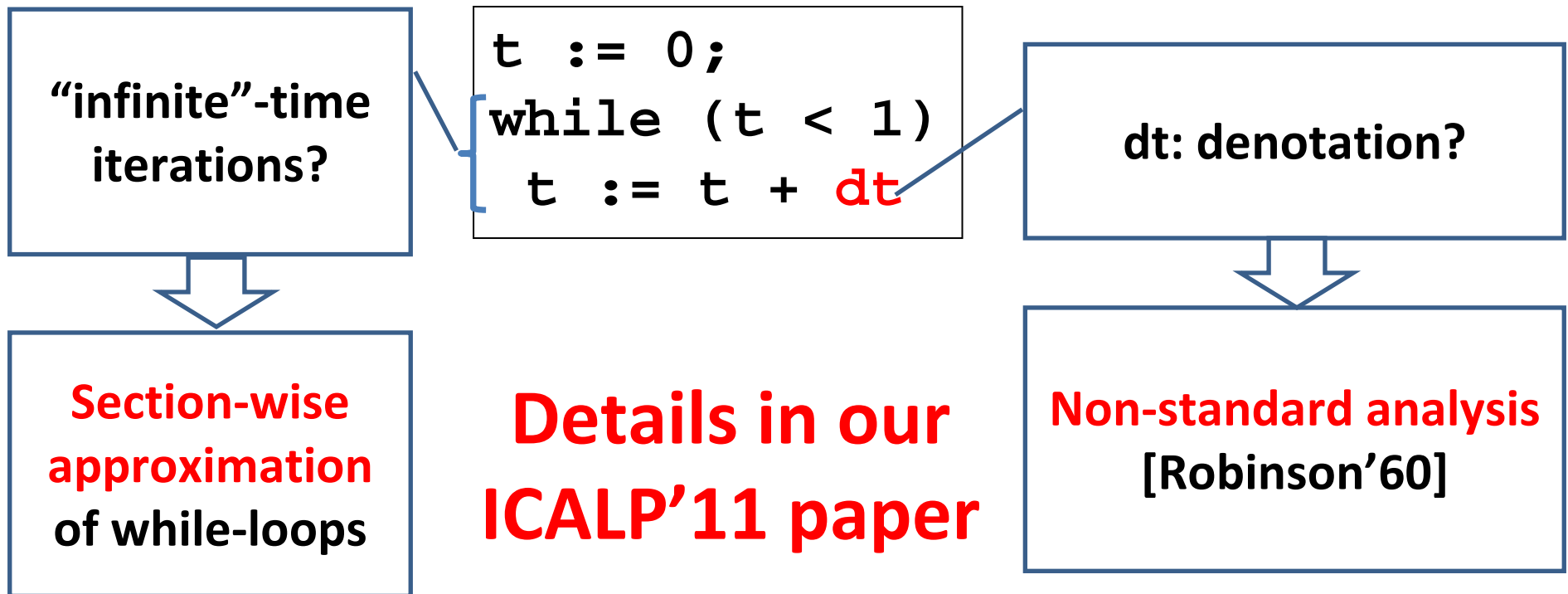
infinitesimal constant
("infinitely small")

Hoare-style program logic (**HOARE^{dt}**) for **WHILE^{dt}**

Sound and relatively complete

Challenges

Mathematically rigorous basis



Modeling in WHILE^{dt}

```
t := 0; x := 0; v := 0; a := 1;  
while (t < 4) {  
  v' := v + a * dt;  
  x' := x + v * dt;  
  v := v'; x := x';  
  
  t := t + dt;  
  a := (x >= 2) ? 1 : -1;  
}
```

Modeling in WHILE^{dt}

```
t := 0; x := 0; v := 0; a := 1;
while (t < 4) {
  v' := v + a * dt;
  x' := x + v * dt;
  v := v'; x := x';

  t := t + dt;
  a := (x >= 2) ? 1 : -1;
}
```

$$dx/dt = v$$

$$dv/dt = a$$

Verification in Hoare^{dt}

{ A } C { B }

Verification in Hoare^{dt}

{ true }

```
t := 0; x := 0; v := 0; a := 1;
while (t < 4) {
  v' := v + a * dt;
  x' := x + v * dt;
  v := v'; x := x';
  t := t + dt;
  a := (x >= 2) ? 1 : -1;
}
```

{ x < 4.01 }

In the paper...

WHILE^{dt}

HOARE^{dt}

Proofs

Examples

Preprint at Suenaga's webpage

Google: Kohei Suenaga