INTEGRABILITY IN NONSTANDARD MODELLING OF HYBRID SYSTEMS

Kengo Kido
(MSc student at The Univ. of Tokyo)
Jan. 25, 2014

- **While**$^{\mathtt{dt}}$ [Suenaga and Hasuo, ICALP '11,
  Hasuo and Suenaga, CAV '12,
  Suenaga, Sekine and Hasuo, POPL '13]

- Integrability

  - If modelled naively, WHILE$^{\mathrm{dt}}$ program may have a problem.

  - Integrability means that the problem does not occur.

While$^{\mathtt{dt}}$

infinitesimal

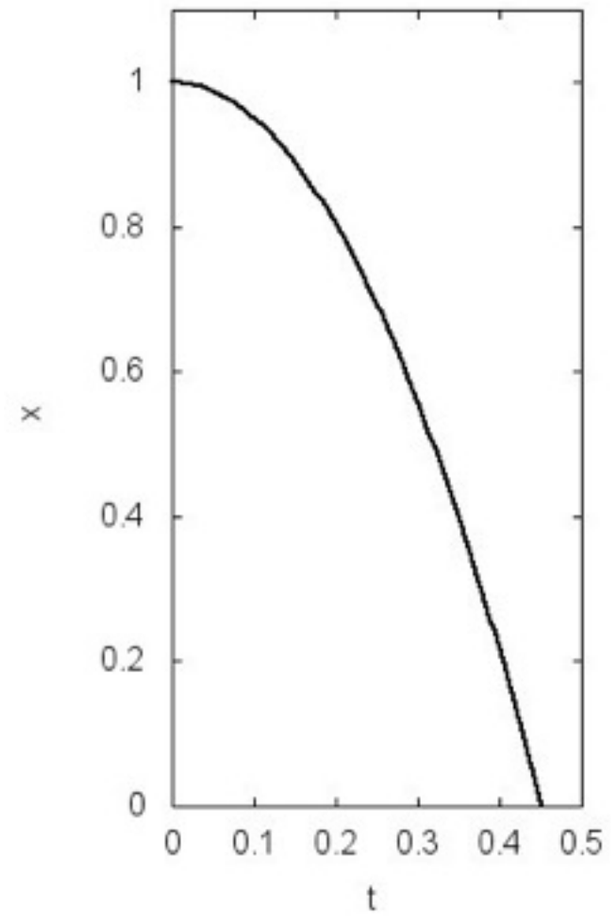
AExp $\ni$ $\quad a ::= x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop} \ a_2 \mid \mathtt{dt}$

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

Cmd $\ni$ $\quad c ::= \mathtt{skip} \mid x := a \mid c_1 ; c_2$

$\quad \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c$

Modelling in While^{dt}



$dx/dt = v$

$dv/dt = -9.8 * dt$

in WHILE^{dt}

```
x := 1;
v := 0;
t := 0;
while ( x > 0 ) do {
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt
}
```

- Hoare logic is applicable
  (exactly the same rules)

- Automatic theorem prover

  [Hasuo and Suenaga, CAV '12]

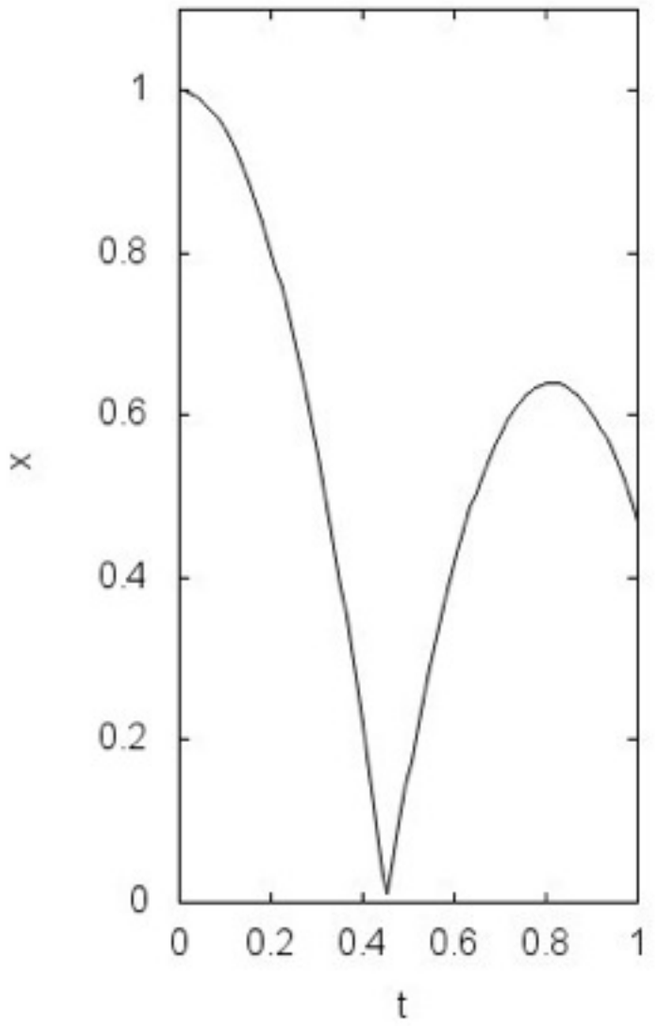
- **While^{dt}** [Suenaga and Hasuo, ICALP '11,
  Hasuo and Suenaga, CAV '12,
  Suenaga, Sekine and Hasuo, POPL '13]

