超準解析による
物理情報システムの形式検証

—— 離散から連続・ハイブリッドへ ——

蓮尾 一郎

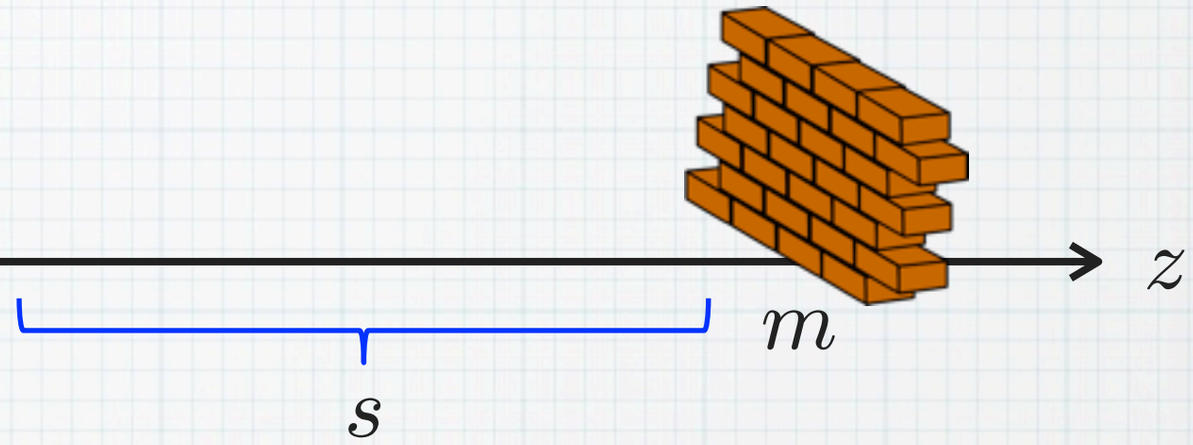
東京大学 大学院情報理工学系研究科 コンピュータ科学専攻・
理学部 情報科学科

http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro

まず最初に．．．

デモを

問題

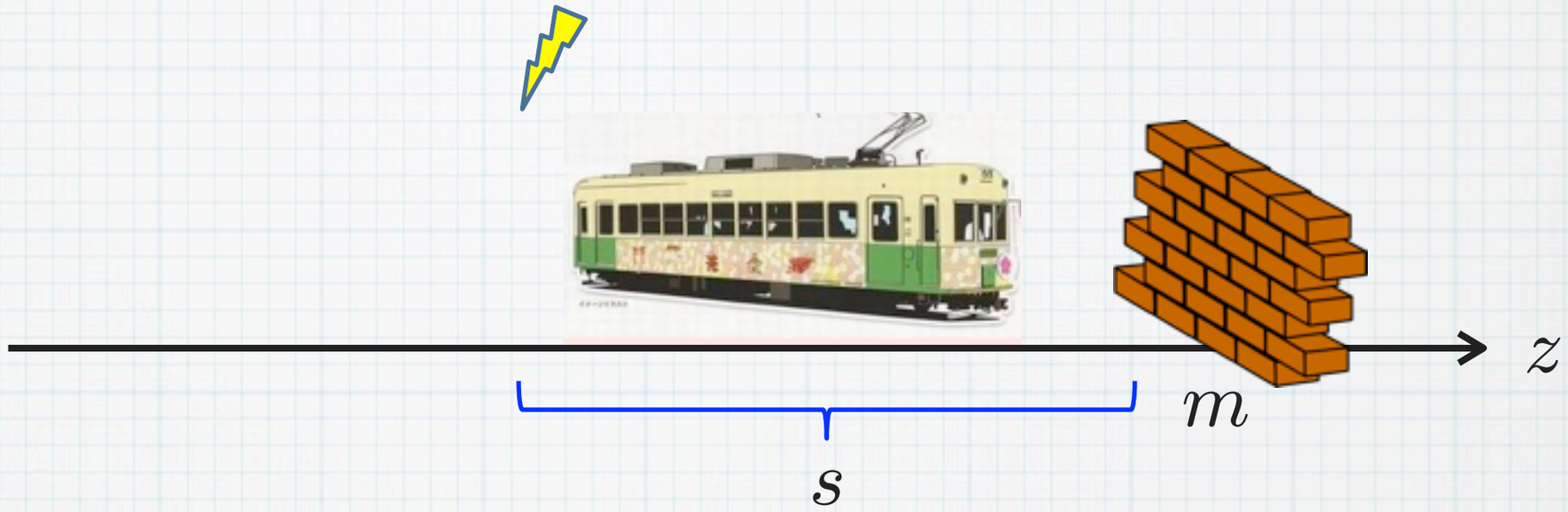


$$\dot{z} = v$$

$$\dot{v} = \begin{cases} a_0 & \text{if } m - z > s \\ -b & \text{if } m - z < s \end{cases}$$

Hasuo (Tokyo)

問題

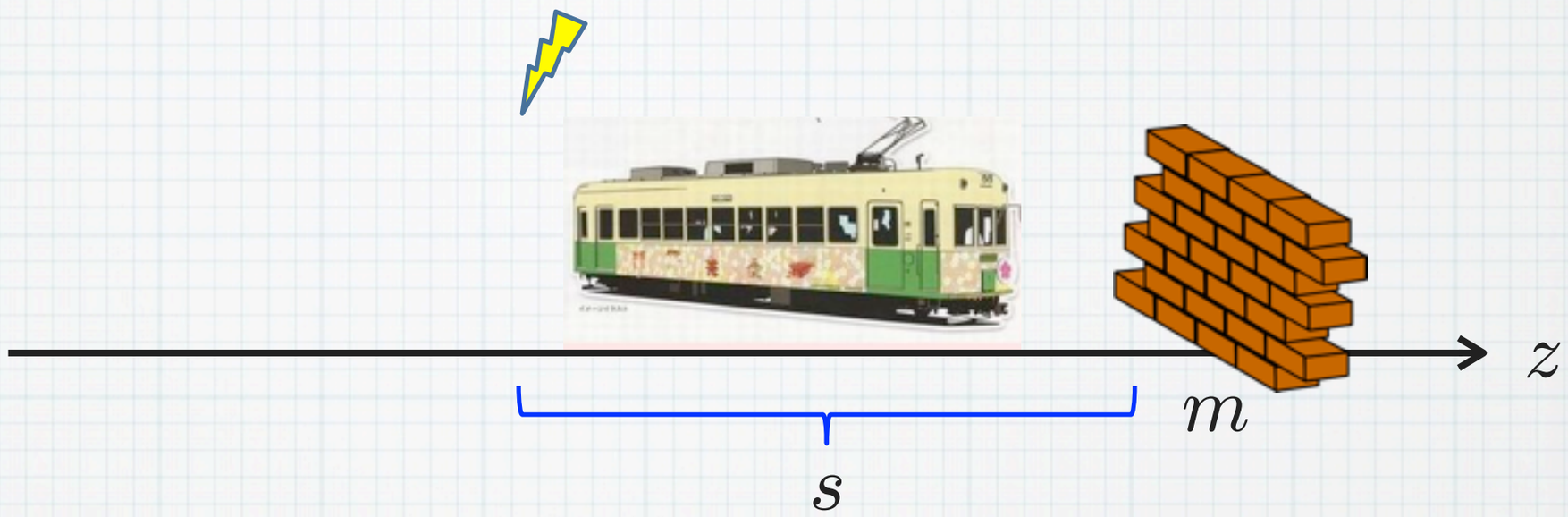


$$\dot{z} = v$$

$$\dot{v} = \begin{cases} a_0 & \text{if } m - z > s \\ -b & \text{if } m - z < s \end{cases}$$

Hasuo (Tokyo)

問題



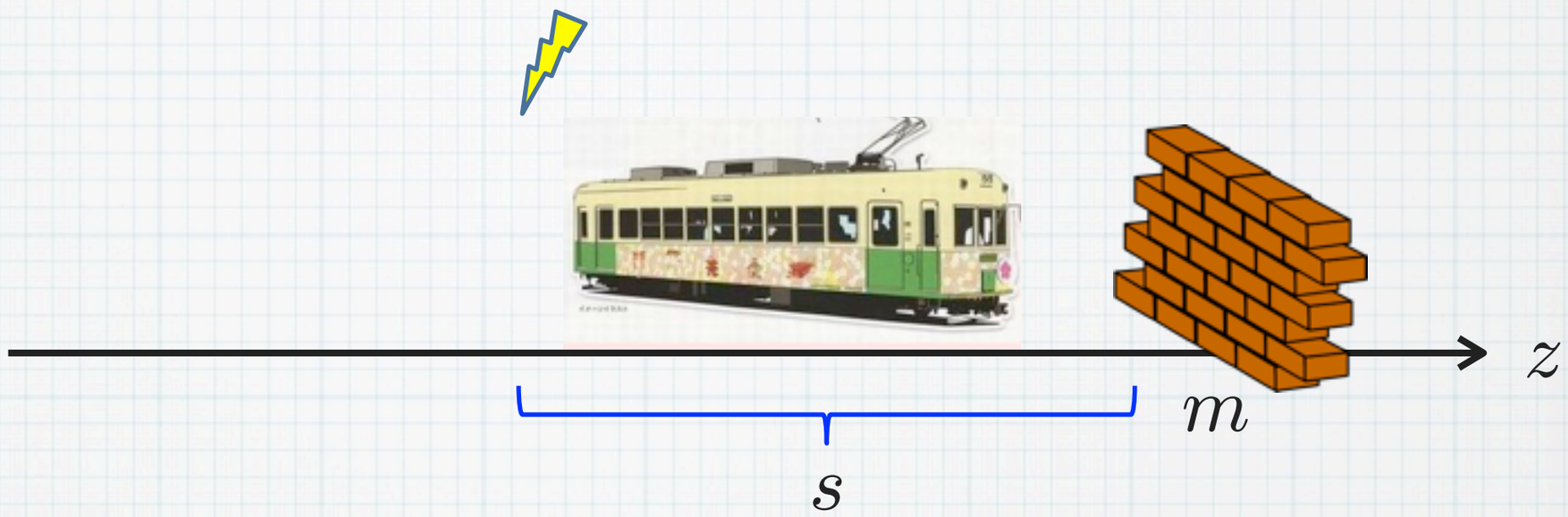
$$\dot{z} = v$$

$$\dot{v} = \begin{cases} a_0 & \text{if } m - z > s \\ -b & \text{if } m - z < s \end{cases}$$

* 等加速度運動
* 安全距離 s で
スイッチング

Hasuo (Tokyo)

問題



$$\dot{z} = v$$

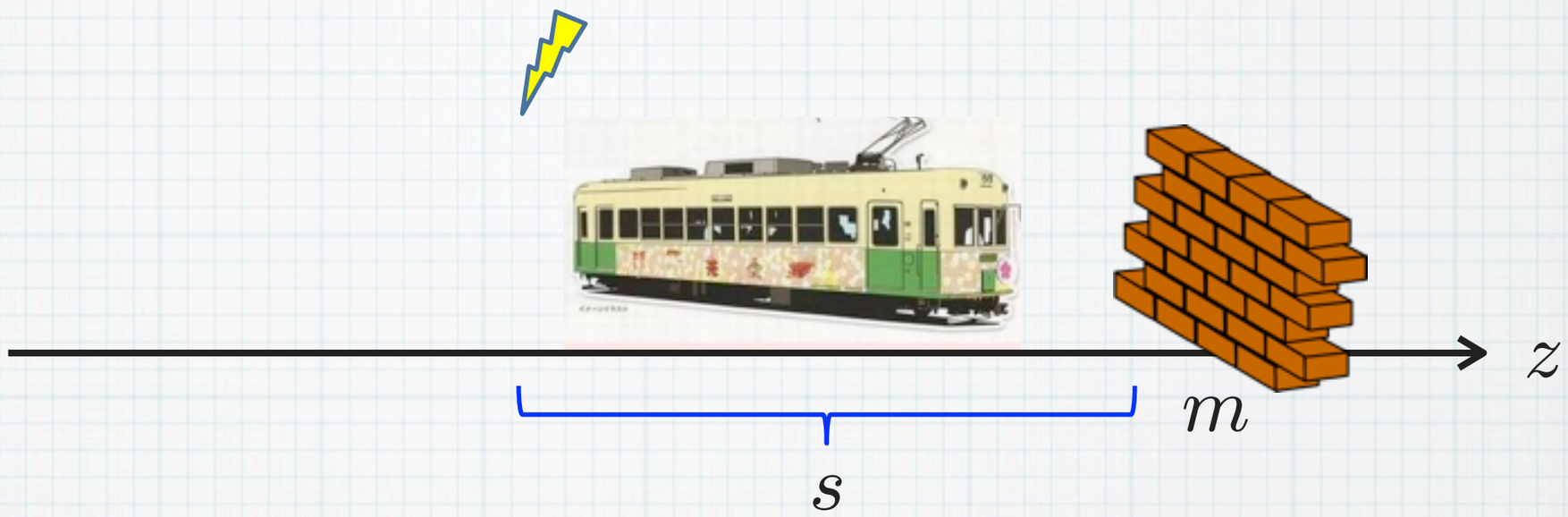
$$\dot{v} = \begin{cases} a_0 & \text{if } m - z > s \\ -b & \text{if } m - z < s \end{cases}$$

* 等加速度運動
* 安全距離 s で
スイッチング

Q. カベにあたらずに止まるための
十分条件は？

Hasuo (Tokyo)

問題



$$\dot{z} = v$$

$$\dot{v} = \begin{cases} a_0 & \text{if } m - z > s \\ -b & \text{if } m - z < s \end{cases}$$

* 等加速度運動
* 安全距離 s で
スイッチング

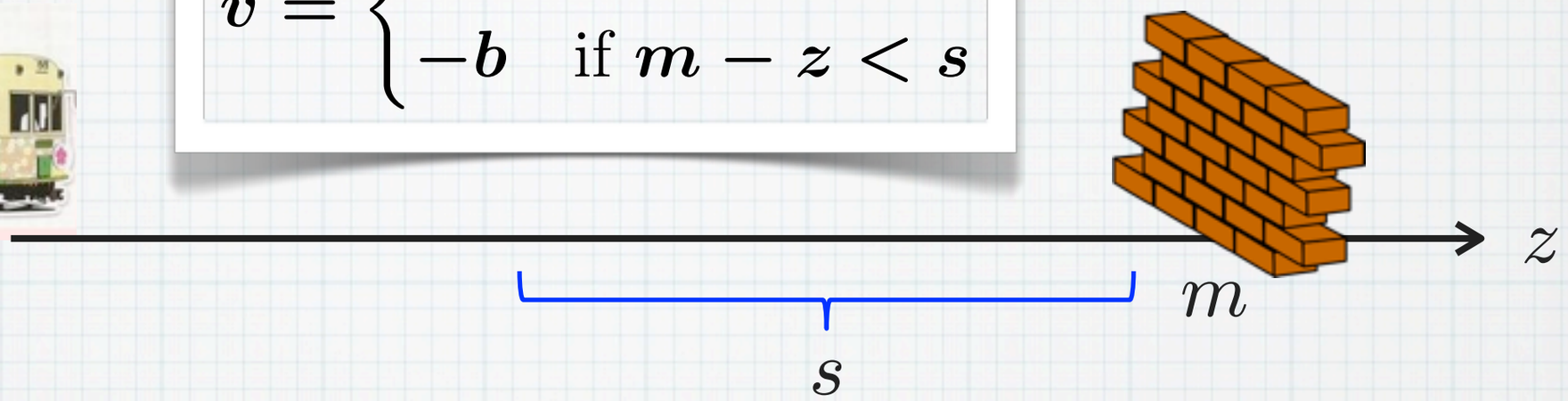
Q. カベにあたらずに止まるための
十分条件は？

s: big enough
b: big enough
a_0: small enough
...

問題

$$\dot{z} = v$$

$$\dot{v} = \begin{cases} a_0 & \text{if } m - z > s \\ -b & \text{if } m - z < s \end{cases}$$



Q. カベにあたらずに止まるための
十分条件は？

* 全自動で条件生成するツール
* 「正しさの証明」付き

Hasuo (Tokyo)

ツールへの入力

```
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}

ツールへの入力

```
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}
```

* プログラム？
* 微分方程式はどこ？

Hoare^{dt} Analyzer

デモ

超準解析による
物理情報システムの形式検証

—— 離散から連続・ハイブリッドへ ——

蓮尾 一郎

東京大学 大学院情報理工学系研究科 コンピュータ科学専攻・
理学部 情報科学科

http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

数学（数理論理学）
による品質保証

* 微小量（infinitesimal）を持つ，
  解析学の形式化
* Ultrafilter によるモデル
  ➜ 数理論理学（モデル理論）の成果！

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

数学（数理論理学）
による品質保証

超準解析による
物理情報システムの形式検証

アウトライン

* 形式検証とは？
  * Hoare 論理を例に
* ハイブリッド・システム
  * 離散＋連続
  * 物理情報システムの一側面
* 超準解析による移転
  * 離散的検証手法を，文字通りそのまま
    ハイブリッド・システムに適用

共同研究者

* 末永 幸平
  京都大学 情報学研究科
* Swarat Chaudhuri
  Rice University (US)
* 片岡 俊基，木戸 肩吾
  東京大学 情報理工学系研究科 博士課程
* 関根 大剛
  東京大学 情報理工学系研究科 修士課程（OB）

論文

* [ICALP'11] K. Suenaga and I. Hasuo. Programming with infinitesimals: A while-language for hybrid system modeling. In L. Aceto, M. Henzinger and J. Sgall, editors, ICALP (2), vol. 6756 of Lect. Notes Comp. Sci., pp. 392–403. Springer, 2011.

* [CAV'12] I. Hasuo and K. Suenaga. Exercises in Nonstandard Static Analysis of hybrid systems. In P. Madhusudan and S.A. Seshia, editors, CAV, vol. 7358 of Lect. Notes Comp. Sci., pp. 462–478. Springer, 2012.

* [POPL'13] K. Suenaga, H. Sekine and I. Hasuo. Hyperstream processing systems: nonstandard modeling of continuous-time signals. In R. Giacobazzi and R. Cousot, editors, POPL, pp. 417–430. ACM, 2013.

論文・スライドなどはウェブページから

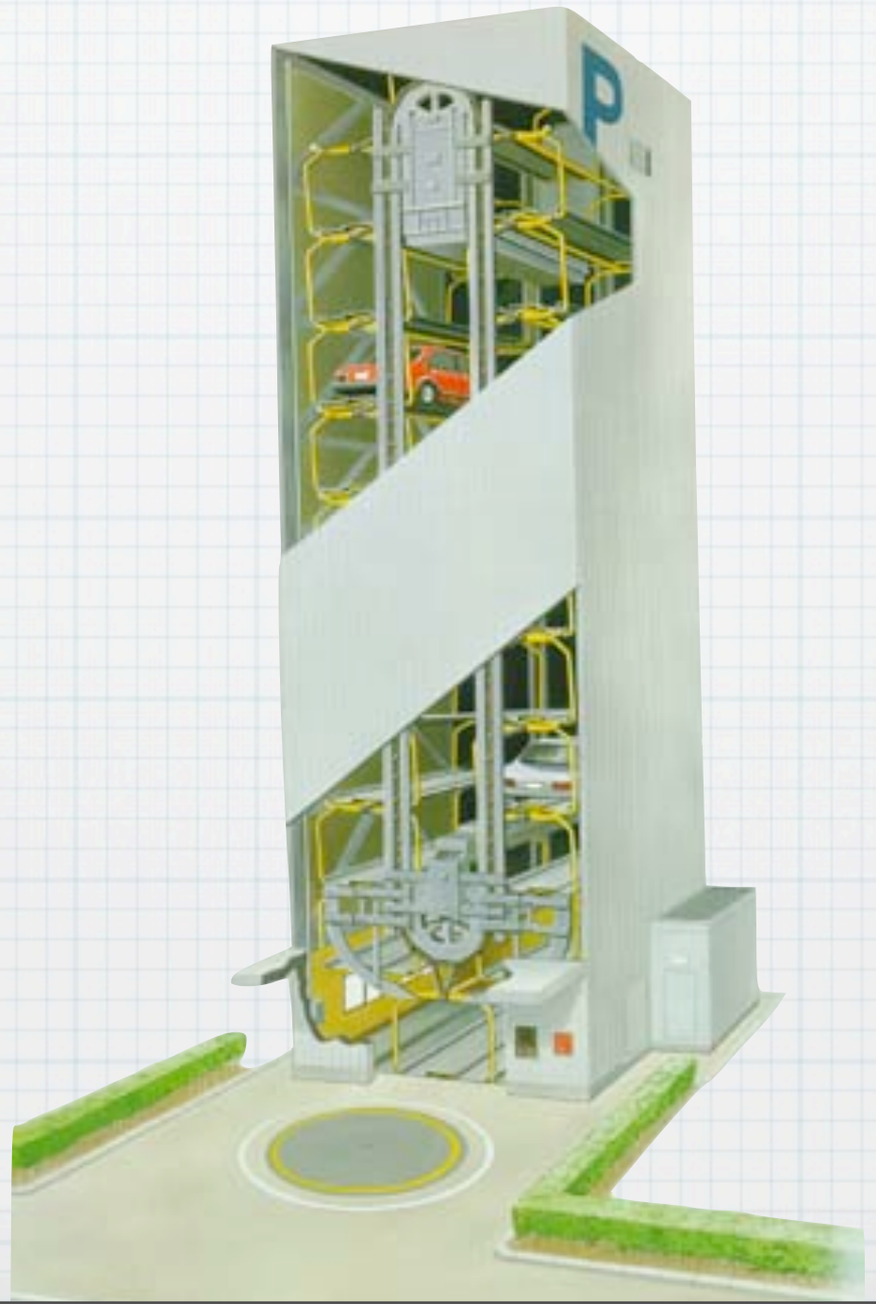
http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro/

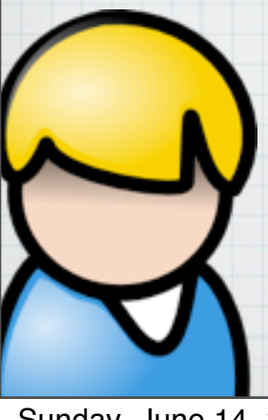
1

形式検証とは

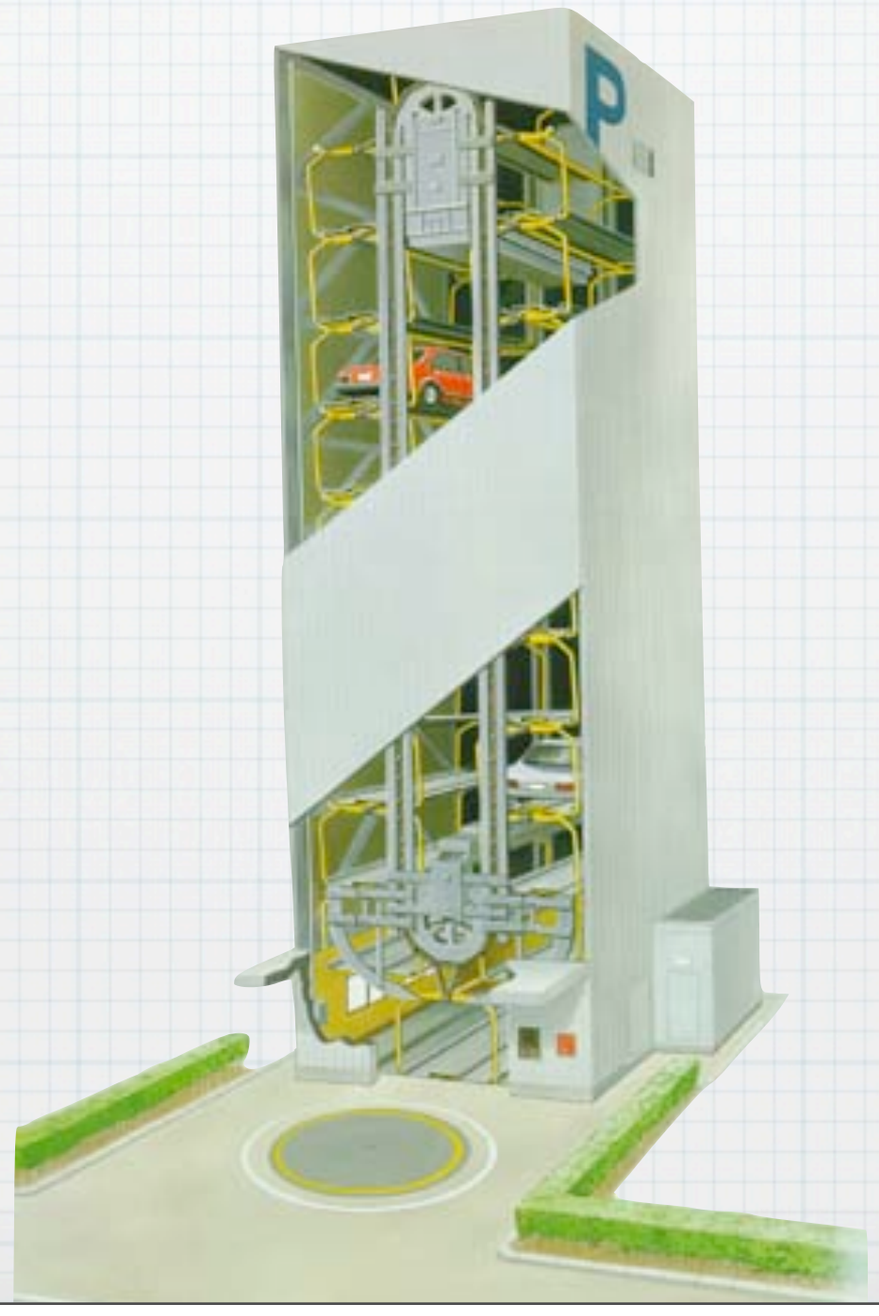
(Hoare 論理を例に)

証明で金もうけ



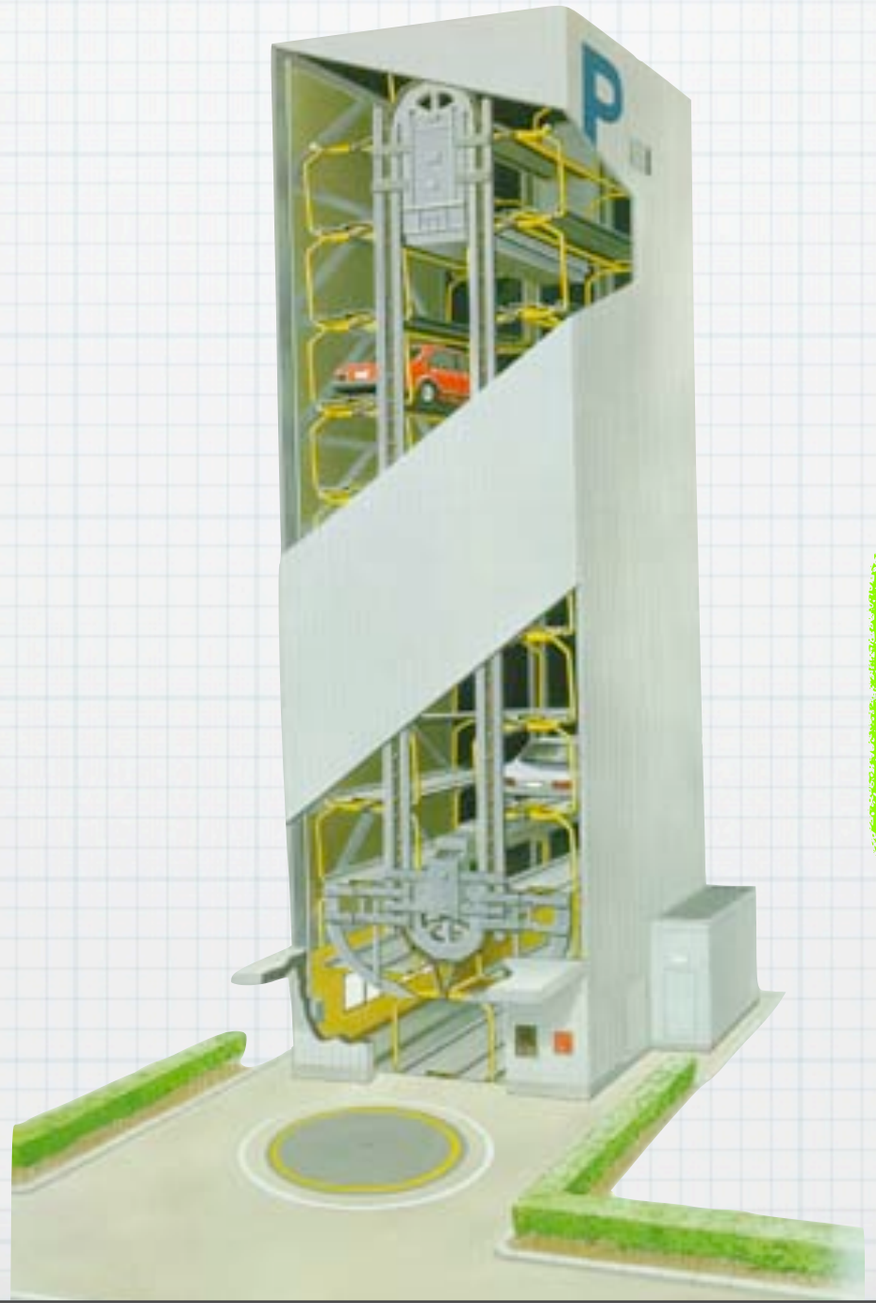


証明で金もうけ



この駐車場,
どうっすか！？

証明で金もうけ



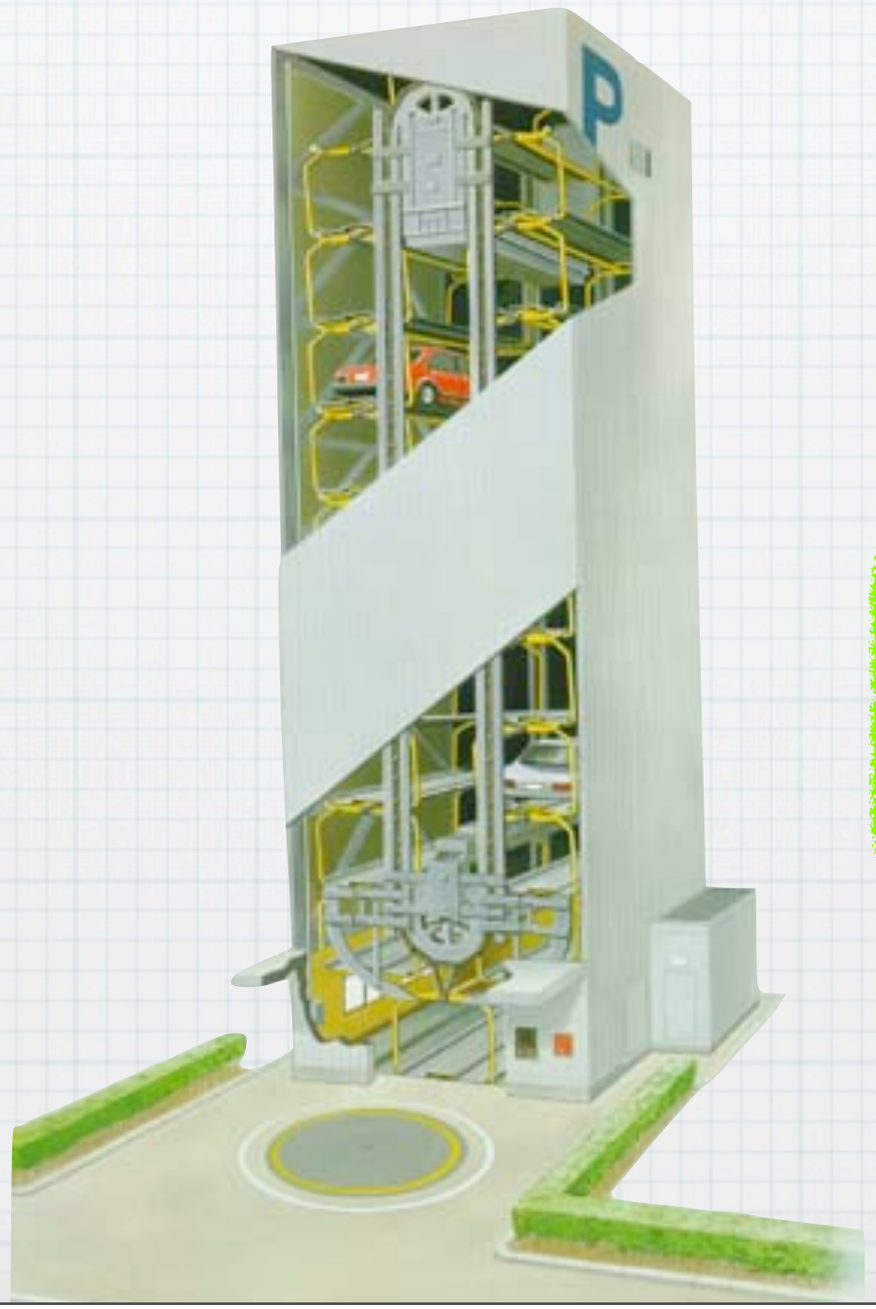
この駐車場,
どうっすか！？

ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ

絶対大丈夫っす！

この駐車場,
どうっすか！？

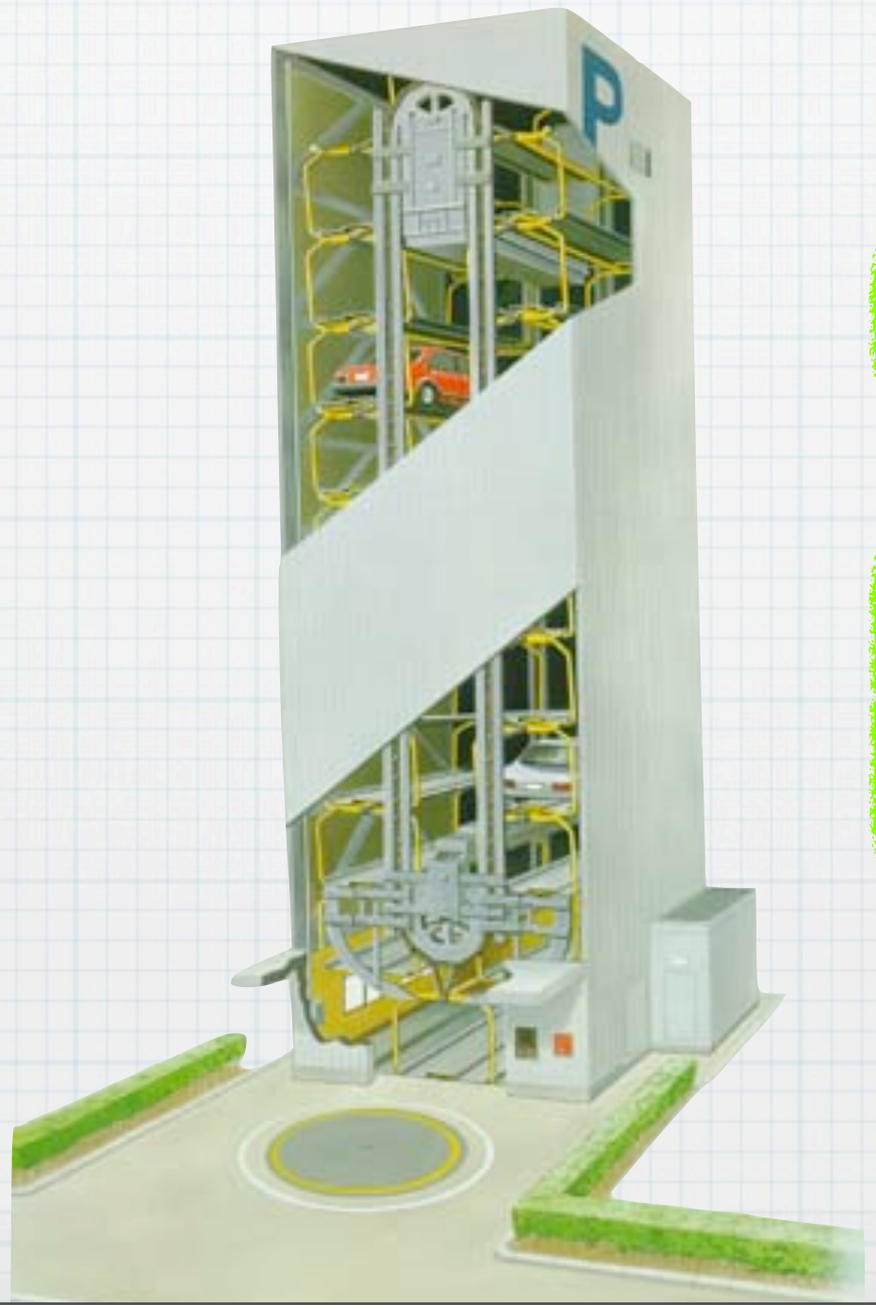


ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ

絶対大丈夫っす！

この駐車場,
どうっすか！？



ホントに？ なんで？

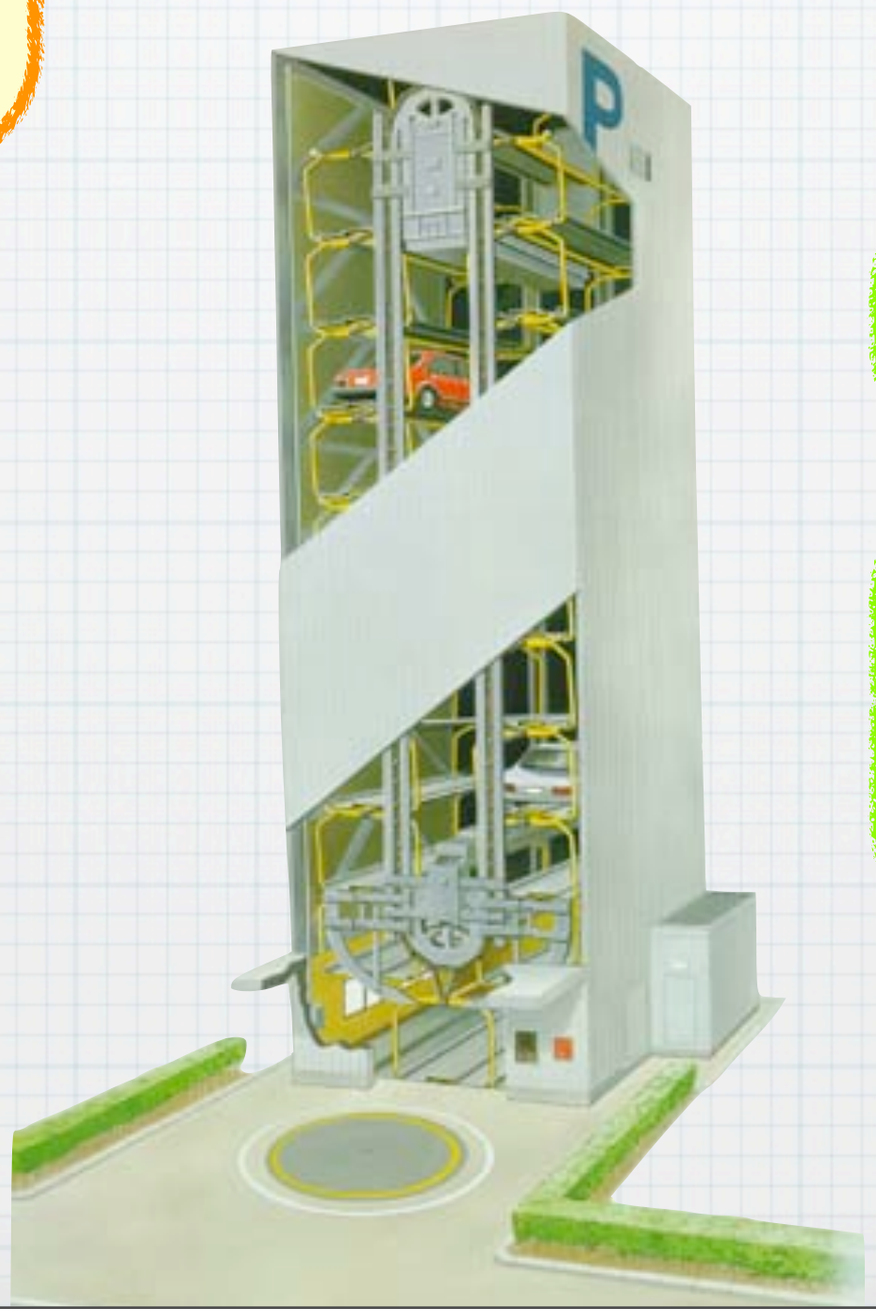
ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ

オレが責任持つっす！

絶対大丈夫っす！

この駐車場,
どうっすか！？



ホントに？ なんで？

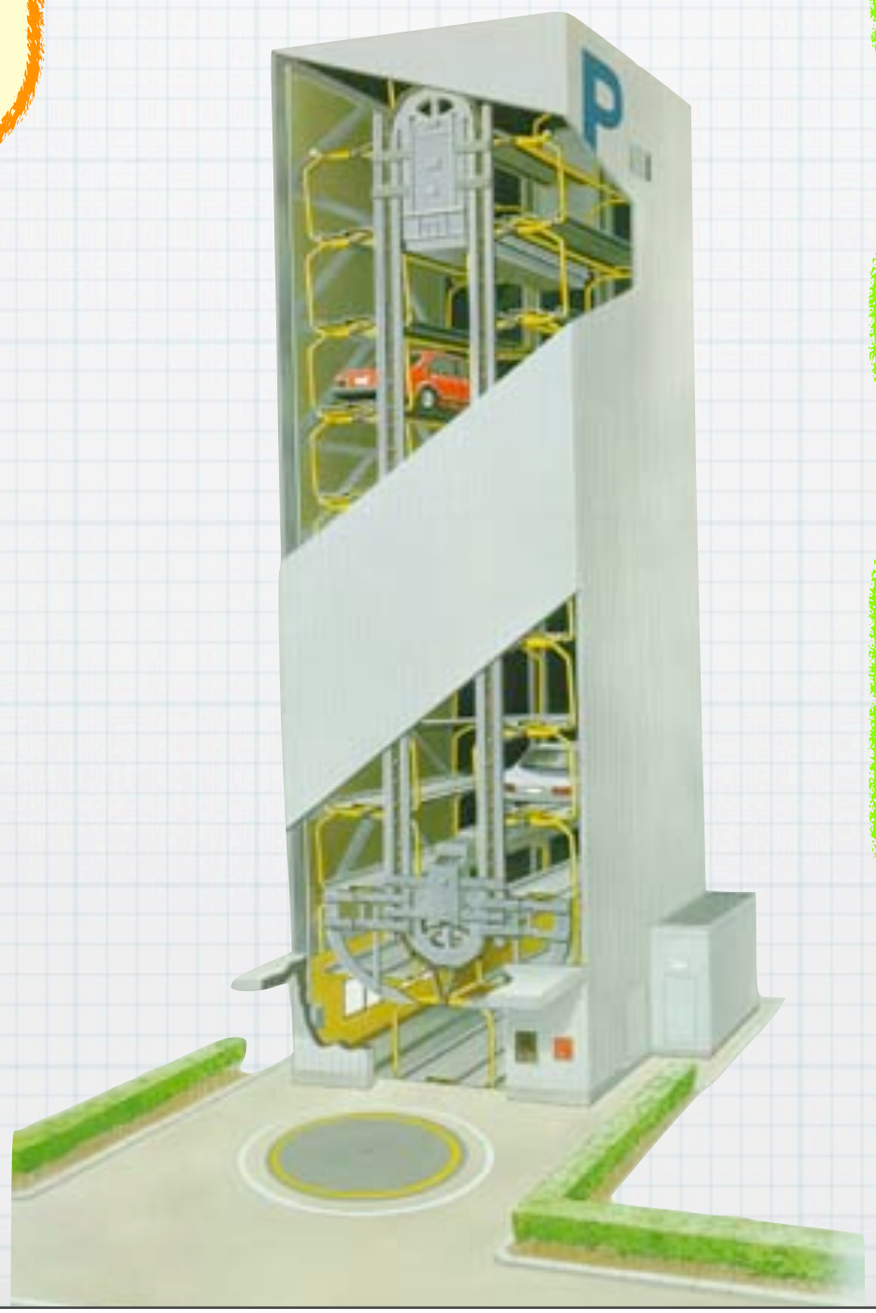
ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ

オレが責任持つっす！

絶対大丈夫っす！

この駐車場,
どうっすか！？

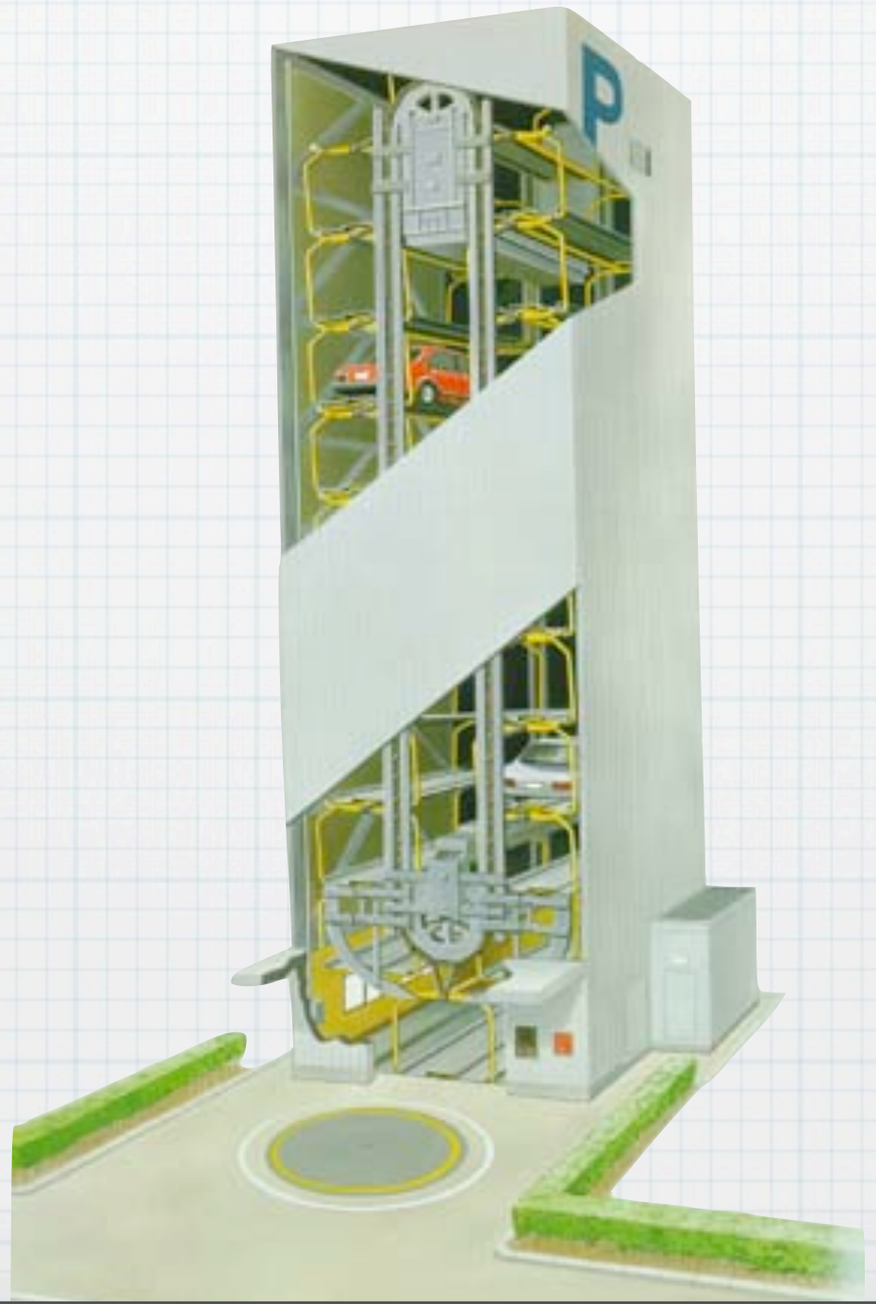


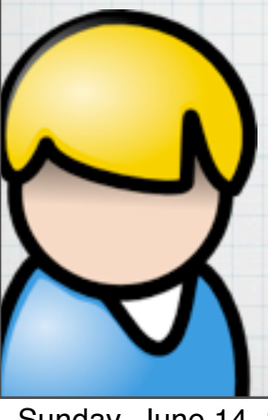
・・・
(ダメだこいつ)

ホントに？ なんで？

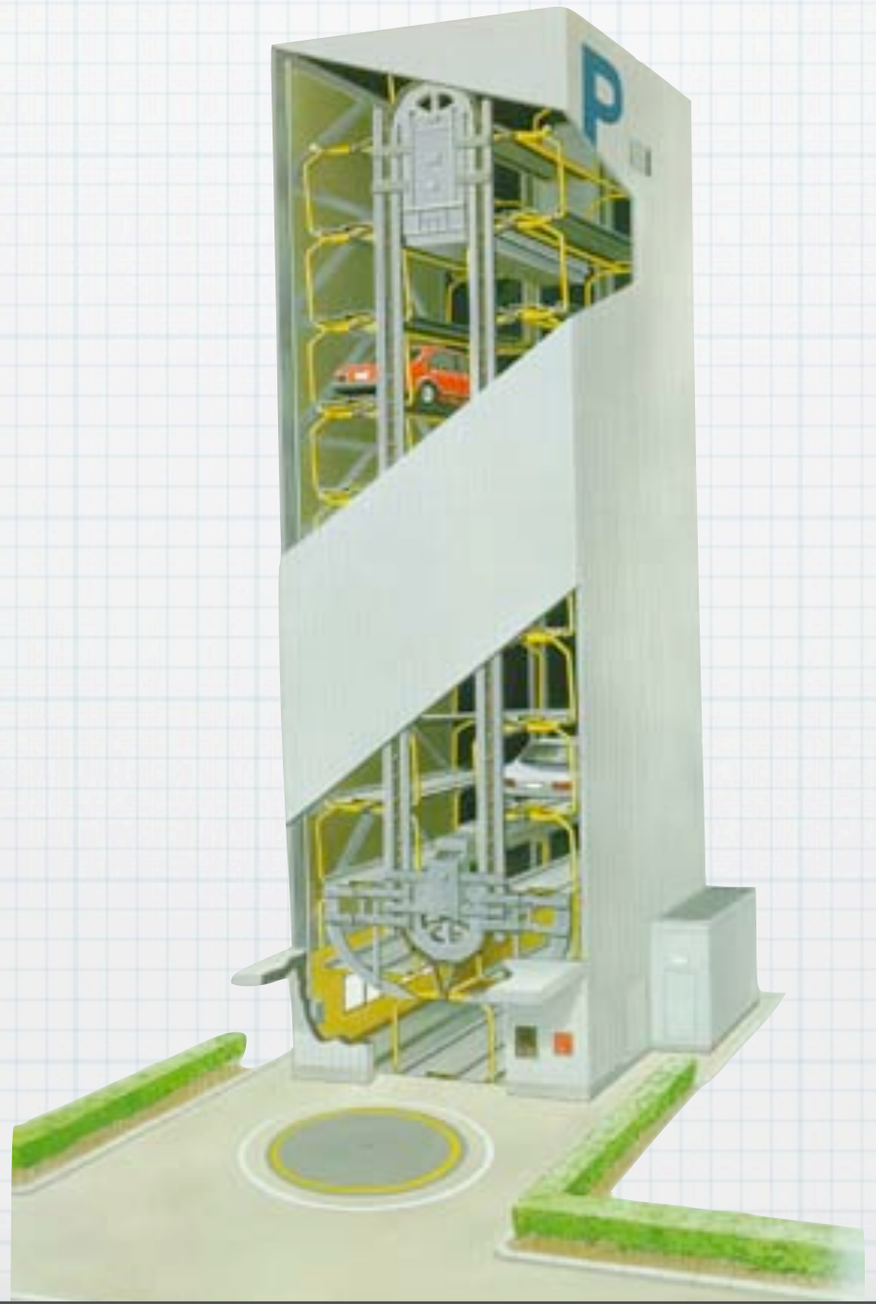
ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ



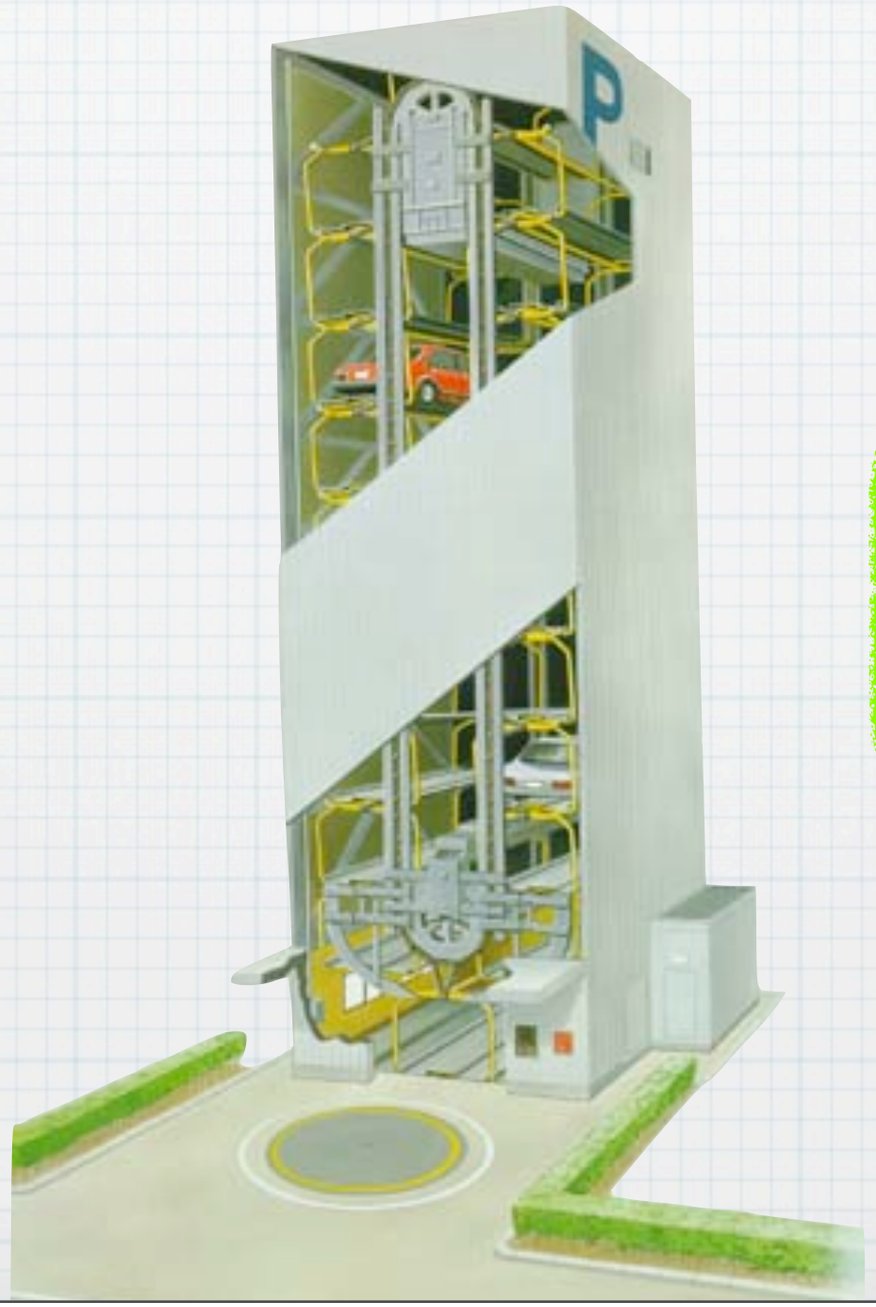


証明で金もうけ



この駐車場,
どうでしょう.

証明で金もうけ



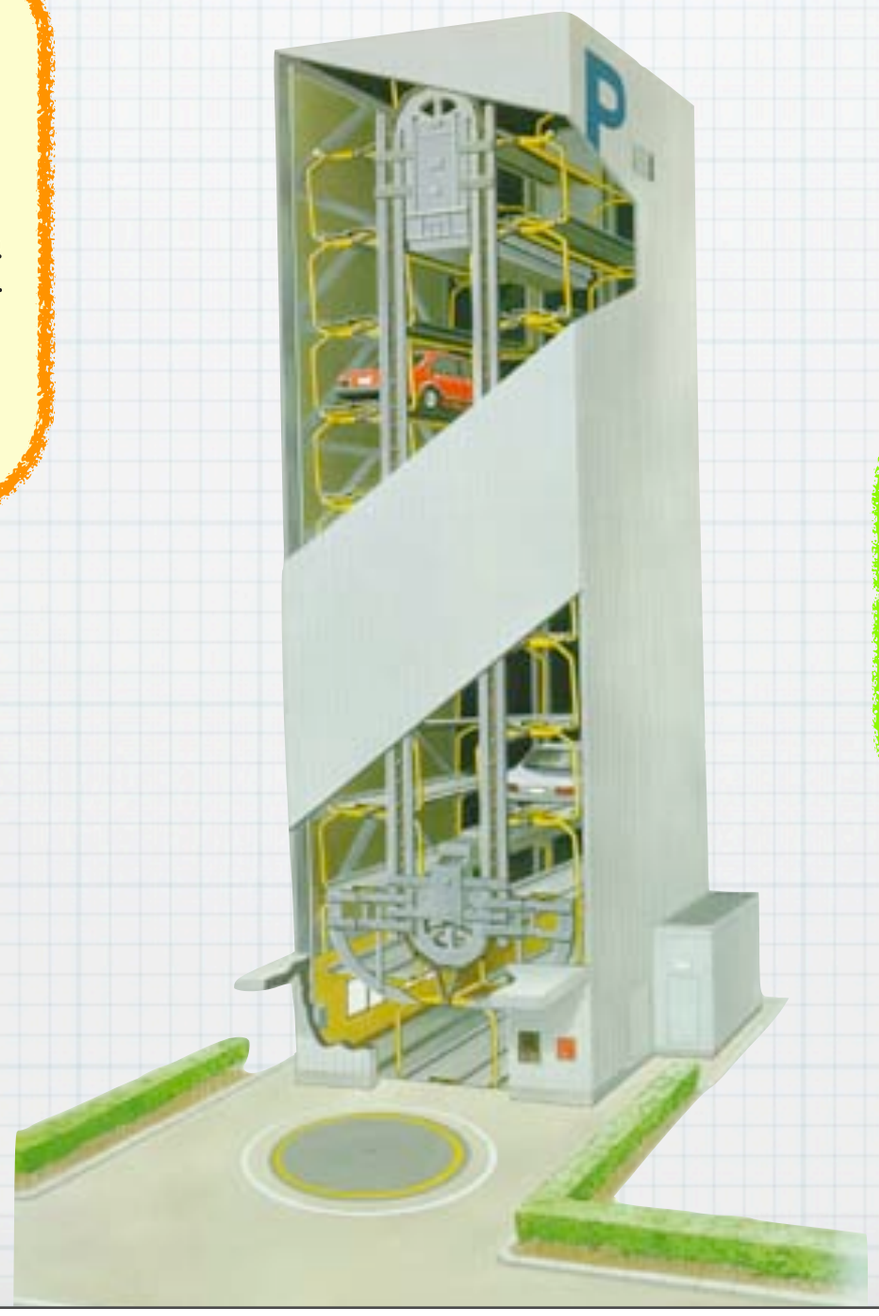
この駐車場,
どうでしょう.

ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ

はい，大丈夫です．
なぜなら，任意の状態 s に対して，ゴンドラ g_1 の位置を x_1 とすると，．．．

この駐車場，
どうでしょう．

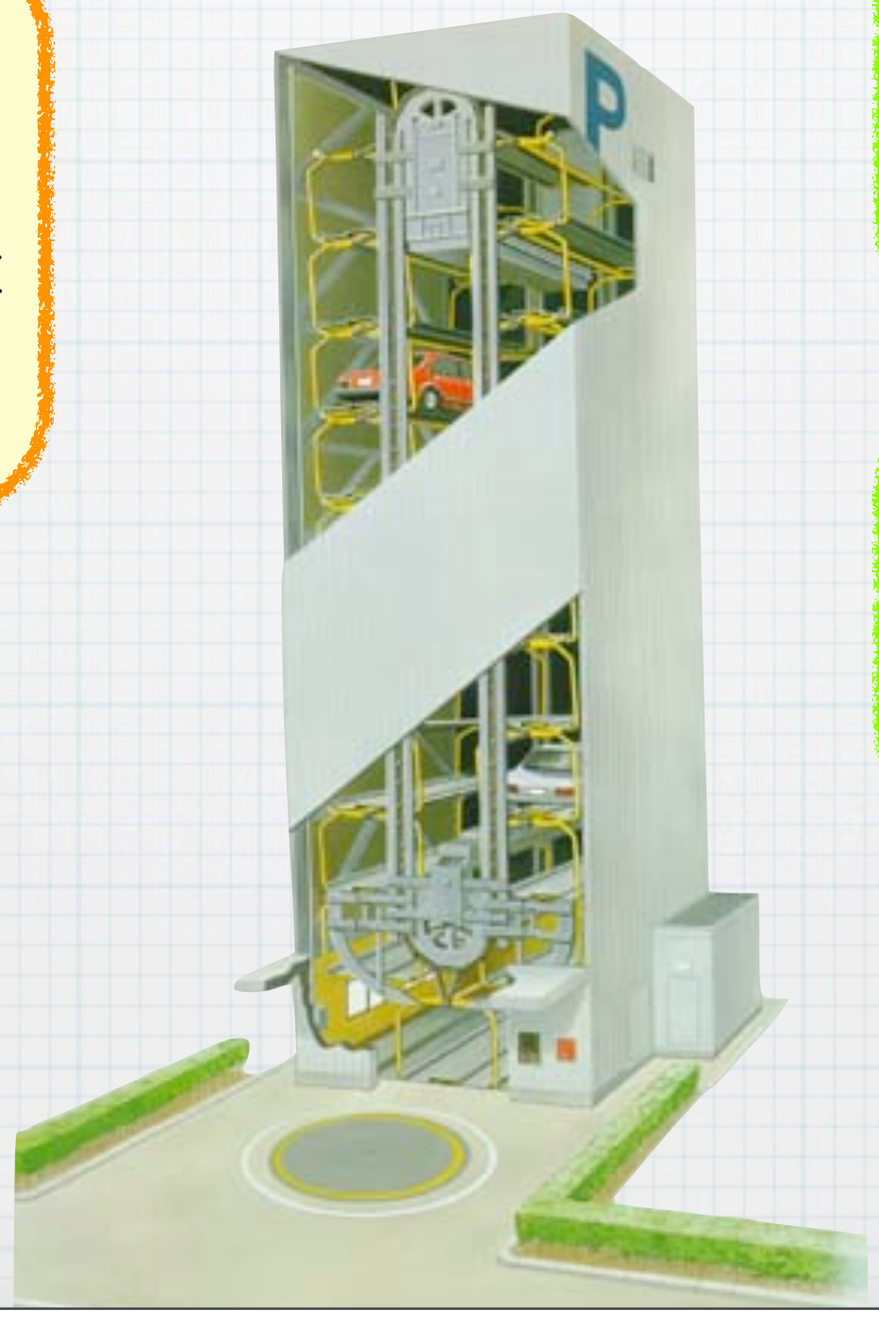


ゴンドラが衝突したりしない？ 大丈夫？

証明で金もうけ

はい，大丈夫です．
なぜなら，任意の状態 s に対して，ゴンドラ g_1 の位置を x_1 とすると，．．．

この駐車場，
どうでしょう．



．．．（読んでる）
なるほど．それでは1基いただこう．

ゴンドラが衝突したりしない？ 大丈夫？

形式検証

* システムが

* 仕様通り動作することを

* 数学的に証明すること.

形式検証

* システムが
* 仕様通り動作することを
* 数学的に証明すること.

形式検証

* システムが

* 仕様通り動作することを

仕様 (specification):
システムの満たすべき性質.
「ゴンドラがぶつからない」
など.

* 数学的に証明すること.

形式検証

* システムが

* 仕様通り動作することを

仕様 (specification):
システムの満たすべき性質.
「ゴンドラがぶつからない」
など.

* 数学的に証明すること.

Proof. By induction on the derivation. We only present the case; the other cases are all similar. Let $J \in {}^*\mathbf{SVarEnv}$ a ${}^*\mathbf{NdEnv}$ be such that $J \models \Gamma$ and $K \models \Delta$.

We consider the i-th section of the whole rule instance of (It results in the following.

$$\Delta|_i; \Gamma|_i \vdash e_1|_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_1|_i\}$$
$$\Delta|_i; \Gamma|_i \vdash e_2|_i : \prod_{v \in \mathbb{N}} \{u \in \mathbb{C} \mid P_2|_i\}$$
$$\models \forall v \in \mathbb{N}. \forall u \in \mathbb{C}. \big((v < r(i+1) \wedge P_1|_i \Rightarrow P|_i) \wedge$$