- Integrability

  - If modelled naively, WHILE^{dt} program may have a problem.

  - Integrability means that the problem does not occur.

Modelling in While^{dt}



dx/dt = v

dv/dt = -9.8 * dt

in WHILE^{dt}

```
x := 1;
v := 0;
t := 0;
while ( t < 1 ) do {
    if ( x < 0 ) then
        v := -0.8 * v;
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt
}
```

Nonstandard Analysis [Robinson '61]

- infinitesimals ≈ (sequences → 0)

e.g. $(1, \frac{1}{2}, \frac{1}{3}, \ldots), (\frac{1}{\pi}, \frac{1}{2\pi}, \frac{1}{3\pi}, \ldots)$

- semantics of a program $[\![c]\!]^{\partial}$

infinitesimal

assuming $\partial = (1, \frac{1}{2}, \ldots, \frac{1}{n}, \ldots)$

$t := 0;$ `while` $(t \leq 1)$ `do` $t := t + \mathbf{1}$, $t := 0;$ `while` $(t \leq 1)$ `do` $t := t + \frac{\mathbf{1}}{\mathbf{2}}$, ⋯ , $t := 0;$ `while` $(t \leq 1)$ `do` $t := t + \frac{\mathbf{1}}{\mathbf{n}}$, ⋯ → $t := 0;$ `while` $(t \leq 1)$ `do` $t := t +$ `dt`

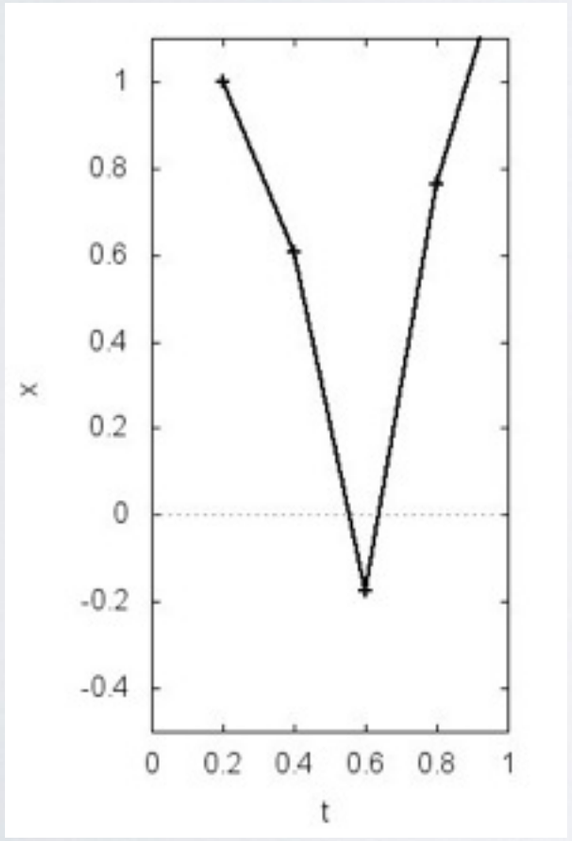
$$t = 2 \quad , \quad \frac{3}{2} \quad , \ldots , \quad 1 + \frac{1}{n} \quad , \ldots \longrightarrow \quad 1$$