$$(v \geq r(i+1) \wedge P_2|_i[(v - \lceil r(i+1) \rceil)/v] \Rightarrow P|$$

形式検証

Formal Verification

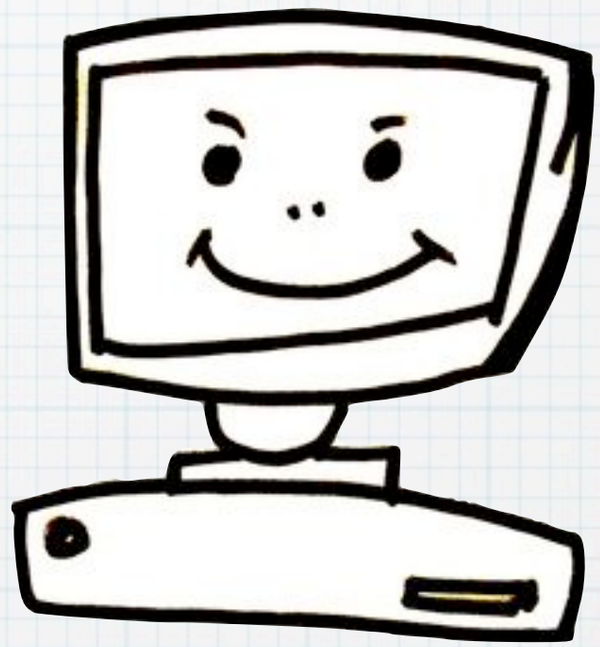
* システム検証 (system verification), 形式手法 (formal methods), ...
* もはや研究者の toy ではない
* デバイスドライバの自動検証（Microsoft）
* 航空機制御ソフトウェアの検証（Airbus）
* ...

システム検証

システム検証

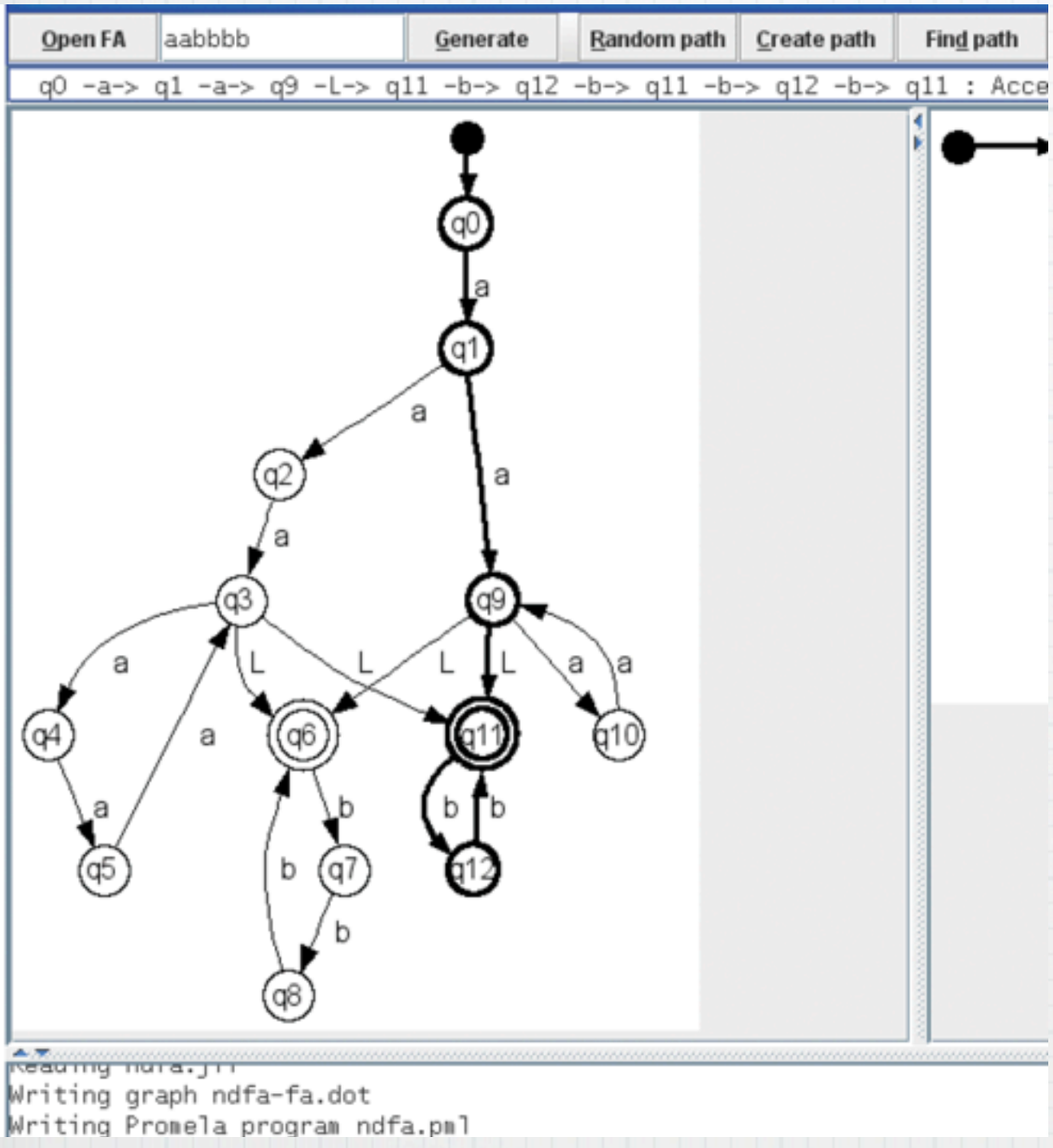
* つまらなくて，間違いやすい証明

システム検証



* つまらなくて，間違いやすい証明
* ➜ 計算機による支援，
  できれば自動化

モデル検査



M. Ben-Ari, ACM Inroads, 2010

* 全状態を総当りでチェック
* SPIN, PRISM, Uppaal, MCRL2, ...

\+ 全自動

− 状態数爆発

− 「∀サイズ」の検証

自動定理証明

ゴンドラが衝突
したりしない？

任意の状態 s に対して，
ゴンドラ g_1 の位置を x_1 と
すると，．．．

* 証明を全自動で書く！
* 応用分野をはっきり決めて，そこに特化
* KeY, KeYmaera, Hoare^{dt} Analyzer, ...

＋ 全自動
＋「∀サイズ」の検証
－ スケーラビリティ

対話型定理証明

* 人間が証明をコンピュータで書く.
* Coq, Agda, PVS, ACL2, ...

\+ 「∀サイズ」の検証
\+ フレキシビリティ
− 人的コスト
（莫大！）

システム品質保証の「スペクトル」

システム品質保証の「スペクトル」

テスト test

検証 verification

- モデル検査
- 自動定理証明
- 対話定理証明

システム品質保証の「スペクトル」

システム品質保証の「スペクトル」

産業界で主流

テスト test

検証 verification

モデル検査

自動定理証明

対話定理証明

低 コスト 高

低 品質保証の度合 高

システム品質保証の「スペクトル」

産業界で主流

総当りで検査

テスト test

検証 verification

モデル検査

自動定理証明

対話定理証明

低 コスト 高

低 品質保証の度合 高

システム品質保証の「スペクトル」

産業界で主流

総当りで検査

テスト test

検証 verification

モデル検査

自動定理証明

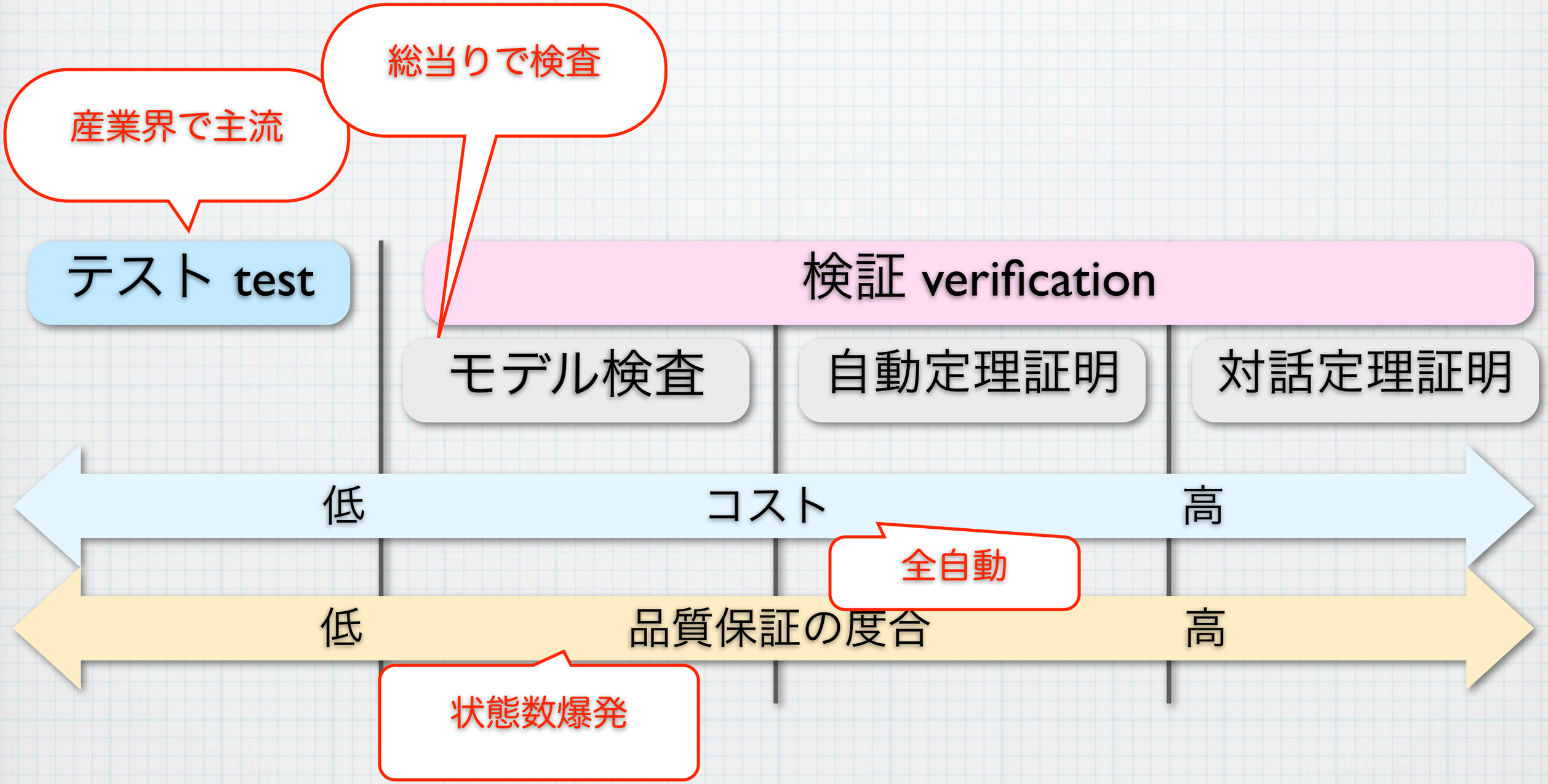
対話定理証明

低 コスト 高

全自動

低 品質保証の度合 高

システム品質保証の「スペクトル」



システム品質保証の「スペクトル」

産業界で主流

総当りで検査

テスト test

検証 verification

モデル検査

自動定理証明

対話定理証明

低 コスト 高

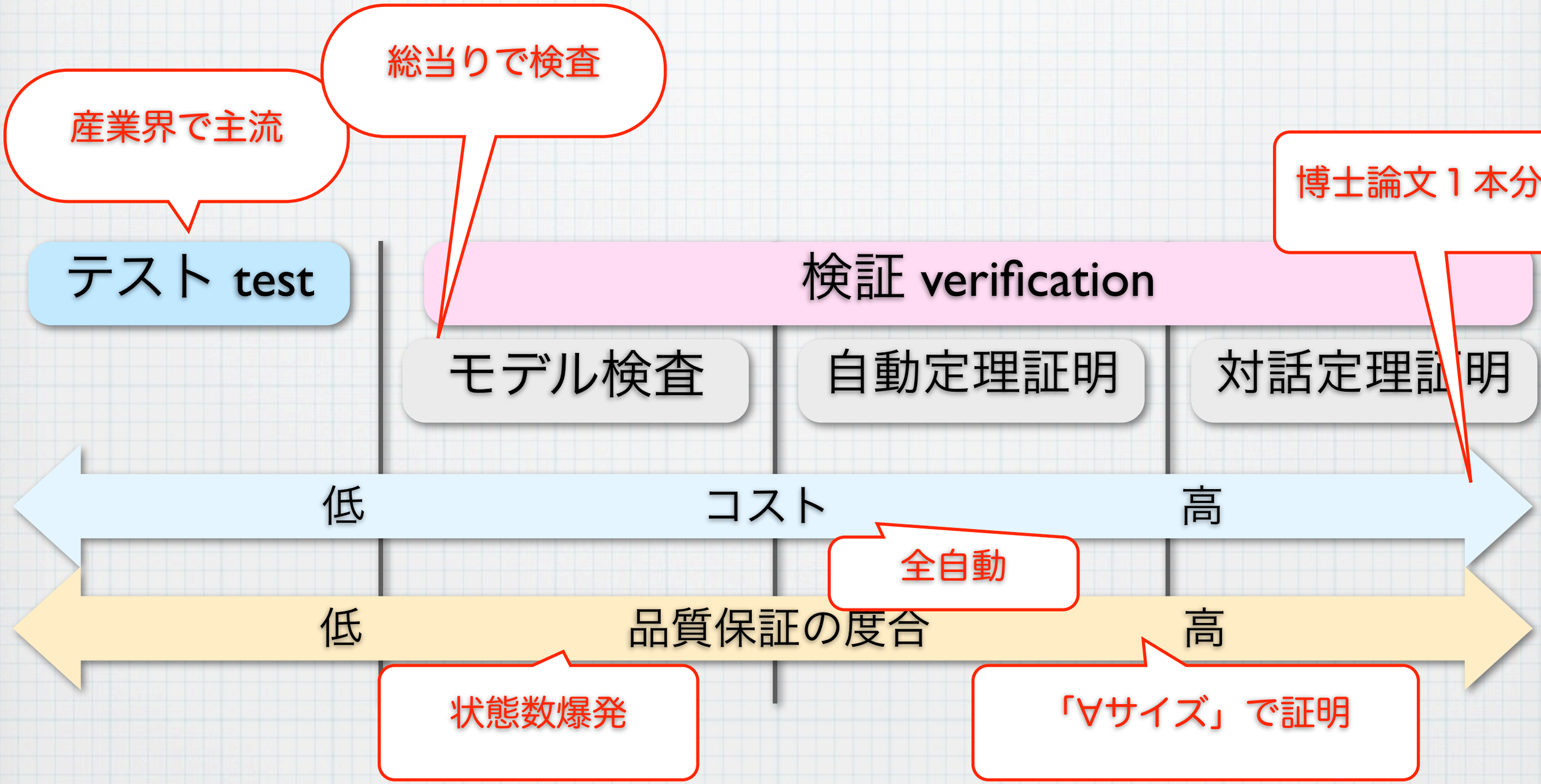
全自動

低 品質保証の度合 高

状態数爆発

「∀サイズ」で証明

システム品質保証の「スペクトル」



Hoare 論理

による形式検証

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

* [Hoare, 1969]

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

* [Hoare, 1969]
* 「システム」 = （命令形）while プログラム
「仕様」= precondition & postcondition

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

* [Hoare, 1969]
* 「システム」 = （命令形）while プログラム
  「仕様」= precondition & postcondition
* Hoare triple を導いていく体系

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

* [Hoare, 1969]
* 「システム」 = （命令形）while プログラム
  「仕様」= precondition & postcondition
* Hoare triple を導いていく体系

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

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

* [Hoare, 1969]
* 「システム」 = （命令形）while プログラム
  「仕様」= precondition & postcondition
* Hoare triple を導いていく体系

{A} P {B}

実行前に成り立つ性質
“precondition”

プログラム

実行後に成り立つ性質
“postcondition”

Hoare 論理

Sir Antony Hoare
(1934.1.11-)
Microsoft Research, Cambridge

* [Hoare, 1969]
* 「システム」 = （命令形）while プログラム
  「仕様」= precondition & postcondition
* Hoare triple を導いていく体系

{ A } P { B }

例：{ n=2 } n:=n+1 { n=3 }

実行前に成り立つ性質
“precondition”

プログラム

実行後に成り立つ性質
“postcondition”

(真である)
Hoare triple の例

$$\{\ k=1 \wedge n=N\ \}$$

while (n>0)
k:=k*n;
n:=n-1

$$\{\ k = N!\ \}$$

Hoare 論理による導出の例

$$\frac{\{\ k*n*((n-1)!)=N! \wedge n-1\geq 0\ \}\ k:=k*n\ \{\ k*((n-1)!)=N! \wedge n-1\geq 0\ \}\quad (\text{Assign}) \qquad \{\ k*((n-1)!)=N! \wedge n-1\geq 0\ \}\ n:=n-1\ \{\ k*(n!)=N! \wedge n\geq 0\ \}\quad (\text{Assign})}{\{\ k*n*((n-1)!)=N! \wedge n-1\geq 0\ \}\ k:=k*n;\ n:=n-1\ \{\ k*(n!)=N! \wedge n\geq 0\ \}}\ (\text{SeqComp})$$

$$\frac{k*(n!)=N! \wedge n\geq 0 \wedge n>0 \Rightarrow k*n*((n-1)!)=N! \wedge n-1\geq 0 \qquad \{\ k*n*((n-1)!)=N! \wedge n-1\geq 0\ \}\ k:=k*n;\ n:=n-1\ \{\ k*(n!)=N! \wedge n\geq 0\ \}}{\{\ k*(n!)=N! \wedge n\geq 0 \wedge n>0\ \}\ k:=k*n;\ n:=n-1\ \{\ k*(n!)=N! \wedge n\geq 0\ \}}\ (\text{Conseq})$$

$$\frac{k=1 \wedge n=N \Rightarrow k*(n!)=N! \wedge n\geq 0 \qquad \{\ k*(n!)=N! \wedge n\geq 0\ \}\ \text{while } (n>0)\ k:=k*n;\ n:=n-1\ \{\ k*(n!)=N! \wedge n\geq 0 \wedge \neg(n>0)\ \}}{\{\ k=1 \wedge n=N\ \}\ \text{while } (n>0)\ k:=k*n;\ n:=n-1\ \{\ k*(n!)=N! \Rightarrow k=N!\ \}}\ (\text{While})\ (\text{Conseq})$$

Hoare 論理の「材料」

* プログラム意味論
  * プログラム ＝ メモリ状態の変換

* メモリ状態の性質を記述するための
  assertion language

* Hoare triple を導くための
  導出規則（ルール），soundness

プログラム意味論

「意味論」とは？

- プログラムの「意味」は何か
  - 正確な答え：
    「実行した際の MacBook 内部の電圧変化」
  - ➜ 細かすぎて「使えない」

```
n := N;
k := 1;
while (n > 0) {
   k := k*n;
   n := n-1;
}
```

- ここではメモリ状態を用いる
  - プログラミング言語による
    （命令型言語だからメモリ状態を使う．たとえば関数型言語ならば関数として意味をつける）

メモリ状態

* 変数と値の対応の表のこと.

$$\begin{bmatrix} \mathrm{x} & \mapsto & 2 \\ \mathrm{y} & \mapsto & 13 \\ & \vdots & \end{bmatrix}$$

* 数学的には： 関数

$$\sigma : \mathbf{Var} \longrightarrow \mathbb{Z}$$

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\mathrm{x} := a]\!] : \quad \sigma \quad \longmapsto \quad \sigma\left[\,\mathrm{x} \mapsto [\![a]\!]\sigma\,\right]$$

x は変数，a は「数の表現」
（たとえば y+1）

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\mathrm{x} := a]\!] : \quad \sigma \quad \longmapsto \quad \sigma\left[\,\mathrm{x} \mapsto [\![a]\!]\sigma\,\right]$$

プログラム x:=a の
「意味」

x は変数，a は「数の表現」
（たとえば y+1）

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\mathrm{x} := a]\!] : \quad \sigma \quad \longmapsto \quad \sigma\left[\,\mathrm{x} \mapsto [\![a]\!]\sigma\,\right]$$

プログラム x:=a の「意味」

メモリ状態，たとえば

$$\begin{bmatrix} \mathrm{x} & \mapsto & 2 \\ \mathrm{y} & \mapsto & 13 \\ & \vdots & \end{bmatrix}$$

x は変数，a は「数の表現」（たとえば y+1）

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\mathrm{x} := a]\!] : \sigma \longmapsto \sigma\left[\mathrm{x} \mapsto [\![a]\!]\sigma\right]$$

プログラム x:=a の「意味」

x は変数，a は「数の表現」（たとえば y+1）

メモリ状態，たとえば

$$\begin{bmatrix} \mathrm{x} & \mapsto & 2 \\ \mathrm{y} & \mapsto & 13 \\ & \vdots & \end{bmatrix}$$

アップデートされたメモリ状態．x の値を，a を σ のもとで計算した値（たとえば

$$[\![\mathrm{y}+1]\!]\sigma = 14$$

とか）にアップデート

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![P_1; P_2]\!] : \quad \sigma \quad \longmapsto \quad [\![P_2]\!]([\![P_1]\!]\sigma)$$

P_1 ，P_2 はプログラム

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![P_1; P_2]\!] : \quad \sigma \quad \longmapsto \quad [\![P_2]\!]([\![P_1]\!]\sigma)$$

まず P_1 によって変換

P_1 ， P_2 はプログラム

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![P_1; P_2]\!] : \quad \sigma \quad \longmapsto \quad [\![P_2]\!]([\![P_1]\!]\sigma)$$

まず P_1 によって変換

次に P_2 によって変換

P_1 ，P_2 はプログラム

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\texttt{if}\ b\ \texttt{then}\ P_1\ \texttt{else}\ P_2]\!] :$$

$$\sigma \quad \longmapsto \quad \begin{cases} [\![P_1]\!]\sigma & \text{if } [\![b]\!]\sigma \text{ is true} \\ [\![P_2]\!]\sigma & \text{if } [\![b]\!]\sigma \text{ is false} \end{cases}$$

P_1，P_2 はプログラム
b は「真偽表現」
（x > 0 とか）

プログラムの意味

$$\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$$

$$[\![\texttt{while}\, b\, P]\!] : \quad \sigma \quad \longmapsto \quad ??$$

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\texttt{while}\, b\, P]\!] : \quad \sigma \quad \longmapsto \quad ??$$

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\texttt{while}\, b\, P]\!] : \quad \sigma \quad \longmapsto \quad ??$$

* 状態変換の繰り返し $[\![P]\!]^n\sigma$を考える：

$$\sigma,\ [\![P]\!]\sigma,\quad [\![P]\!]([\![P]\!]\sigma),\quad [\![P]\!]([\![P]\!]([\![P]\!]\sigma)),\quad \ldots$$

  * ある時点で b が偽になれば，つまり $[\![b]\!]([\![P]\!]^n\sigma) = \mathrm{false}$ なら，そうなるような最初の n に対する $[\![P]\!]^n\sigma$ を返す
  * b がずっと真であれば，⊥（未定義，非停止）

プログラムの意味

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

「停止しない」

$$[\![\texttt{while}\, b\, P]\!] : \quad \sigma \quad \longmapsto \quad ??$$

* 状態変換の繰り返し $[\![P]\!]^n\sigma$を考える：

$$\sigma,\ [\![P]\!]\sigma,\quad [\![P]\!]([\![P]\!]\sigma),\quad [\![P]\!]([\![P]\!]([\![P]\!]\sigma)),\quad \ldots$$

  * ある時点で b が偽になれば，つまり $[\![b]\!]([\![P]\!]^n\sigma) = \mathrm{false}$ なら，そうなるような最初の n に対する $[\![P]\!]^n\sigma$ を返す
  * b がずっと真であれば，⊥（未定義，非停止）

プログラムの意味論：まとめ

「停止しない」

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\mathrm{x} := a]\!] : \quad \sigma \quad \longmapsto \quad \sigma\left[\mathrm{x} \mapsto [\![a]\!]\sigma\right]$$

$$[\![P_1; P_2]\!] : \quad \sigma \quad \longmapsto \quad [\![P_2]\!]\big([\![P_1]\!]\sigma\big)$$

$$[\![\texttt{if}\ b\ \texttt{then}\ P_1\ \texttt{else}\ P_2]\!] :$$

$$\sigma \quad \longmapsto \quad \begin{cases} [\![P_1]\!]\sigma & \text{if } [\![b]\!]\sigma \text{ is true} \\ [\![P_2]\!]\sigma & \text{if } [\![b]\!]\sigma \text{ is false} \end{cases}$$

$$[\![\texttt{while}\ b\ P]\!] : \quad \sigma \quad \longmapsto \quad \ldots$$

プログラムの意味論：まとめ

「停止しない」

* メモリ状態の変換として
  * つまり，関数 $\mathbf{MSt} \longrightarrow \mathbf{MSt} \cup \{\bot\}$

$$[\![\mathrm{x} := a]\!] : \quad \sigma \quad \longmapsto \quad \sigma\left[\mathrm{x} \mapsto [\![a]\!]\sigma\right]$$

$$[\![P_1; P_2]\!] : \quad \sigma \quad \longmapsto \quad [\![P_2]\!]\big([\![P_1]\!]\sigma\big)$$

$$[\![\texttt{if}\ b\ \texttt{then}\ P_1\ \texttt{else}\ P_2]\!] :$$

$$\sigma \quad \longmapsto \quad \begin{cases} [\![P_1]\!]\sigma & \text{if } [\![b]\!]\sigma \text{ is true} \\ [\![P_2]\!]\sigma & \text{if } [\![b]\!]\sigma \text{ is false} \end{cases}$$

$$[\![\texttt{while}\ b\ P]\!] : \quad \sigma \quad \longmapsto \quad \ldots$$

ポイント：

* 数学的に厳密な定義
* 要素還元的（大きなプログラムの意味は，その部品の意味から決まる）

Hoare 論理の「材料」

* プログラム意味論
  * プログラム ＝ メモリ状態の変換

* メモリ状態の性質を記述するための
  assertion language

* Hoare triple を導くための
  導出規則（ルール）, soundness

Assertion Language

* Assertion: メモリ状態の性質を記述する論理式．例：
  * $x = 5 \wedge y \leq 3$
  * $\exists z. (x = 2 * z \wedge y = 3 * z)$

Assertion Language

* Assertion: メモリ状態の性質を記述する論理式．例：
  * $x = 5 \wedge y \leq 3$
  * $\exists z. (x = 2 * z \wedge y = 3 * z)$

Assertion Language

* Assertion: メモリ状態の性質を記述する論理式．例：
  * $x = 5 \wedge y \leq 3$
  * $\exists z.\ (\ x = 2 * z \wedge y = 3 * z\ )$

* 算術のための一階述語論理を用いる．

$$
\begin{array}{llll}
\mathbf{AExp} \ni & a & ::= & x \mid n \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \\
\mathbf{Fml} \ni & A & ::= & \texttt{true} \mid \texttt{false} \mid A_1 \wedge A_2 \mid \neg A \mid a_1 < a_2 \mid \\
 & & & \forall x \in \mathbb{N}.\, A \qquad \text{where } x \in \mathbf{Var}
\end{array}
$$

  * そうすると相対完全性も成り立つ
  * 自動定理証明器の実装では，命題論理に制限．
    （限量子除去 QE を適宜用いる）

Hoare 論理の「材料」

* プログラム意味論
  * プログラム ＝ メモリ状態の変換

* メモリ状態の性質を記述するための
  assertion language

* Hoare triple を導くための
  導出規則（ルール）, soundness

Hoare 論理の導出規則

$$\frac{}{\{A\}\,\texttt{skip}\,\{A\}}\ (\textsc{Skip})$$

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1;c_2\,\{B\}}\ (\textsc{Seq})$$

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\texttt{while}\ b\ \texttt{do}\ c\,\{A \wedge \neg b\}}\ (\textsc{While})$$

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

$$\frac{\{A \wedge b\}\,c_1\,\{B\} \quad \{A \wedge \neg b\}\,c_2\,\{B\}}{\{A\}\,\texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\,\{B\}}\ (\textsc{If})$$

$$\frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Hoare 論理の導出規則

$$\frac{\{A\}\ P_1\ \{C\} \qquad \{C\}\ P_2\ \{B\}}{\{A\}\ P_1; P_2\ \{B\}}\ \text{(SeqComp)}$$

* 例：

$$\dfrac{\dfrac{}{\{x-1=2\}\ y:=x\ \{y-1=2\}}\ \text{(Assign)} \qquad \dfrac{}{\{y-1=2\}\ y:=y-1\ \{y=2\}}\ \text{(Assign)}}{\{x-1=2\}\ y:=x;\ y:=y-1\ \{y=2\}}\ \text{(SeqComp)}$$

Hoare 論理の導出規則

$$\frac{\{ A \wedge b \}\ P_1\ \{ A \}}{\{ A \}\ \text{while } b\ P_1\ \{ A \wedge \neg b \}}\ \text{(While)}$$

* 例：

$$\frac{\{ x \geq 0 \wedge x > 0 \}\ x := x-1\ \{ x \geq 0 \}}{\{ x \geq 0 \}\ \text{while } x>0\ (\ x := x-1\ )\ \{ x \geq 0 \wedge \neg(x>0) \}}\ \text{(While)}$$

Hoare 論理の導出規則

A はループ不変量！

$$\frac{\{ A \wedge b \}\ P_1\ \{ A \}}{\{ A \}\ \text{while b } P_1\ \{ A \wedge \neg b \}}\ \text{(While)}$$

* 例：

$$\frac{\{ x \geq 0 \wedge x > 0 \}\ x := x-1\ \{ x \geq 0 \}}{\{ x \geq 0 \}\ \text{while } x>0\ (\ x := x-1\ )\ \{ x \geq 0 \wedge \neg(x>0) \}}\ \text{(While)}$$

Hoare 論理の導出規則

A はループ不変量！

$$\frac{\{A \wedge b\}\ P_1\ \{A\}}{\{A\}\ \text{while b } P_1\ \{A \wedge \neg b\}}\ \text{(While)}$$

ループを脱出した ➜
b は成立しないはず

* 例：

$$\frac{\{x \geq 0 \wedge x > 0\}\ x := x-1\ \{x \geq 0\}}{\{x \geq 0\}\ \text{while } x>0\ (\ x := x-1\ )\ \{x \geq 0 \wedge \neg(x>0)\}}\ \text{(While)}$$

Hoare 論理の健全性定理

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

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

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

Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff} \left[\begin{array}{l} \text{for each memory state } \sigma, \\ \sigma \models A \text{ implies } [\![c]\!](\sigma) \models B. \end{array}\right]$$

Hoare 論理の健全性定理

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

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

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

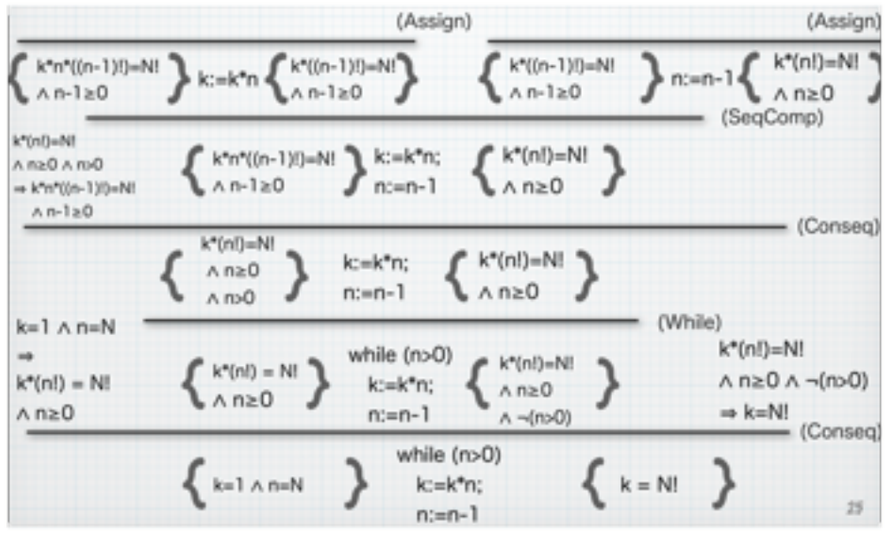
Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff}$$

$$\left[\begin{array}{l} \text{for each memory state } \boldsymbol{\sigma}, \\ \boldsymbol{\sigma} \models A \text{ implies } [\![c]\!](\boldsymbol{\sigma}) \models B. \end{array}\right]$$



プログラム c の「意味」
（メモリ状態の変換として）

ータ科学）

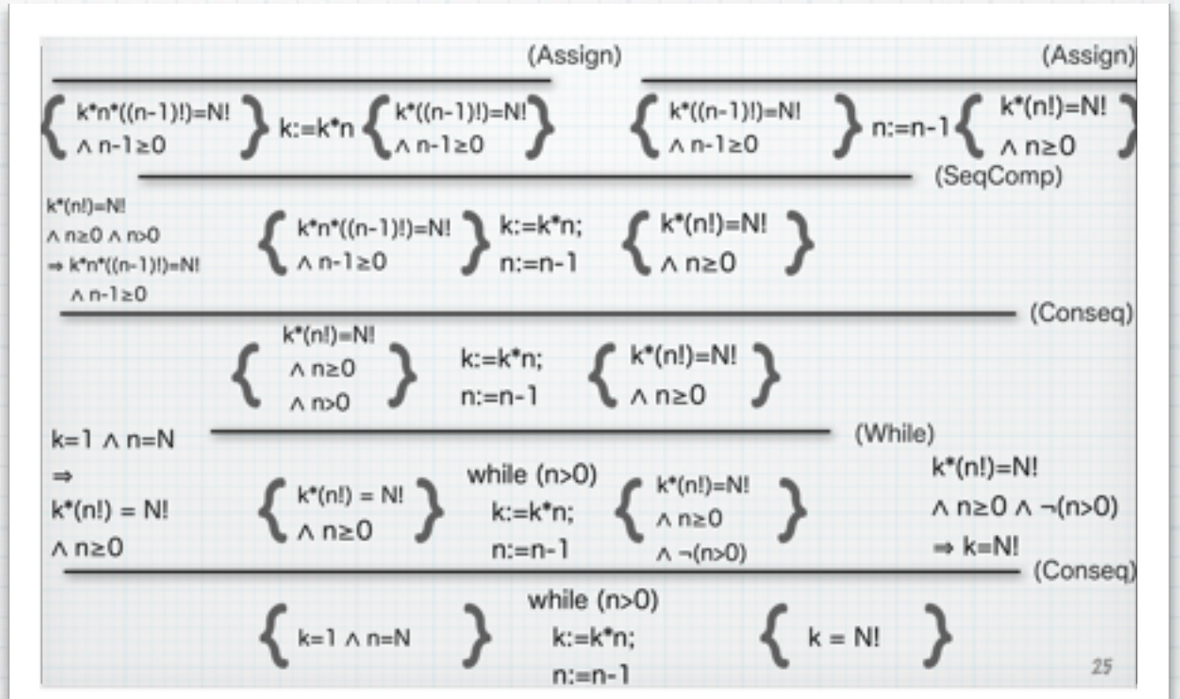
Hoare 論理による形式検証（まとめ）

* システム + 仕様 を Hoare triple `{A} P {B}` として表現．真？

Hoare 論理による形式検証（まとめ）

* システム + 仕様 を Hoare triple {A} P {B} として表現．真？

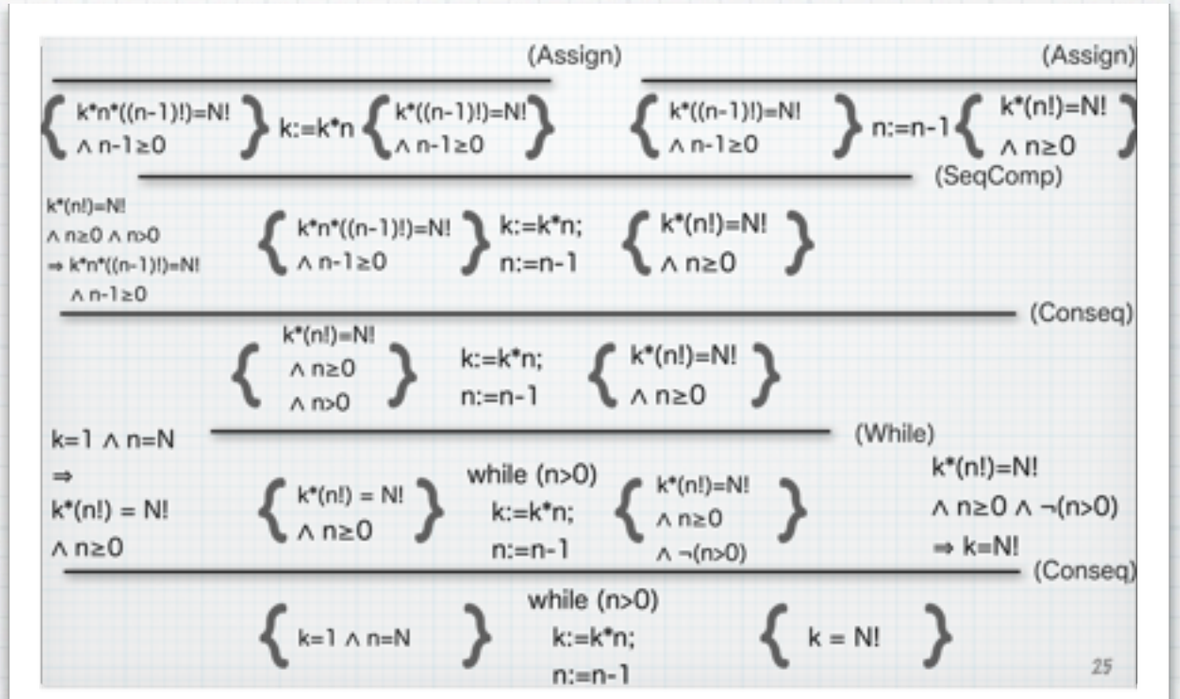
* Hoare 論理で，自動証明（証明検索）



Hoare 論理による形式検証（まとめ）

* システム + 仕様 を Hoare triple
  `{A} P {B}` として表現．真？

* Hoare 論理で，
  自動証明
  （証明検索）



* 記号操作で証明木が書けたら，
  意味論的にも真 （健全性定理による）

超準解析による
物理情報システムの形式検証

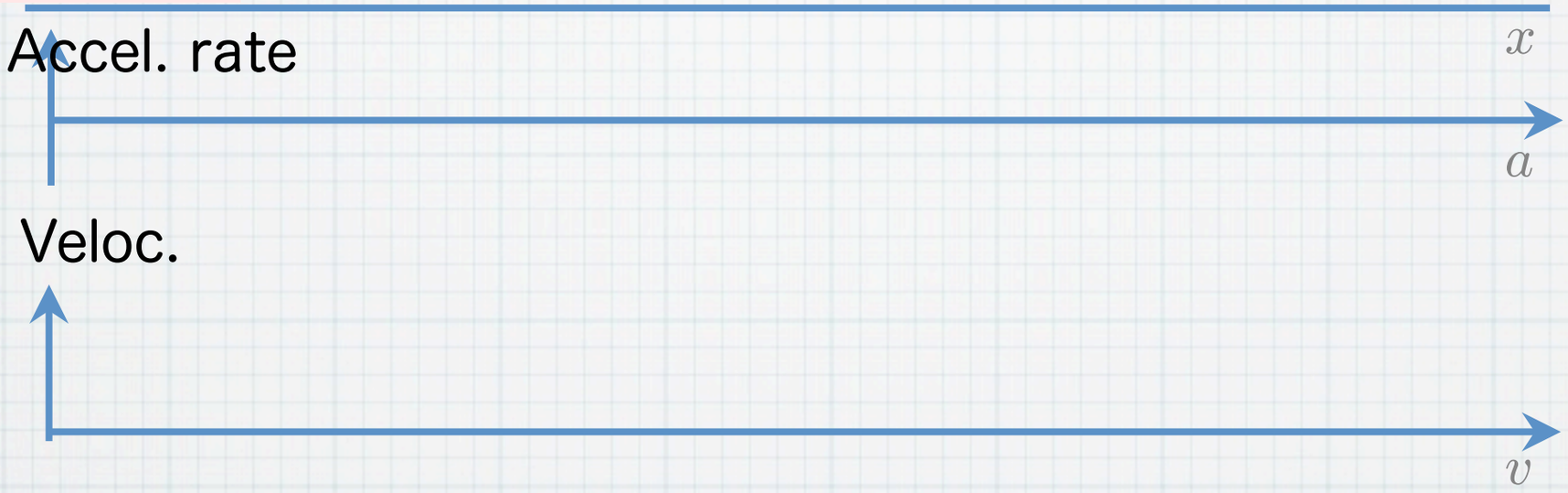
アウトライン

* 形式検証とは？
  * Hoare 論理を例に
* ハイブリッド・システム
  * 離散＋連続
  * 物理情報システムの一側面
* 超準解析による移転
  * 離散的検証手法を，文字通りそのまま
    ハイブリッド・システムに適用

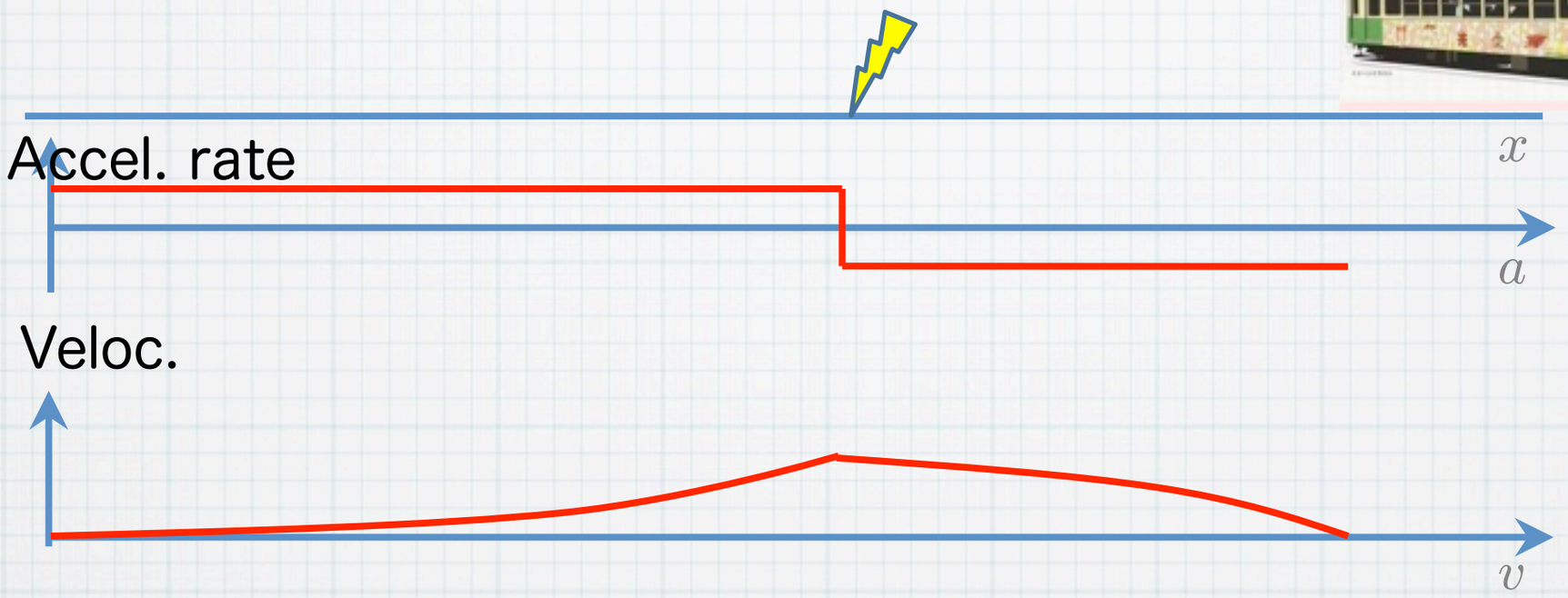
2

ハイブリッド・システム

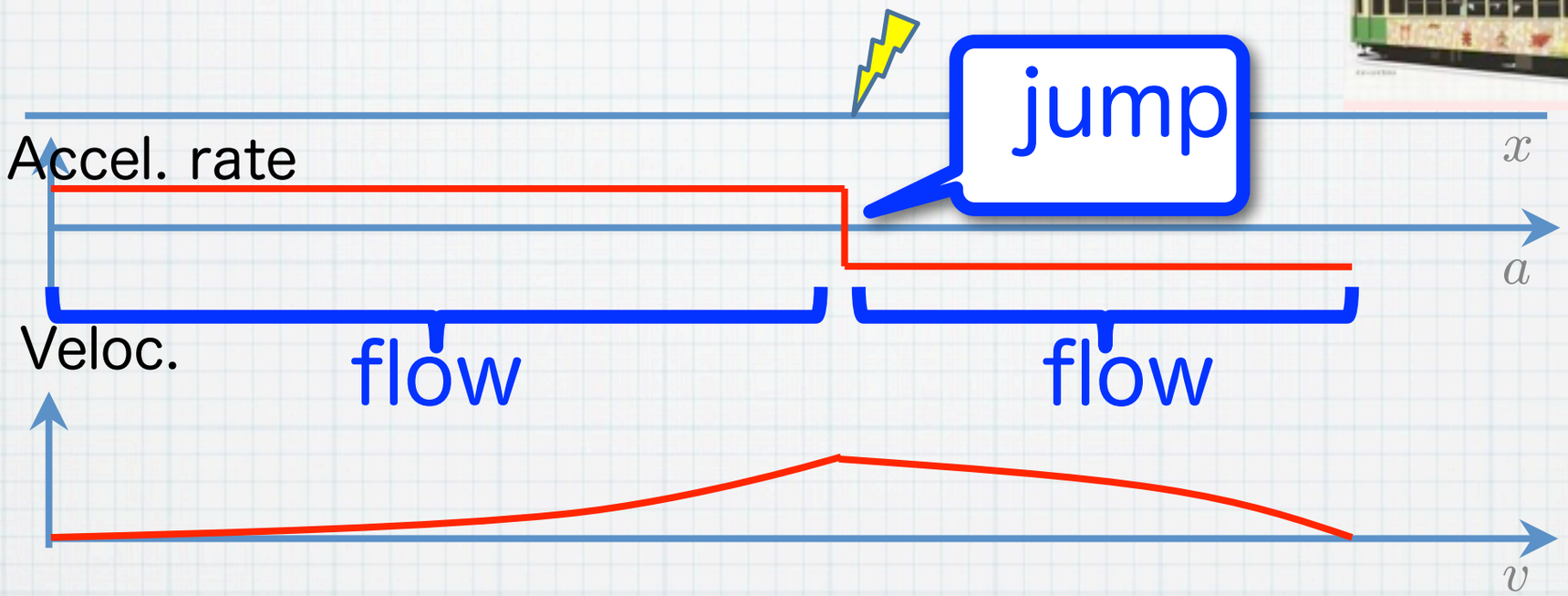
ハイブリッド・システム



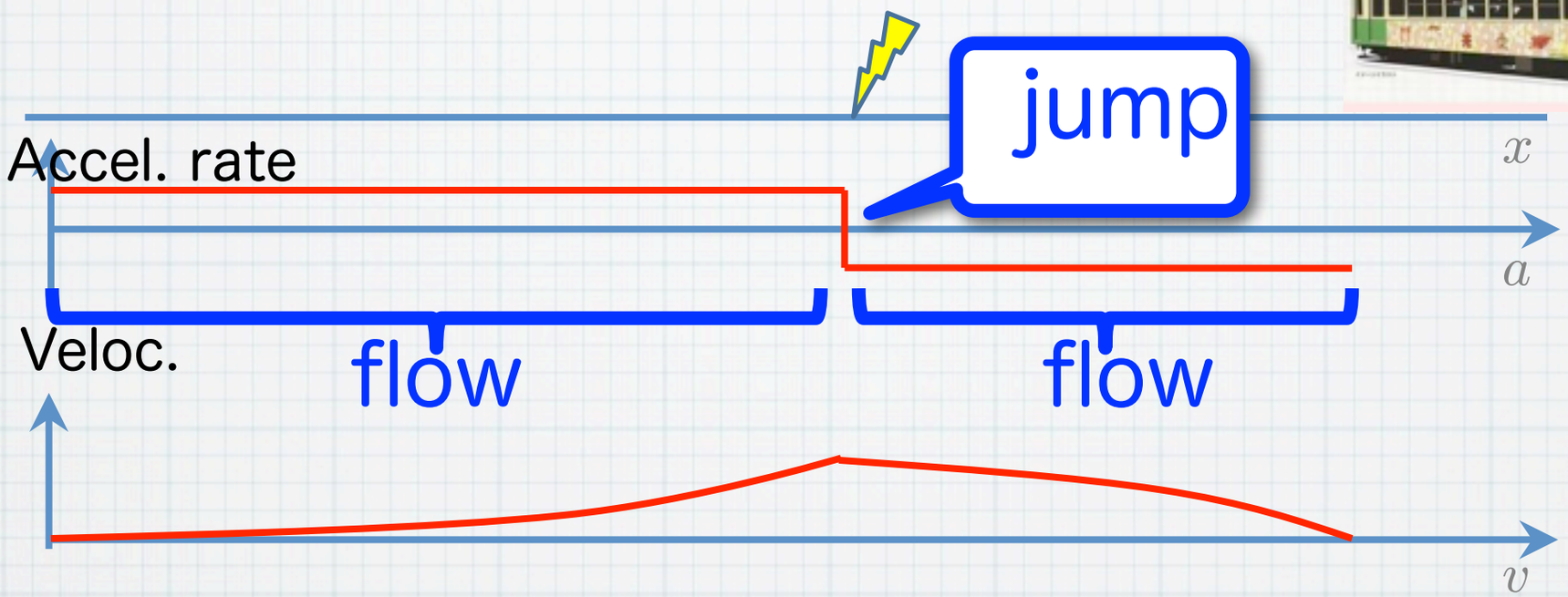
ハイブリッド・システム



ハイブリッド・システム



ハイブリッド・システム



* Flow & jump
  * 物理系におけるデジタル制御
  * 物理情報システムの一側面

物理情報システム CPS: Cyber Physical System

* The term cyber-physical systems refers to the tight conjoining of and coordination between computational and physical resources.
(NSF Program Solicitation, NSF 08-611)

物理情報システム CPS: Cyber Physical System

* The term cyber-physical systems refers to the tight conjoining of and coordination between computational and physical resources.
(NSF Program Solicitation, NSF 08-611)

* 物理情報システム CPS:
  * Real-Time （実時間）
  * Embedded （組み込み）
  * Sensor Network
  * Hybrid System

ハイブリッド・システム

「物理情報システム」
自動車，飛行機，etc.

離散的データ
(Jump)

と

連続的データ
(Flow)

ハイブリッド・システム

「物理情報システム」
自動車，飛行機，etc.

離散的データ
(Jump)

と

連続的データ
(Flow)

ハイブリッド・システム

「物理情報システム」
自動車，飛行機，etc.

形式検証

離散的データ
(Jump)

と

連続的データ
(Flow)

制御理論・
力学系理論

ハイブリッド・システム

「物理情報システム」
自動車，飛行機，etc.

形式検証

「ハイブリッドだ！」

離散的データ
(Jump)

と

連続的データ
(Flow)

「ハイブリッドでしょ！」

制御理論・
力学系理論

ハイブリッド・システム

「物理情報システム」
自動車，飛行機，etc.

離散的データ
(Jump)

と

連続的データ
(Flow)

形式検証

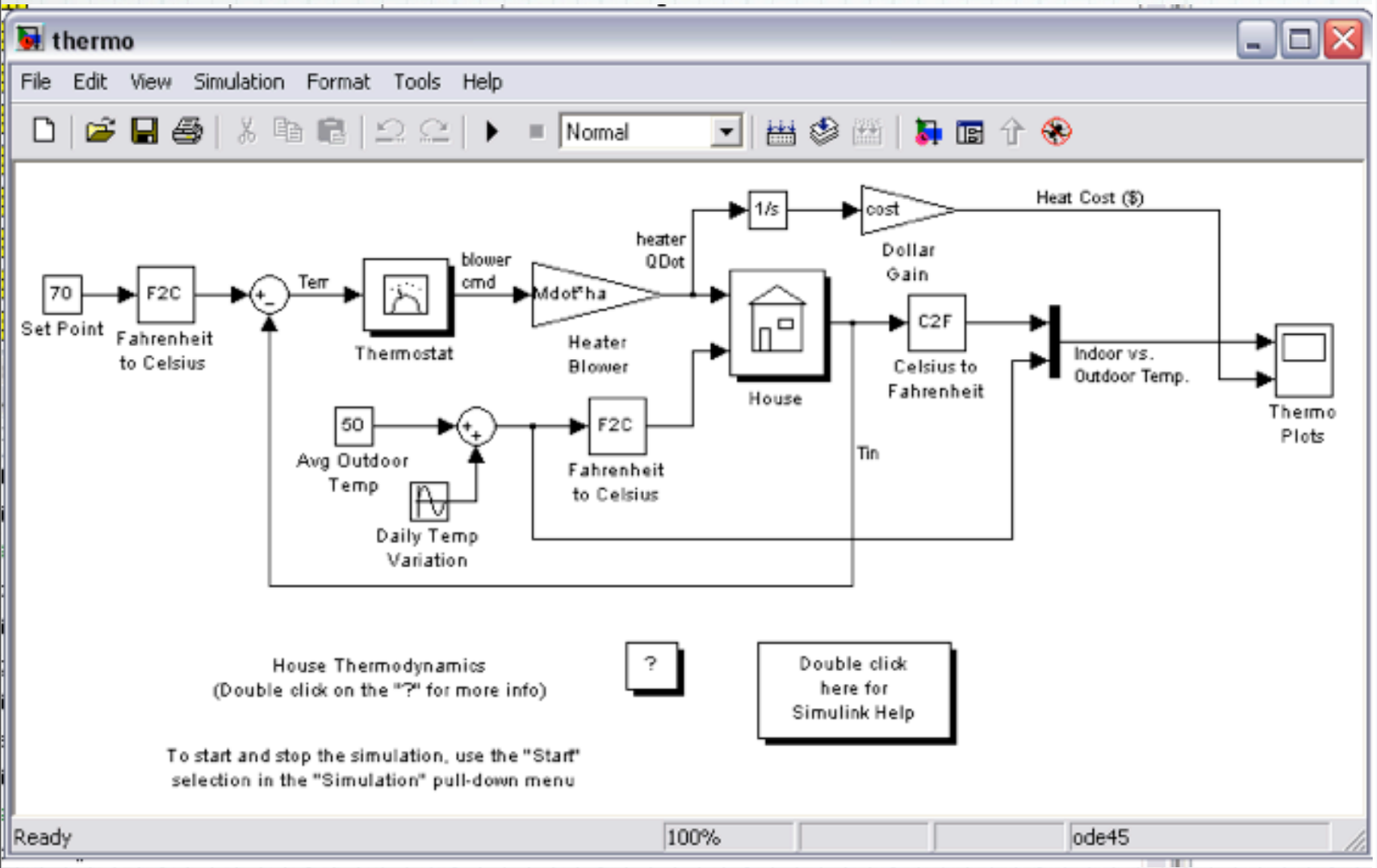
「ハイブリッドだ！」

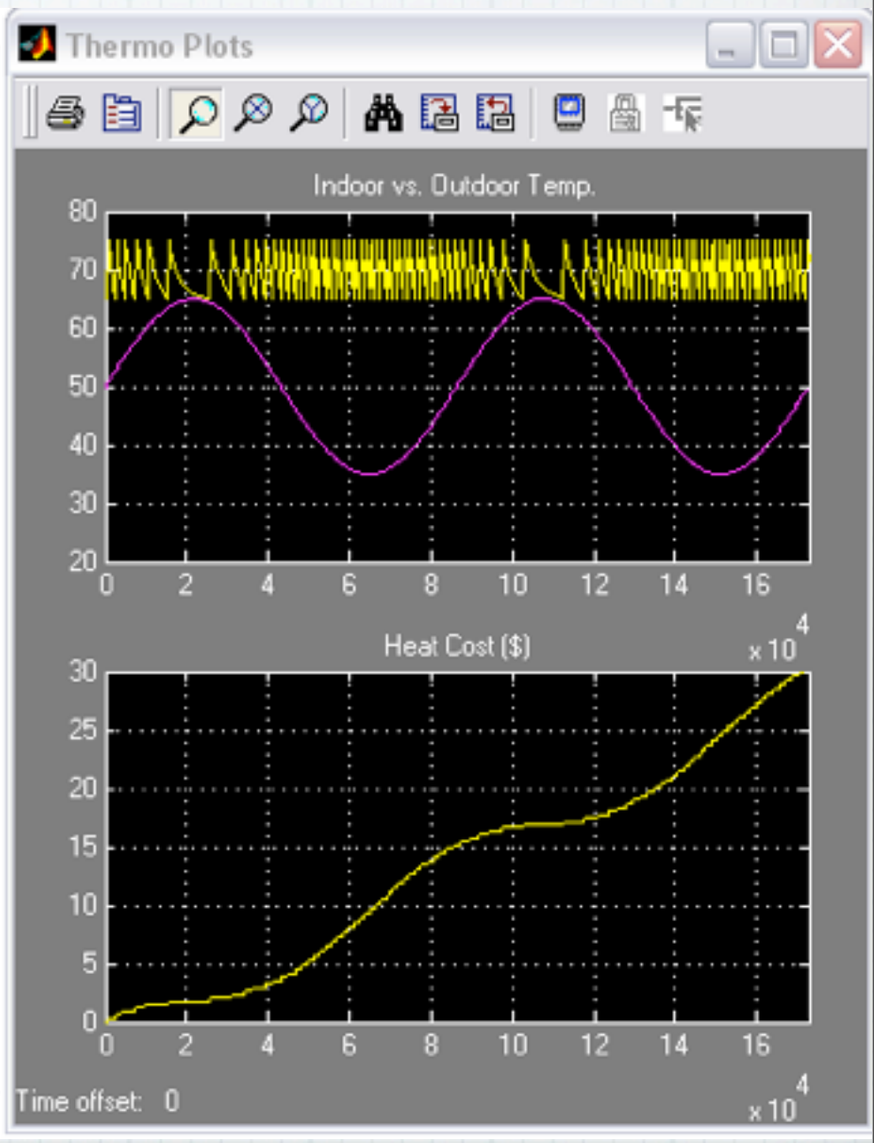
??

「ハイブリッドでしょ！」

制御理論・
力学系理論

Matlab/Simulink





- 産業界でのデファクト・スタンダード（オープンソースの代替も多数）
- 用途： モデリング，シミュレーション（テスト），解析

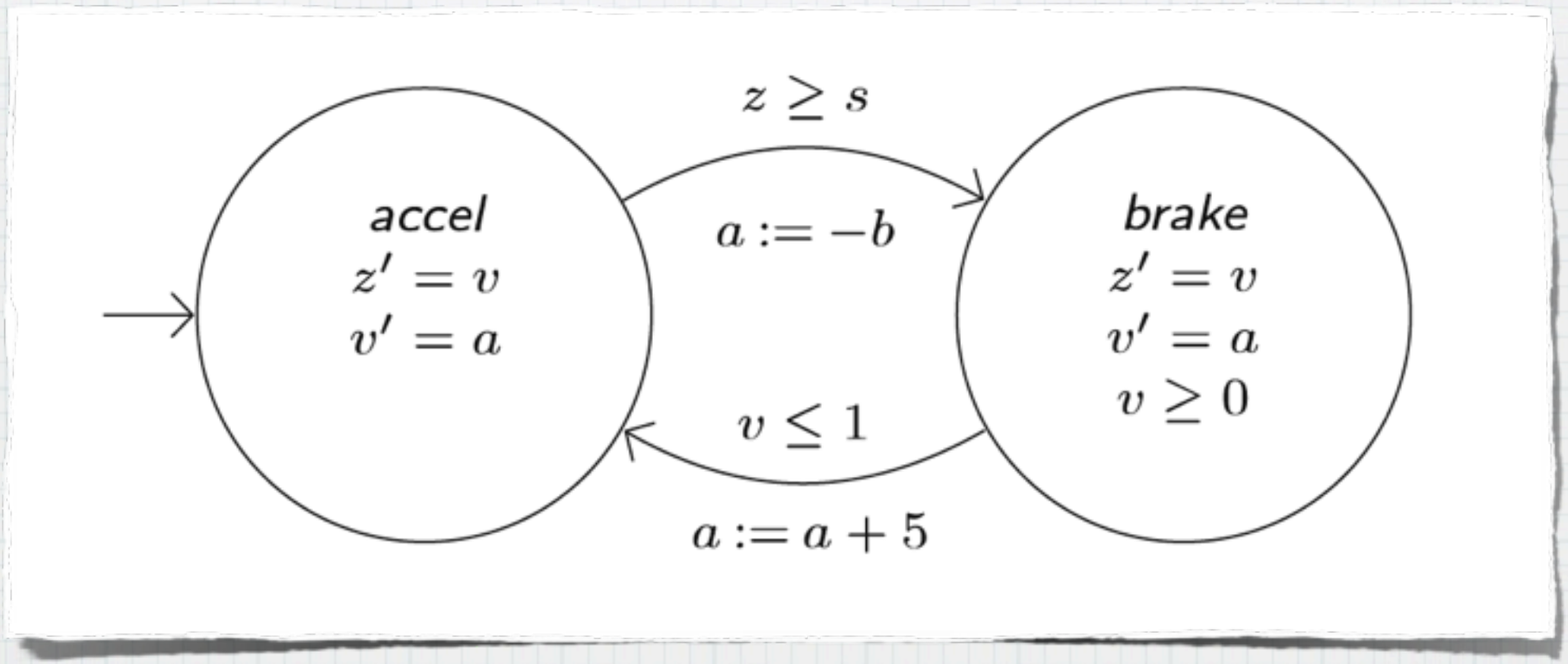
システム検証：
離散から
ハイブリッドへ

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ

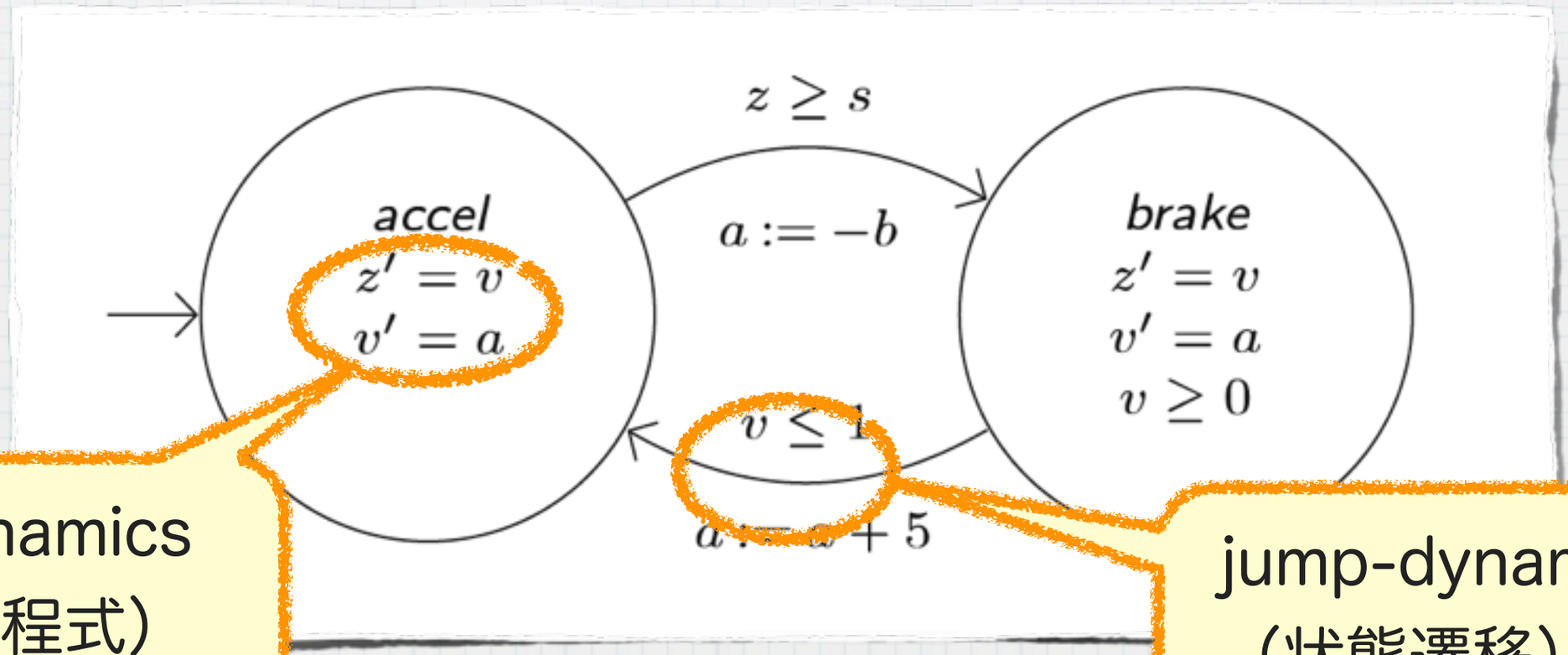
離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
* ハイブリッド・オートマトン [Alur]



離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * ハイブリッド・オートマトン [Alur]



flow-dynamics
（微分方程式）

jump-dynamics
（状態遷移）

蓮尾 一郎

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * Differential Dynamic Logic [Platzer]

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * Differential Dynamic Logic [Platzer]