$$[\![\texttt{dt}]\!]^{\partial} = (\ r_1, r_2, \ldots\ )$$

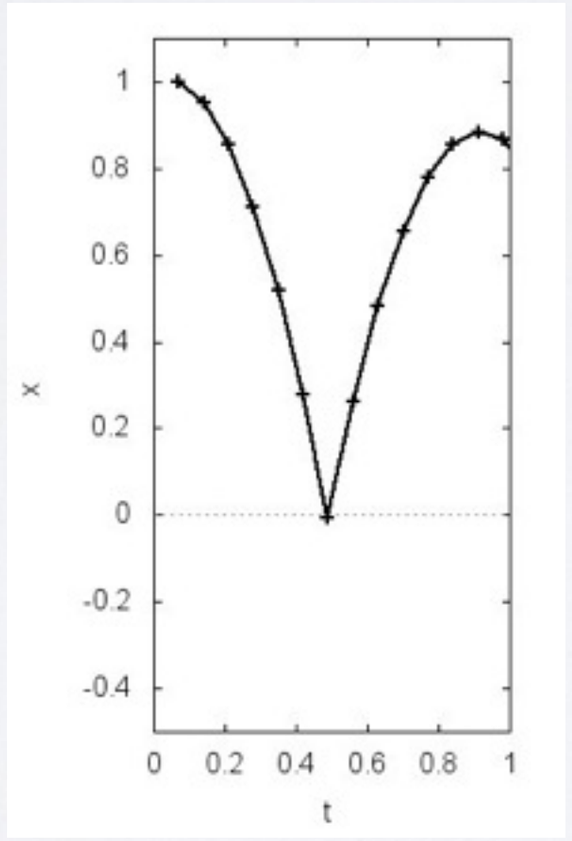
```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt}
```

```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * r1;
    v := v - 9.8 * r1;
    t := t + r1}
```

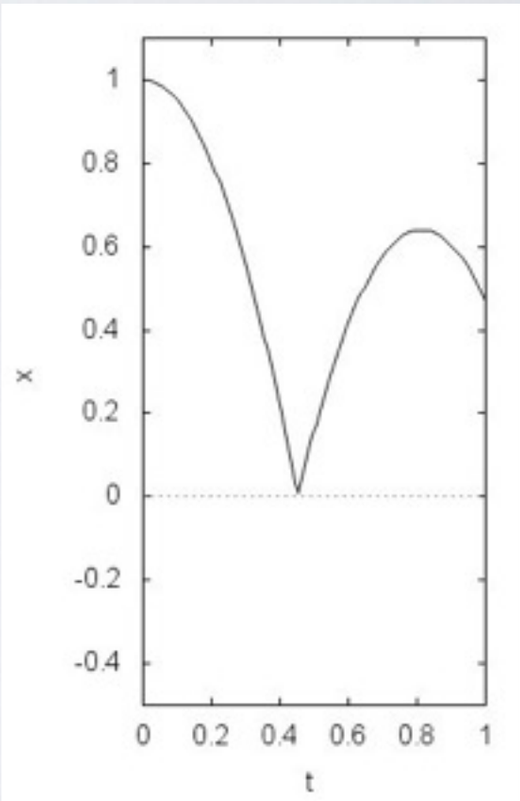
```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * r2;
    v := v - 9.8 * r2;
    t := t + r2}
```



,



, ...