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

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * 問題点： 微分方程式を書きたくない！

* 提案手法：
  * 既存の離散的検証手法が
    何でも，文字通り移転できる（原理的には）
  * 数学的にスジよく，超準解析で

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * 問題点： 微分方程式を書きたくない！

* 提案手法：

Flow を Jump に変換

  * 既存の離散的検証手法が
    何でも，文字通り移転できる（原理的には）
  * 数学的にスジよく，超準解析で

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * 問題点： 微分方程式を書きたくない！

* 提案手法：

Flow を Jump に変換

  * 既存の離散的検証手法が
    何でも，文字通り移転できる（原理的には）

離散からハイブリッドへ

* 既存研究： 微分方程式を書いちゃえ
  * 問題点： 微分方程式を書きたくない！

* 提案手法：

Flow を Jump に変換

  * 既存の離散的検証手法が
    何でも，文字通り移転できる（原理的には）
  * 数学的にスジよく，超準解析で

Flow を Jump に変換

Flow を Jump に変換

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

Flow を Jump に変換

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

* Infinitesimal number dt
  * "Infinitely small" :
    0 < dt < r for **any** positive real r
* 実行後 t = 1?
* 超準解析

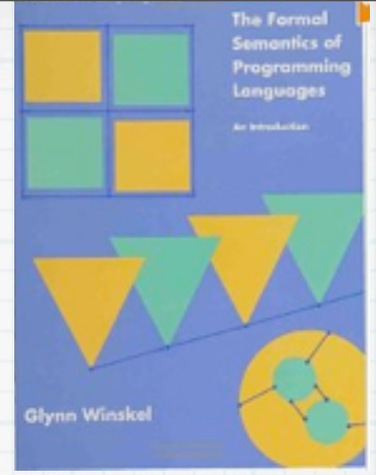
  Nonstandard analysis!

  [Robinson '60s]

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

Theoretical Framework

[Suenaga&H., ICALP'11]

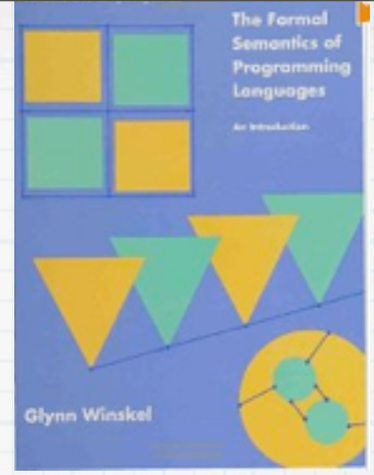


The standard textbook [Winskel]

Hasuo (Tokyo)

Theoretical Framework

[Suenaga&H., ICALP'11]



The standard textbook [Winskel]

While

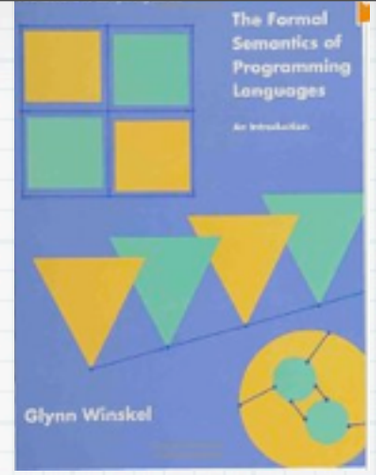
Programming lang.

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

Hasuo (Tokyo)

Theoretical Framework

[Suenaga&H., ICALP'11]



The standard textbook [Winskel]

While

Programming lang.

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

Assn

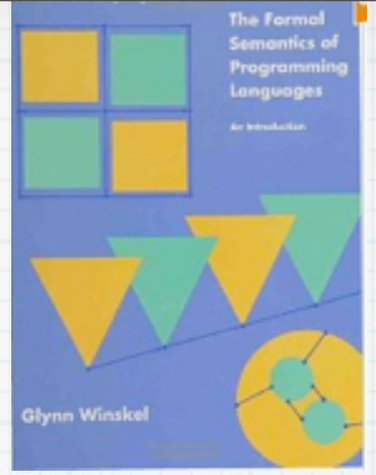
First-order assertion lang.

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

Hasuo (Tokyo)

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.

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

Hoare

Hoare-style program logic

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

Hasuo (Tokyo)

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.

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

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]

While^{dt}

Programming lang.

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

Assn^{dt}

First-order assertion lang.

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

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 nonstandard analysis

Hasuo (Tokyo)

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.

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

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 nonstandard analysis

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

Syntax

While^{dt}

$$
\begin{array}{lll}
\textbf{AExp} \ni & a & ::= \quad x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop} \ a_2 \mid \mathtt{dt} \\
& \multicolumn{2}{l}{\text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}} \\
\textbf{BExp} \ni & b & ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\textbf{Cmd} \ni & c & ::= \quad \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & \quad \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}
$$

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$^{\mathtt{dt}}$

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

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

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

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

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

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

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

Tokyo)

Syntax

While$^{\mathtt{dt}}$

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

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

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

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

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

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

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

Tokyo)

While$^{\mathtt{dt}}$

While + dt

Syntax

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

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

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

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

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

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

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

Tokyo)

Syntax

While^{dt}

While + dt

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

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}

$$\frac{}{\{A\}\ \mathtt{skip}\ \{A\}}\ (\textsc{Skip})$$

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

$$\frac{\{A\}\ c_1\ \{C\} \quad \{C\}\ c_2\ \{B\}}{\{A\}\ c_1 ; c_2\ \{B\}}\ (\textsc{Seq})$$

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

$$\frac{\{A \wedge b\}\ c\ \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\ \{A \wedge \neg b\}}\ (\textsc{While})$$

$$\frac{\models A \Rightarrow A' \quad \{A'\}\ c\ \{B'\} \quad \models B' \Rightarrow B}{\{A\}\ c\ \{B\}}\ (\textsc{Conseq})$$

Tokyo)

Syntax

While$^{\mathtt{dt}}$

While + dt

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

Assn$^{\mathtt{dt}}$

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

Hoare$^{\mathtt{dt}}$

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

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

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

Tokyo)

Syntax

While$^{\mathtt{dt}}$

While + dt

$$
\begin{array}{lll}
\textbf{AExp} \ni & a & ::= \quad x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop} \ a_2 \mid \mathtt{dt} \\
& \multicolumn{2}{l}{\text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}} \\
\textbf{BExp} \ni & b & ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\textbf{Cmd} \ni & c & ::= \quad \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & \quad \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}
$$

Assn$^{\mathtt{dt}}$

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

Hoare$^{\mathtt{dt}}$

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

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

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

Tokyo)

Syntax

While$^{\mathtt{dt}}$

While + dt

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

Assn$^{\mathtt{dt}}$

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

$$\frac{}{\{A\}\ \mathtt{skip}\ \{A\}}\ (\textsc{Skip})$$

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

$$\frac{\{A\}\ c_1\ \{C\} \quad \{C\}\ c_2\ \{B\}}{\{A\}\ c_1; c_2\ \{B\}}\ (\textsc{Seq})$$

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

$$\frac{\{A \wedge b\}\ c\ \{A\}}{\{A\}\ \mathtt{while}\ b\ \mathtt{do}\ c\ \{A \wedge \neg b\}}\ (\textsc{While})$$

$$\frac{\models A \Rightarrow A' \quad \{A'\}\ c\ \{B'\} \quad \models B' \Rightarrow B}{\{A\}\ c\ \{B\}}\ (\textsc{Conseq})$$

Tokyo)

Syntax

While^dt

While + dt

$$
\begin{array}{lllll}
\textbf{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop} \ a_2 \mid \mathtt{dt} \\
& \multicolumn{4}{l}{\text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}} \\
\textbf{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\textbf{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & & \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}
$$

Assn^dt

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

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

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

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

Tokyo)

Syntax

While^dt

While + dt

$$
\begin{array}{lllll}
\textbf{AExp} \ni & a & ::= & x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop} \ a_2 \mid \mathtt{dt} \\
& \multicolumn{4}{l}{\text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}} \\
\textbf{BExp} \ni & b & ::= & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\textbf{Cmd} \ni & c & ::= & \mathtt{skip} \mid x := a \mid c_1 ; c_2 \\
& & & \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}
$$

Assn^dt

Assn, *-transformed

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

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip})$$

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

$$\frac{\{A\}\,c_1\,\{C\} \quad \{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq})$$

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

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\ b\ \mathtt{do}\ c\,\{A \wedge \neg b\}}\ (\textsc{While})$$

$$\frac{\models A \Rightarrow A' \quad \{A'\}\,c\,\{B'\} \quad \models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$

Tokyo)

Syntax

While^dt

While + dt

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

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

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

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

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

Tokyo)

Syntax

While^dt

While + dt

$$
\begin{array}{lll}
\textbf{AExp} \ni & a & ::= \quad x \mid \mathtt{c}_r \mid a_1 \ \mathtt{aop}\ a_2 \mid \mathtt{dt} \\
& \multicolumn{2}{l}{\text{where } \mathtt{c}_r \text{ is a const. for } r \in \mathbb{R},\ \mathtt{aop} \in \{+, -, \cdot, \hat{}, /\}} \\
\textbf{BExp} \ni & b & ::= \quad \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\textbf{Cmd} \ni & c & ::= \quad \mathtt{skip} \mid x := a \mid c_1; c_2 \\
& & \quad \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c
\end{array}
$$

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

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

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

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

Tokyo)

Syntax

While$^{\mathtt{dt}}$

While + dt

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

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

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

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

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

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

Hoare$^{\mathtt{dt}}$

Hasuo (Tokyo)

Syntax

While^{dt}

While + dt

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

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

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

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

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

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

Hoare^{dt}

Precisely the same rules

Hasuo (Tokyo)

Syntax

While^{dt}

While + dt

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

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^{dt}

While + dt

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

Thm.

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

Hoare^{dt}

Precise

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

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

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

Tokyo)

While^{dt} プログラムの例

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

While^{dt} プログラムの例


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

Hasuo (Tokyo)

While^{dt} プログラムの例