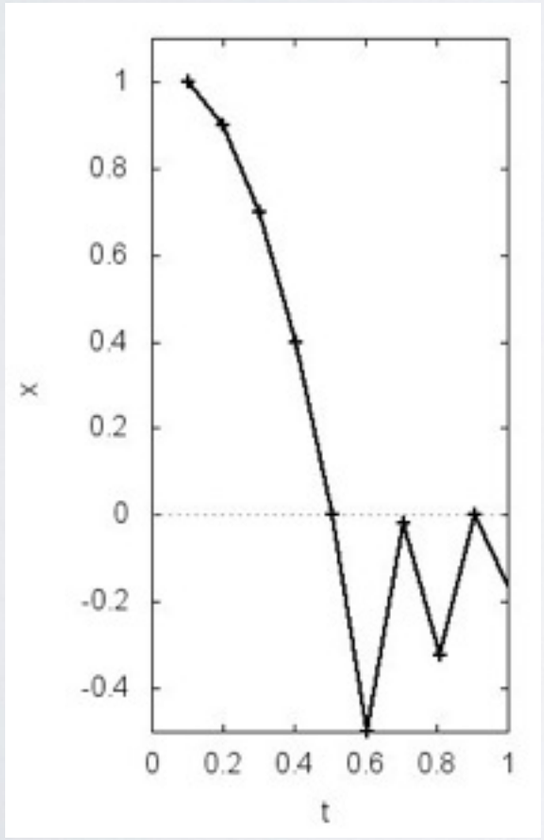


$$[\![\texttt{dt}]\!]^{\partial'} = (\, r_1', r_2', \ldots \,)$$

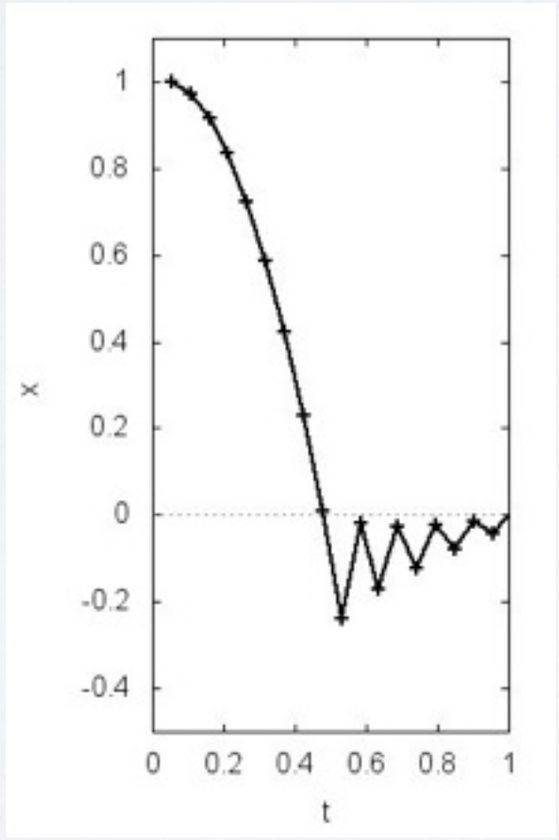
```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt}
```

```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * r1';
    v := v - 9.8 * r1';
    t := t + r1'}
```

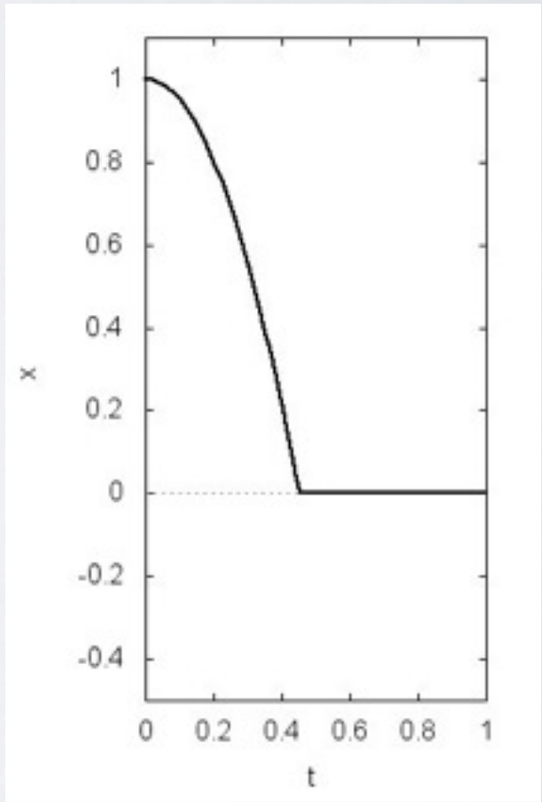
```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * r2';
    v := v - 9.8 * r2';
    t := t + r2'}
```



,



, ...



Problem

choices of ⟦dt⟧ result in gaps that is greater than infinitesimal.

infinitely small difference

Integrability

a **While**$^{\mathtt{dt}}$ program c is *integrable*

if $[\![c]\!]^{\partial}\sigma \simeq [\![c]\!]^{\partial'}\sigma$ for any state σ and

for any infinitesimals ∂ and ∂'.