```
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
    }}
```

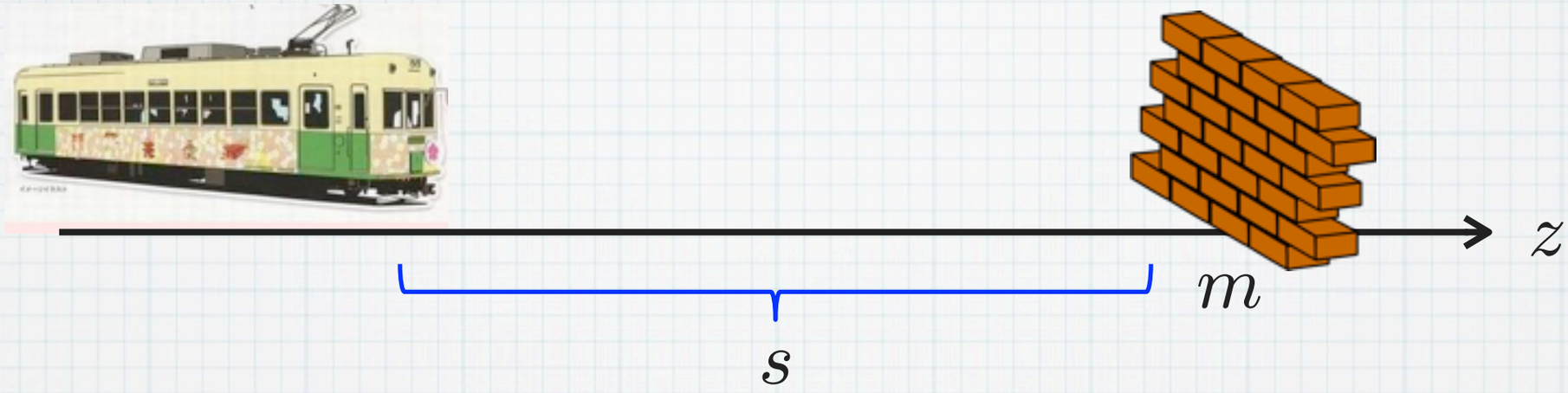
Hasuo (Tokyo)

While^{dt} プログラムの例