Riemann Integrability

non-integrable (previous example)

```
x := 1; v := 0; t := 0;
while ( t < 1 ) do {
    if ( x < 0 ) then
        v := -0.8 * v;
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt
}
```

integrable 1

```
x := 1; v := 0; t := 0;
while ( t < 1 ) do {
    if ( x < 0 ∧ v < 0) then
        v := -0.8 * v;
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt
}
```

integrable 2

```
x := 1; v := 0; t := 0;
while ( t < 1 ) do {
    if ( x < 0 ) then {
        v := -0.8 * v;
        x := 0;
    }
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt
}
```

Conclusions

Identified integrability of **While**$^{\texttt{dt}}$ programs

We are looking at

- Verification of integrability
- Automatic correction of non-integrable programs

Possible approach

Applying **non-interference verification** [Terauchi and Aiken, SAS '05, Sabelfeld and Sands, CSF '00]

Non-interference

affects the observable value

Value of high-security variable…

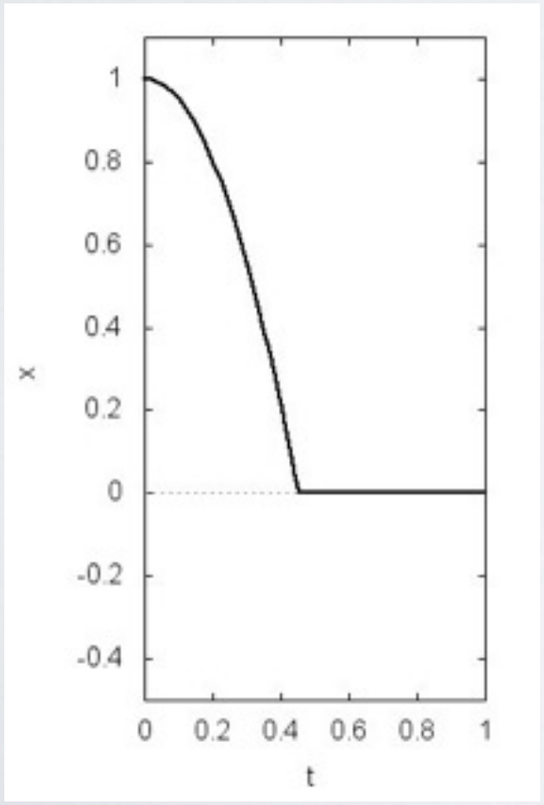
```
if (h > 0) then
   l := 0;
else
   l := 1;
```

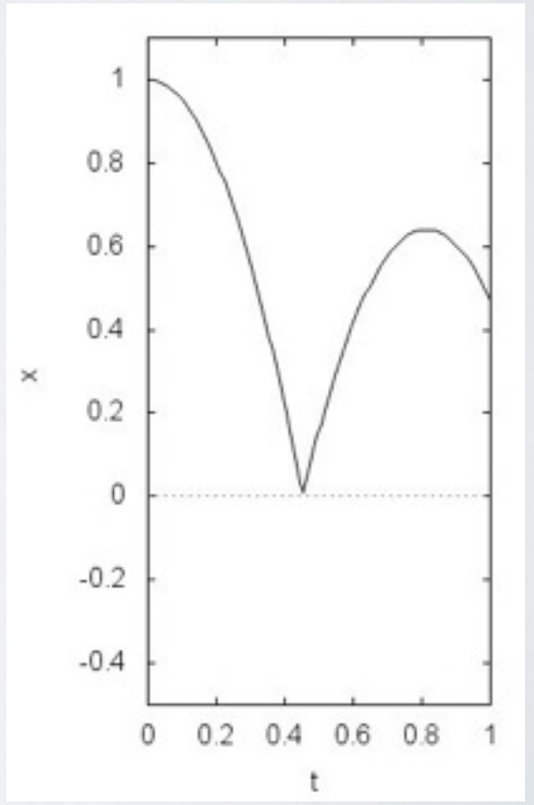
Integrability and Non-interference

affects the observable value

Value of dt…

```
x := 1; v := 0; t := 0;
while (t < 1) do {
    if (x < 0) then
        v := -0.8 * v;
    x := x + v * dt;
    v := v - 9.8 * dt;
    t := t + dt}
```





Conclusions

Identified integrability of **While**$^{\texttt{dt}}$ programs

We are looking at

- Verification of integrability
- Automatic correction of non-integrable programs

Possible approach

Applying **non-interference verification** [Terauchi and Aiken, SAS '05, Sabelfeld and Sands, CSF '00]