```
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
    }}
```

Hasuo (Tokyo)

While^{dt} プログラムの例




```
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
    }}
```

Hasuo (Tokyo)

(真な)
Hoare^{dt} triple の例

$$\left\{\, v^2 \le 2b(m-z) \,\right\} \left[\begin{array}{l} \texttt{while}\ v > 0\ \texttt{do} \\ \quad \begin{array}{l} z := z + v \cdot \mathrm{dt}\,; \\ v := v - b \cdot \mathrm{dt}\,; \\ t := t + \mathrm{dt} \end{array} \end{array}\right] \left\{\, z < m \,\right\}$$

Hasuo (Tokyo)

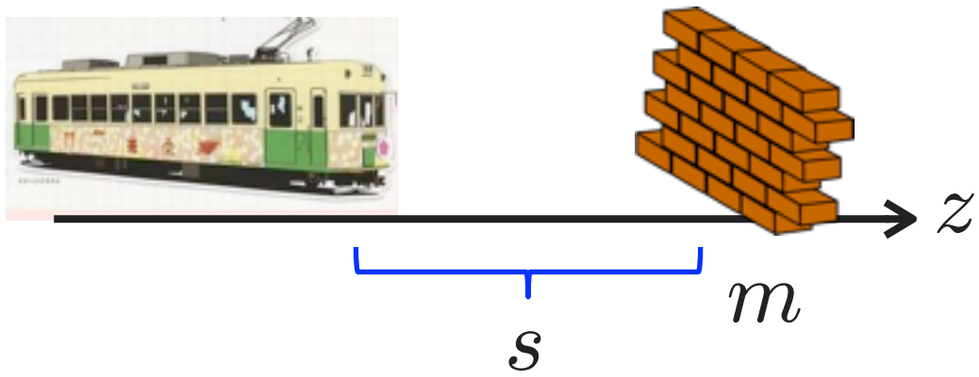
(真な)
Hoare^{dt} triple の例

$$\left\{\, v^2 \le 2b(m-z) \,\right\} \left[\begin{array}{l} \texttt{while}\ v > 0\ \texttt{do} \\ \quad \begin{array}{l} z := z + v \cdot \mathrm{dt}\,; \\ v := v - b \cdot \mathrm{dt}\,; \\ t := t + \mathrm{dt} \end{array} \end{array}\right] \left\{\, z < m \,\right\}$$

Hoare^{dt} で
簡単に導出できます
（ループ不変量を使って）

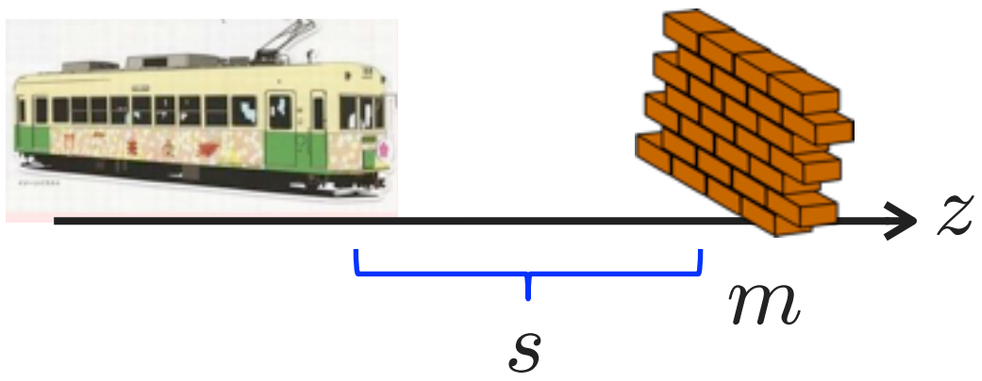
(Tokyo)

提案手法のワークフロー



「壁にぶつかったら
困るじゃないか．．．
ええと，仮定Aのもとで」

提案手法のワークフロー



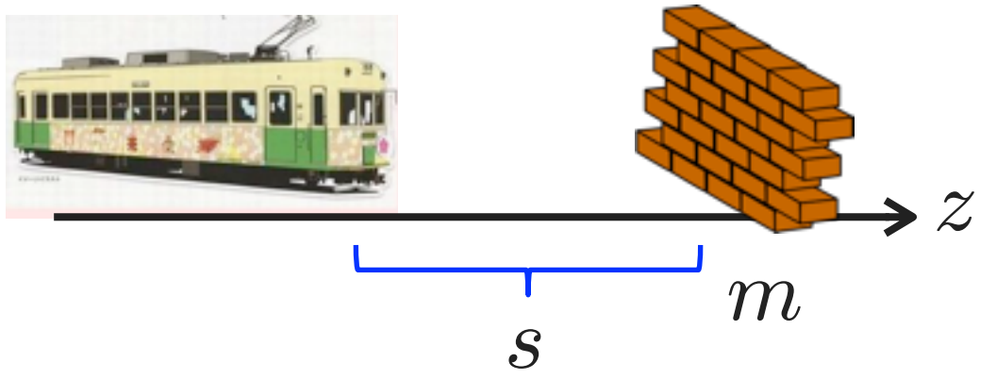
「壁にぶつかったら
困るじゃないか．．．
ええと，仮定Aのもとで」

1. モデリング

$c :=$

```
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
    }}
```

提案手法のワークフロー



「壁にぶつかったら
困るじゃないか．．．
ええと，仮定Aのもとで」

1. モデリング

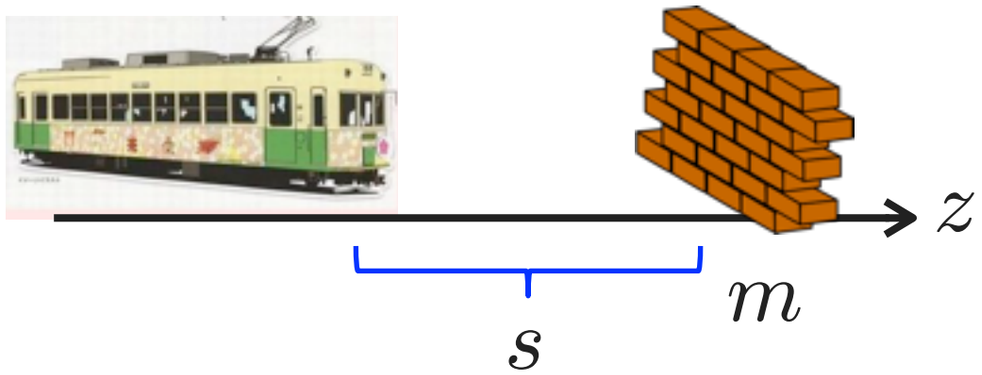
$c :=$

```
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
    }}
```

2. 仕様記述

$$\{A\}\ c\ \{z < m\}$$

提案手法のワークフロー



「壁にぶつかったら
困るじゃないか．．．
ええと，仮定Aのもとで」

1. モデリング

$c :=$

```
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
    }}
```

2．仕様記述

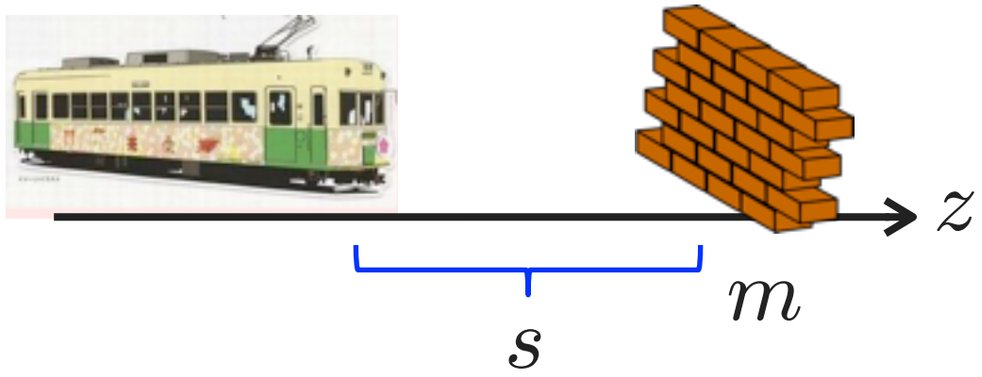
$$\{A\}\ c\ \{z < m\}$$

3．形式検証

Hoare 論理で
証明検索

蓮尾 一郎（東

提案手法のワークフロー
(本当は)



「壁にぶつかったら
困るじゃないか．．．
事前条件は何かな？」

1. モデリング

$c :=$

```
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
    }}
```

2．仕様記述

$$\{A\}\ c\ \{z < m\}$$

3．形式検証

事前条件 A と証明を同時に検索

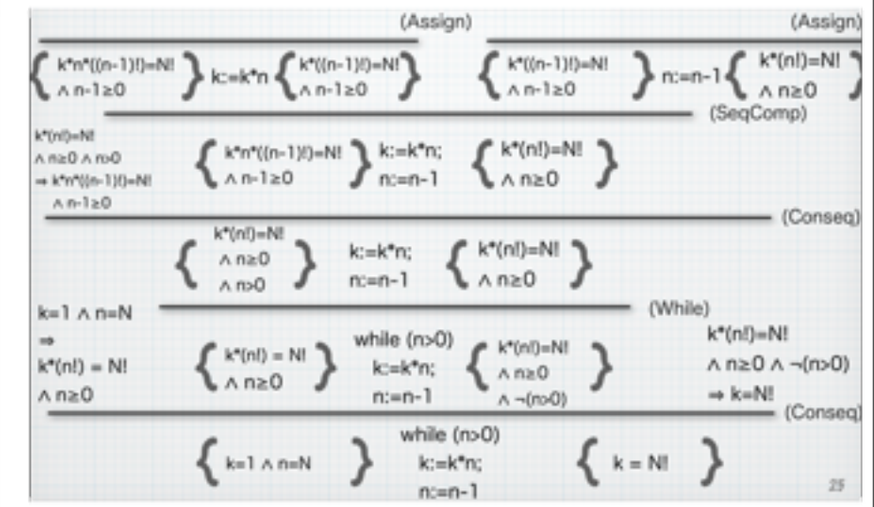
蓮尾 一郎

Hoare^{dt} の健全性

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

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

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



Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff} \left[\begin{array}{l}\text{for each memory state } \sigma,\\ \sigma \models A \text{ implies } [\![c]\!](\sigma) \models B.\end{array}\right]$$

Hoare^{dt} の健全性

$$\frac{}{\{A\}\,\mathtt{skip}\,\{A\}}\ (\textsc{Skip})$$

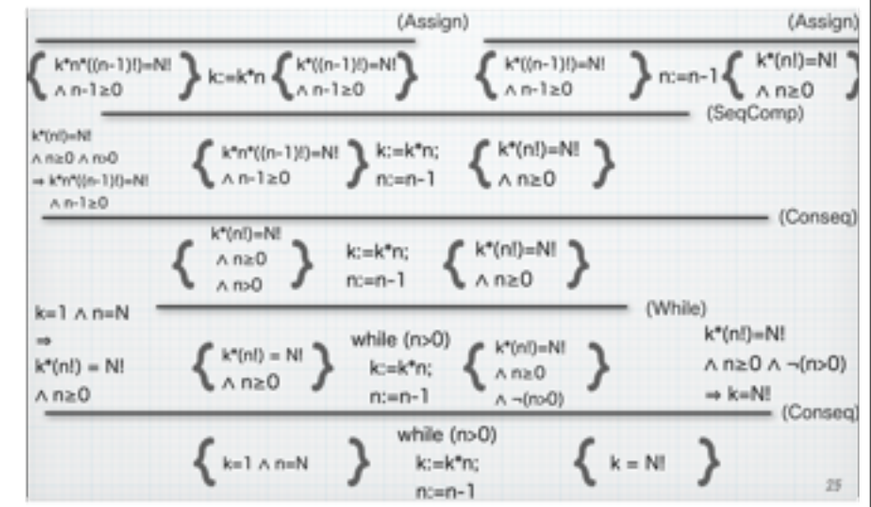
$$\frac{}{\{A[a/x]\}\,x := a\,\{A\}}\ (\textsc{Assign})$$

$$\frac{\{A\}\,c_1\,\{C\}\quad\{C\}\,c_2\,\{B\}}{\{A\}\,c_1; c_2\,\{B\}}\ (\textsc{Seq})$$

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

$$\frac{\{A \wedge b\}\,c\,\{A\}}{\{A\}\,\mathtt{while}\ b\ \mathtt{do}\ c\,\{A \wedge \neg b\}}\ (\textsc{While})$$

$$\frac{\models A \Rightarrow A'\quad\{A'\}\,c\,\{B'\}\quad\models B' \Rightarrow B}{\{A\}\,c\,\{B\}}\ (\textsc{Conseq})$$



Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff} \left[\begin{array}{l}\text{for each memory state } \sigma,\\ \sigma \models A \text{ implies } [\![c]\!](\sigma) \models B.\end{array}\right]$$

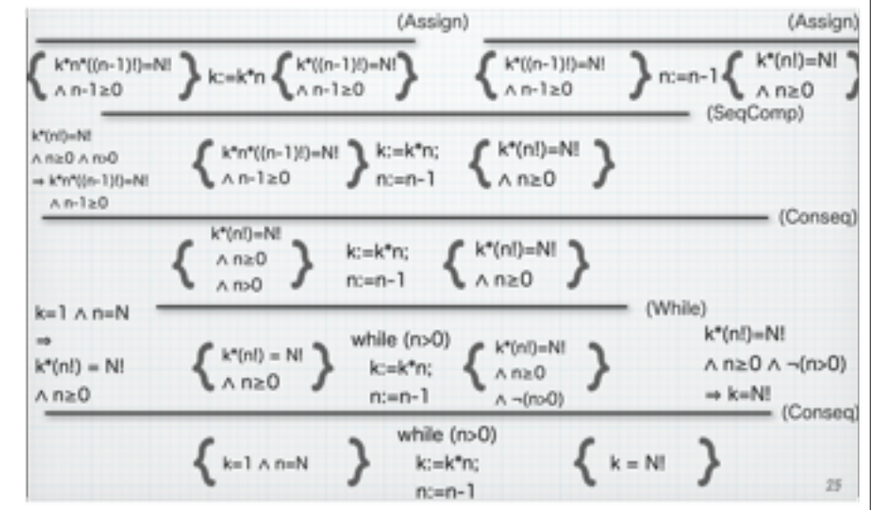
プログラム c の「意味」
（メモリ状態の変換として）

Hoare^{dt} の健全性

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

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

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



Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff} \left[\begin{array}{l}\text{for each memory state } \sigma,\\ \sigma \models A \text{ implies } [\![c]\!](\sigma) \models B.\end{array}\right]$$

これは？？

プログラム c の「意味」
（メモリ状態の変換として）

While^{dt} プログラムの例

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

超準解析による
物理情報システムの形式検証

アウトライン

* 形式検証とは？
  * Hoare 論理を例に
* ハイブリッド・システム
  * 離散＋連続
  * 物理情報システムの一側面
* 超準解析による移転
  * 離散的検証手法を，文字通りそのまま
    ハイブリッド・システムに適用

3

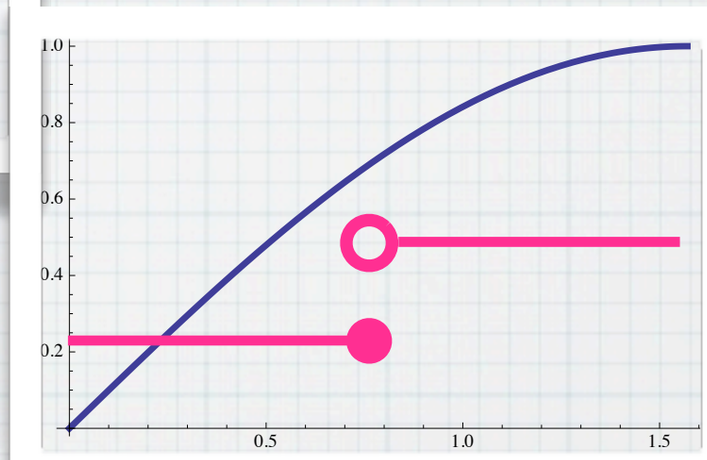
超準解析による

移転

Nonstandard Analysis

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

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



* Done naively ➜ contradiction!

Nonstandard Analysis

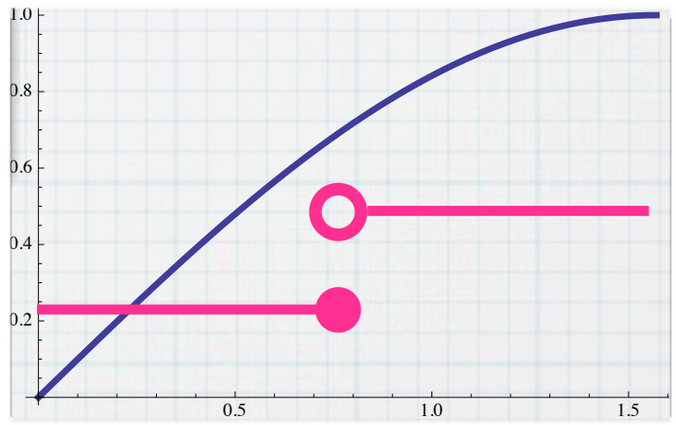
* Analysis with an **infinitesimal** ∂, e.g.

"Infinitely small"

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

f is continuous $\iff$

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



* Done naively ➜ contradiction!

Hasuo (Tokyo)

Nonstandard Analysis

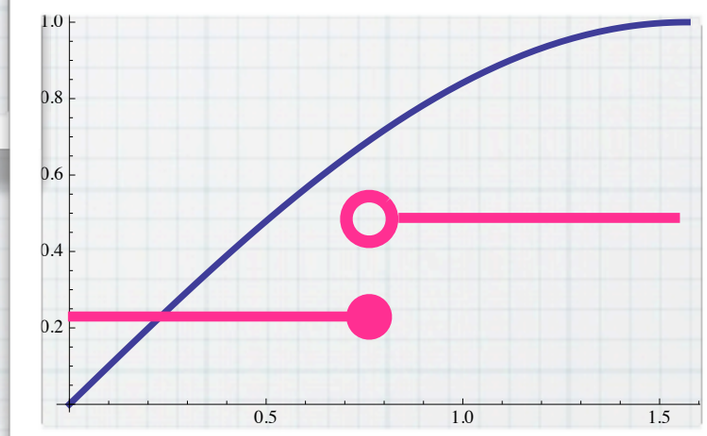
* Analysis with an **infinitesimal** ∂, e.g.

"Infinitely small"

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

f is continuous $\iff$

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



* 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

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

Ignore

Hyperreals

= Reals + Infinitesimals + ...

Defn.

The set of *hyperreal numbers* is

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

Ignore

0th section

1st section

2nd section

Hyperreals

= Reals + Infinitesimals + ...

Defn.

The set of *hyperreal numbers* is

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

0th section

1st section

2nd section

Ignore

* Operations: sectionwise

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.

The set of *hyperreal numbers* is

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

0th section 1st section 2nd section

Ignore

- Operations: sectionwise

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

- Reals are hyperreals

$$\mathbb{R} \hookrightarrow {}^*\mathbb{R},$$

$$r \mapsto \left[(r, r, \dots)\right]$$

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

* Predicates:
sectionwise,
"for almost all i"

Hyperreals

= Reals + Infinitesimals + …

Defn.
The set of *hyperreal numbers* is

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

* Predicates:
sectionwise,
"for almost all i"

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + …

Defn.
The set of *hyperreal numbers* is

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

* Predicates:
sectionwise,
"for almost all i"

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

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

$$\begin{aligned}
&\left[(a_i)_{i\in\mathbb{N}}\right] < \left[(b_i)_{i\in\mathbb{N}}\right] \\
&\iff \quad a_i < b_i \qquad \text{“for almost every } i\text{”} \\
&\impliedby \quad \{i \in \mathbb{N} \mid a_i \not< b_i\} \qquad \text{is finite}
\end{aligned}$$

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

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

Hasuo (Tokyo)

Hyperreals

= Reals + Infinitesimals + ...

Defn.
The set of *hyperreal numbers* is

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

OK! ^ ✖ ✖ ✖ ✖ ✓ ✓ ...

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

Hasuo (Tokyo)

Trouble... Resolved

$$0 \qquad [(\,1, -1,\ 1, -1, \ldots)]$$

Trouble... Resolved

$$0 \begin{matrix} > \\ = \\ < \end{matrix} \left[(\,1, -1,\ 1, -1, \ldots)\right]$$

??

Trouble... Resolved

$$0 \quad \begin{matrix} > \\ = \\ < \end{matrix} \quad [(\,1, -1,\ 1, -1, \ldots)]$$

??

* Meaning of "almost every i" extended
  * ... so that

> For each $S \subset \mathbb{N}$, exactly one of
>
> $$S \quad \text{and} \quad \mathbb{N} \setminus S$$
>
> is "almost all i."

Trouble... Resolved

$$0 \quad \begin{matrix} > \\ = \\ < \end{matrix} \quad [(\,1, -1,\ 1, -1, \ldots)]$$

??

* Meaning of "almost every i" extended
  * ... so that

For each $S \subset \mathbb{N}$, exactly one of

$$S \quad \text{and} \quad \mathbb{N} \setminus S$$

is "almost all i."

* ➜ Ultrafilter!

Defn.
The set of *hyperreal numbers* is

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

Trouble... Resolved

$$0 \quad \begin{matrix} > \\ = \\ < \end{matrix} \quad [(\,1, -1,\ 1, -1, \ldots)]$$

??

* Meaning of "almost every i" extended
  * ... so that

For each $S \subset \mathbb{N}$, exactly one of

$$S \quad \text{and} \quad \mathbb{N} \setminus S$$

is "almost all i."

* ➜ Ultrafilter!

Defn.
The set of *hyperreal numbers* is

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

Filters & Ultrafilters

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

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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.
A *filter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is that which satisfies Cond. 2.–4.

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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.
A *filter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is that which satisfies Cond. 2.–4.

Prop.

$$\mathcal{F}_{\mathbf{c}} := \{S \subseteq \mathbb{N} \mid \mathbb{N} \setminus S \text{ is finite}\}$$

is a filter (the *cofinite/Frechet* filter).

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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.
A *filter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is that which satisfies Cond. 2.–4.

Prop.

$$\mathcal{F}_{\mathbf{c}} := \{S \subseteq \mathbb{N} \mid \mathbb{N} \setminus S \text{ is finite}\}$$

is a filter (the *cofinite/Frechet* filter).

Prop.
Any filter $\mathcal{F}'$ can be extended to an ultrafilter $\mathcal{F} \supseteq \mathcal{F}'$. (By Zorn's lemma)

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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.
A *filter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is that which satisfies Cond. 2.–4.

Prop.

$$\mathcal{F}_c := \{S \subseteq \mathbb{N} \mid \mathbb{N} \setminus S \text{ is finite}\}$$

is a filter (the *cofinite/Frechet* filter).

Prop.
Any filter $\mathcal{F}'$ can be extended to an ultrafilter $\mathcal{F} \supseteq \mathcal{F}'$. (By Zorn's lemma)

Cor.
There is an ultrafilter $\mathcal{F}$ such that $\mathcal{F}_c \subseteq \mathcal{F}$.

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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.
A *filter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is that which satisfies Cond. 2.–4.

Prop.

$$\mathcal{F}_c := \{S \subseteq \mathbb{N} \mid \mathbb{N} \setminus S \text{ is finite}\}$$

is a filter (the *cofinite/Frechet* filter).

Prop.
Any filter $\mathcal{F}'$ can be extended to an ultrafilter $\mathcal{F} \supseteq \mathcal{F}'$. (By Zorn's lemma)

Cor.
There is an ultrafilter $\mathcal{F}$ such that $\mathcal{F}_c \subseteq \mathcal{F}$.

Filters & Ultrafilters

Given $X \subseteq \mathbb{N}$ is "yes, almost all!" or "no!"

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.
A *filter* $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$ is that which satisfies Cond. 2.–4.

Prop.

$$\mathcal{F}_c := \{S \subseteq \mathbb{N} \mid \mathbb{N} \setminus S \text{ is finite}\}$$

is a filter (the *cofinite/Frechet* filter).

Prop.
Any filter $\mathcal{F}'$ can be extended to an ultrafilter $\mathcal{F} \supseteq \mathcal{F}'$. (By Zorn's lemma)

Cor.
There is an ultrafilter $\mathcal{F}$ such that $\mathcal{F}_c \subseteq \mathcal{F}$.

Fix one such

Hyperreals

Defn.
The set of *hyperreal numbers* is

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

Hyperreals

Defn.
The set of *hyperreal numbers* is

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

$$(a_0, a_1, \dots) \sim_{\mathcal{F}} (b_0, b_1, \dots)$$

$$\overset{\text{def}}{\iff} \{i \in \mathbb{N} \mid a_i = b_i\} \in \mathcal{F}$$

Hyperreals

Defn.
The set of *hyperreal numbers* is

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

$$(a_0, a_1, \dots) \sim_{\mathcal{F}} (b_0, b_1, \dots)$$
$$\overset{\text{def}}{\iff} \{i \in \mathbb{N} \mid a_i = b_i\} \in \mathcal{F}$$

* Predicates: pointwise, "for almost every i"

$$\begin{aligned} &[(a_i)_{i\in\mathbb{N}}] < [(b_i)_{i\in\mathbb{N}}] \\ \iff\quad & a_i < b_i \qquad \text{for "almost every } i\text{"} \\ \iff\quad & \{i \in \mathbb{N} \mid a_i < b_i\} \in \mathcal{F} \end{aligned}$$

* Consequences: ω is infinite; ω^{-1} is infinitesimal;

*R is an ordered field;
[(0,1,0,1,...)] is either 0 or 1; ...

移転原理

The Transfer Principle

Thm.
For any first-order sentence φ,

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

移転原理

The Transfer Principle

Thm.
For any first-order sentence φ,

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

$\forall x.\, \psi$

$\forall x.\, \psi$

移転原理

The Transfer Principle

Thm.
For any first-order sentence φ,

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

$\forall x \in \mathbb{R}.\, \psi$

$\forall x \in {}^*\mathbb{R}.\, \psi$

移転原理

The Transfer Principle

Thm.
For any first-order sentence φ,

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

$\forall x \in \mathbb{R}.\, \psi$ —*-transform→ $\forall x \in {}^*\mathbb{R}.\, \psi$

移転原理

The Transfer Principle

Thm.
For any first-order sentence φ,

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

$\forall x, y.\, (x < y \vee x = y \vee x > y)$

$\forall x.\, (x \neq 0 \implies \exists y.\, (xy = 1))$

$\forall x \in \mathbb{R}.\, \psi$ —*-transform→ $\forall x \in {}^*\mathbb{R}.\, \psi$

移転原理

The Transfer Principle

$\forall x, y.\,(x < y \vee x = y \vee x > y)$

$\forall x.\,(x \neq 0 \implies \exists y.\,(xy = 1))$

Thm.
For any first-order sentence φ,

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

$\forall x \in \mathbb{R}.\,\psi$ —*-transform→ $\forall x \in {}^*\mathbb{R}.\,\psi$

我々の戦略：「Hoare 論理に対する移転原理」

THE COAUTHOR

Kohei Suenaga

- PhD (U. Tokyo, 2008)
  with Naoki Kobayashi
- Program verification, static analysis,
  type systems
- Industiral experience
  (IBM Research)
- Assoc. Prof. at Kyoto U. (2013-)

THE COAUTHOR

Kohei Suenaga

- PhD (U.Tokyo, 2008)
  with Naoki Kobayashi
- Program verification, static analysis,
  type systems
- Industiral experience
  (IBM Research)
- Assoc. Prof. at Kyoto U. (2013-)

THE COAUTHOR

Kohei Suenaga

- PhD (U. Tokyo, 2008)
  with Naoki Kobayashi
- Program verification, static analysis,
  type systems
- Industiral experience
  (IBM Research)
- Assoc. Prof. at Kyoto U. (2013-)

THE COAUTHOR

Kohei Suenaga

- PhD (U.Tokyo, 2008)
  with Naoki Kobayashi
- Program verification, static analysis,
  type systems
- Industiral experience
  (IBM Research)
- Assoc. Prof. at Kyoto U. (2013-)

THE COAUTHOR

Kohei Suenaga

- PhD (U.Tokyo, 2008)
  with Naoki Kobayashi
- Program verification, static analysis,
  type systems
- Industiral experience
  (IBM Research)
- Assoc. Prof. at Kyoto U. (2013-)

THE COAUTHOR

Kohei Suenaga

- PhD (U.Tokyo, 2008)
  with Naoki Kobayashi
- Program verification, static analysis,
  type systems
- Industiral experience
  (IBM Research)
- Assoc. Prof. at Kyoto U. (2013-)

[Fixed pt. obs., Braga, PT, 2007]

超準解析による While^{dt} の意味論

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

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

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

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

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

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := (0,0,0,...);
while (t < (1,1,1,...))
    t := t + (1, 1/2, 1/3, ...) ;
```

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

0th section

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

  t := t + 1 ;
```

1st section

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

  t := t + 1/2 ;
```

2nd section

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

  t := t + 1/3 ;
```

...

Hasuo (Tokyo)

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 + 1/2 ;`
		`t := 0;`
		`while (t < 1)`
		`  t := t + 1/3 ;`
			…

`t = 1`	`t = 1`	`t = 1`	…

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := (0,0,0,...);
while (t < (1,1,1,...))
    t := t + (1, 1/2, 1/3, ...) ;
```

```
t = (1,1,1,...)
```

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

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

```
t = 1
```

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := 0;
while (true)
  t := t + dt;
```

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := 0;
while (true)
   t := t + dt;
```

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := (0,0,0,...);
while (true)

   t := t + (1, 1/2, 1/3, ...) ;
```

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

0th section

```
t := 0;
while (true)

  t := t + 1 ;
```

1st section

```
t := 0;
while (true)

  t := t + 1/2 ;
```

2nd section

```
t := 0;
while (true)

  t := t + 1/3 ;
```

...

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

0th section	1st section	2nd section	
`t := 0; while (true) t := t + 1;`	`t := 0; while (true) t := t + 1/2;`	`t := 0; while (true) t := t + 1/3;`	…
$\bot$	$\bot$	$\bot$	…

Hasuo (Tokyo)

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := (0,0,0,...);
while (true)
   t := t + (1, 1/2, 1/3, ...) ;
```

$$t = (\bot, \bot, \bot, \ldots)$$

Denotational Semantics

* Execute sectionwise and bundle up the outcomes!

```
t := 0;
while (true)
   t := t + dt;
```

⊥

Denotational Semantics

$$
\begin{array}{lcl}
[\![x]\!]\sigma & := & \sigma(x) \\
[\![a_1 \mathbin{\mathtt{aop}} a_2]\!]\sigma & := & [\![a_1]\!]\sigma \mathbin{\mathtt{aop}} [\![a_2]\!]\sigma \\
[\![\mathtt{dt}]\!]\sigma & := & \omega^{-1} = \left[\left(1, \tfrac{1}{2}, \tfrac{1}{3}, \dots\right) \right]
\end{array}
\qquad
\begin{array}{lcl}
[\![\mathtt{c}_r]\!]\sigma & := & r \quad \text{for each } r \in \mathbb{R} \\
[\![\infty]\!]\sigma & := & \omega = \left[\left(1, 2, 3, \dots\right) \right]
\end{array}
$$

$$
\begin{array}{lcl}
[\![\mathtt{true}]\!]\sigma & := & \mathrm{tt} \\
[\![b_1 \wedge b_2]\!]\sigma & := & [\![b_1]\!]\sigma \wedge [\![b_2]\!]\sigma \\
[\![a_1 < a_2]\!]\sigma & := & [\![a_1]\!]\sigma < [\![a_2]\!]\sigma
\end{array}
\qquad
\begin{array}{lcl}
[\![\mathtt{false}]\!]\sigma & := & \mathrm{ff} \\
[\![\neg b]\!]\sigma & := & \neg([\![b]\!]\sigma)
\end{array}
$$

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

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

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

Sectionwise definition

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

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

Hoare^{dt} の健全性

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

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

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

Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff} \left[\begin{array}{l}\text{for each memory state } \sigma,\\ \sigma \models A \text{ implies } [\![c]\!](\sigma) \models B.\end{array}\right]$$

Hoare^{dt} の健全性

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

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

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

Thm. (Soundness)

$$\vdash \{A\}\,c\,\{B\} \implies \models \{A\}\,c\,\{B\}\ ,$$

where

$$\models \{A\}\,c\,\{B\} \overset{\text{def}}{\iff} \left[\begin{array}{l}\text{for each memory state } \sigma,\\ \sigma \models A \text{ implies } [\![c]\!](\sigma) \models B.\end{array}\right]$$

プログラム c の「意味」
（メモリ状態の変換として）

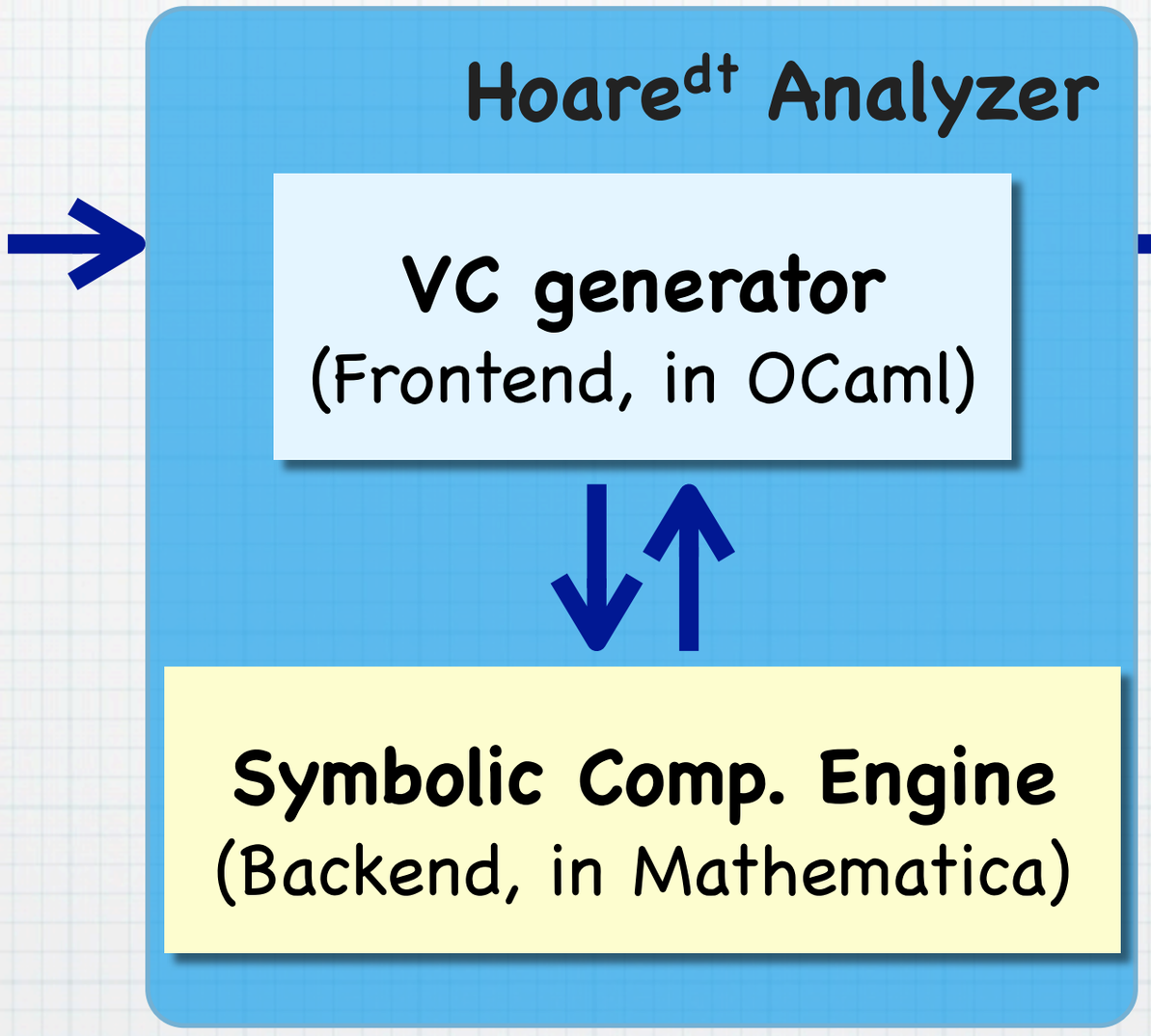
Prototype Automatic Prover

P

(While^{dt} program)

B

(postcondition)



A

(precondition)

s.t.

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

* 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

Hasuo (Tokyo)

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

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

Hasuo (Tokyo)

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.
x x86 (64-bit)

th one manual insertion of invariants

Q. Does the choice of dt matter?

Q. Does the choice of dt matter?

* A.
  Yes, for some "pathological" programs

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

Terminates with dt = (1, 1/2, 1/3, …)

Doesn't with dt = (1/π, 1/2π, 1/3π, …)

Q. While^{dt} program って実行できなくない？

Q. While^{dt} program って実行できなくない？

* A. その通り，実行できません.

Q. While^{dt} program って実行できなくない？

* A. その通り，実行できません.
  * ~~プログラミング言語~~ モデリング言語

Q. While^{dt} program って実行できなくない？

* A. その通り，実行できません.
  * ~~プログラミング言語~~ モデリング言語
  * ~~数値的近似~~
    無限小値による，exact なモデリング

Q. While^{dt} program って実行できなくない？

* A. その通り，実行できません.
  * ~~プログラミング言語~~ モデリング言語
  * ~~数値的近似~~
    無限小値による，exact なモデリング

* Static analysis
  ➜ 実行できなくてOK！

Q. While^{dt} program って実行できなくない？

* A. その通り，実行できません.
  * ~~プログラミング言語~~ モデリング言語
  * ~~数値的近似~~
    無限小値による，exact なモデリング
  * Static analysis
    ➜ 実行できなくてOK！

Static analysis （静的解析）

* 実行することなく解析
* ↔ Dynamic analysis
* 例： プログラム論理による検証

Q. While^{dt} program って実行できなくない？

* A. その通り，実行できません.
  * ~~プログラミング言語~~ モデリング言語
  * ~~数値的近似~~
    無限小値による，exact なモデリング
  * Static analysis
    ➜ 実行できなくてOK！
  * 「意味論＋健全性」で十分

Static analysis （静的解析）

* 実行することなく解析
* ↔ Dynamic analysis
* 例： プログラム論理による検証

まとめとこれから

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

数学（数理論理学）
による品質保証

* 微小量（infinitesimal）を持つ，
  解析学の形式化
* Ultrafilter によるモデル
  ➜ 数理論理学（モデル理論）の成果！

超準解析による

物理情報システムの形式検証

形式検証の新たなターゲット

* 連続ダイナミクス（物理系）
* 離散ダイナミクス
  （デジタル制御）

数学（数理論理学）
による品質保証

論文

* [ICALP'11] K. Suenaga and I. Hasuo. Programming with infinitesimals: A while-language for hybrid system modeling. In L. Aceto, M. Henzinger and J. Sgall, editors, ICALP (2), vol. 6756 of Lect. Notes Comp. Sci., pp. 392–403. Springer, 2011.

* [CAV'12] I. Hasuo and K. Suenaga. Exercises in Nonstandard Static Analysis of hybrid systems. In P. Madhusudan and S.A. Seshia, editors, CAV, vol. 7358 of Lect. Notes Comp. Sci., pp. 462–478. Springer, 2012.

* [POPL'13] K. Suenaga, H. Sekine and I. Hasuo. Hyperstream processing systems: nonstandard modeling of continuous-time signals. In R. Giacobazzi and R. Cousot, editors, POPL, pp. 417–430. ACM, 2013.

論文・スライドなどはウェブページから

http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro/

ハイブリッド・システム

形式検証

離散的データ
(Jump)

と

連続的データ
(Flow)

制御理論・
力学系理論

ハイブリッド・システム

離散的データ
(Jump)

と

連続的データ
(Flow)

形式検証

「ハイブリッドだ！」

「ハイブリッドでしょ！」

制御理論・
力学系理論

離散からハイブリッドへ

* 超準解析によるアプローチ（今回）
  * 連続ダイナミクスを「なかったことに」
  * 論理学的基盤（健全性），
    形式検証に適す

離散からハイブリッドへ

* 超準解析によるアプローチ（今回）
  * 連続ダイナミクスを「なかったことに」
  * 論理学的基盤（健全性），
    形式検証に適す

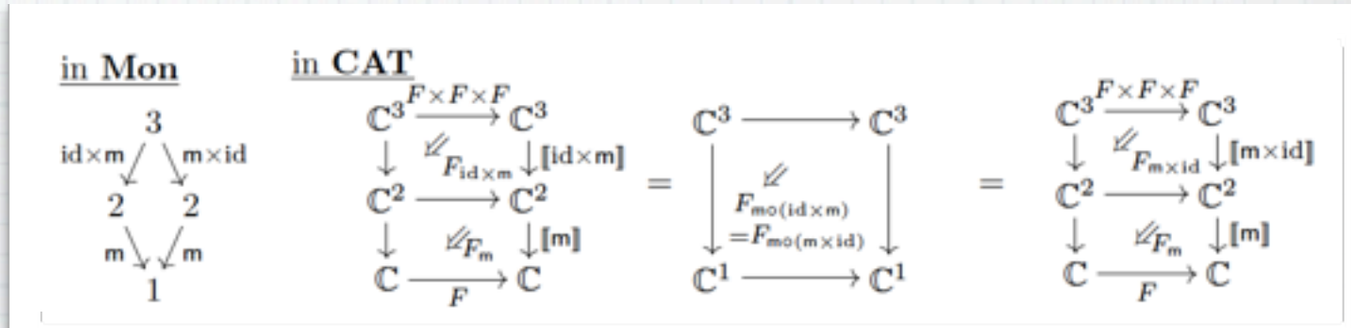
離散からハイブリッドへ

* 超準解析によるアプローチ（今回）
  * 連続ダイナミクスを「なかったことに」
  * 論理学的基盤（健全性），
    形式検証に適す

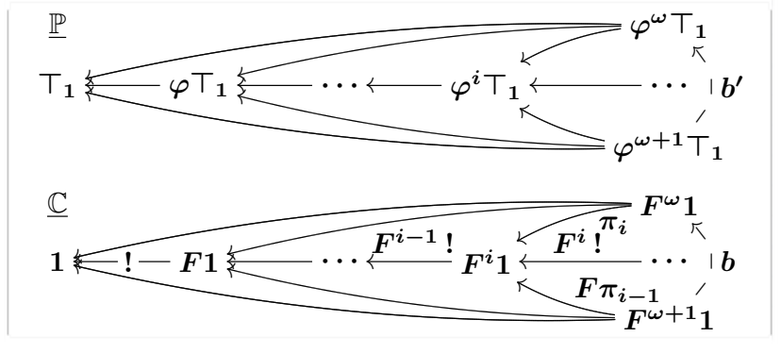
* 微分方程式によるアプローチ
  * 制御理論・力学系理論との協働
    [Tabuada, Pappas, Girard, Dang, Frehse, ...]

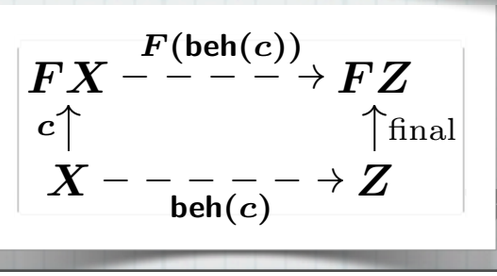
離散からハイブリッドへ

* 超準解析によるアプローチ（今回）
  * 連続ダイナミクスを「なかったことに」
  * 論理学的基盤（健全性），形式検証に適す



* 微分方程式によるアプローチ
  * 制御理論・力学系理論との協働
    [Tabuada, Pappas, Girard, Dang, Frehse, ...]
  * 特に： 圏論 (category theory) による協働





蓮尾 一郎（東大・

圏論的一般化と異分野協働

圏論的に一般化された
模倣関係の理論
[Hasuo '06]

定義 模倣関係とは次のような射 f

$$\begin{array}{ccc} FX & \xrightarrow{Ff} & FY \\ {\scriptstyle c}\uparrow & \rightrightarrows & \uparrow{\scriptstyle d} \\ X & \xrightarrow[f]{} & Y \end{array}$$

一般化,
抽象化
[Hasuo '06]

具体化１
[Hasuo '06]

具体化２
[本研究で期待される成果]

模倣関係による
定性的検証の理論
[Lynch & Vaandrager '95]

定義 模倣関係とは二項関係 R であって…

模倣関係による
定量的検証の理論
[Hasuo '06]

定義 模倣関係とは確率カーネル f であって…

連続系・ハイブリッド系
のための**近似模倣**の理論
cf. [Haghverdi et al. '05] [Girard et al. '11]

定義 ε-模倣関係とは…

制御論的無限モデルと
情報科学的有限モデルとの
間の架け橋

圏論的一般化と異分野協働

圏論的に一般化された
模倣関係の理論
[Hasuo '06]

定義 模倣関係とは次のような射 f

$$\begin{array}{ccc} FX & \xrightarrow{Ff} & FY \\ {\scriptstyle c}\uparrow & \rightrightarrows & \uparrow{\scriptstyle d} \\ X & \xrightarrow[f]{} & Y \end{array}$$

一般化,
抽象化
[Hasuo '06]

具体化１
[Hasuo '06]

具体化２
[本研究で期待される成果]

模倣関係による
定性的検証の理論
[Lynch & Vaandrager '95]

定義 模倣関係とは二項関係 R であって…

模倣関係による
定量的検証の理論
[Hasuo '06]

定義 模倣関係とは確率カーネル f であって…

連続系・ハイブリッド系
のための**近似模倣**の理論
cf. [Haghverdi et al. '05] [Girard et al. '11]

定義 ε**-模倣関係**とは…

制御論的無限モデルと
情報科学的有限モデルとの
間の架け橋

ポイント： 離散系とハイブリッド系に共通する「数学的本質」の抽出

まとめ：

ヘテロジニアス・システムのために，

数理モデリング手法の「一座建立」

- 既存のモデリング手法
  - 微分方程式（連続）
  - 計算機科学，
    有限状態オートマトン（離散）
- これらを「つなぐ」数理のコトバ
  - 論理学（e.g. 超準解析 → 今日の内容）
  - 圏論，category theory
    - 詳細はまたの機会に
    - 圏論の歩き方 （日本評論社） 今夏刊行予定

まとめ：
ヘテロジニアス・システムのために，
数理モデリング手法の「一座建立」

ご清聴ありがとうございました！

蓮尾 一郎（東大・情報理工・コンピュータ科学)
http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro

* 既存のモデリング手法
  * 微分方程式（連続）
  * 計算機科学，
    有限状態オートマトン（離散）
* これらを「つなぐ」数理のコトバ
  * 論理学（e.g. 超準解析 → 今日の内容）
  * 圏論，category theory
    * 詳細はまたの機会に
    * 圏論の歩き方 （日本評論社） 今夏刊行予定