Lipschitz Continuous O.D.E.s are Polynomial-Space Complete

Akitoshi KAWAMURA

University of Toronto

24th IEEE Conference on Computational Complexity
July 16, 2009

Problem

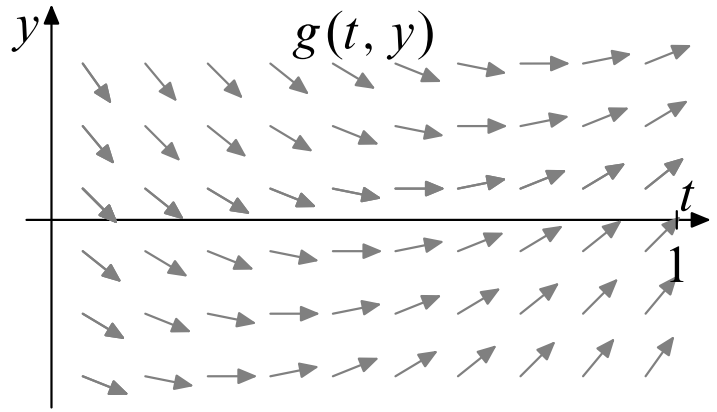
Given $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$, consider the IVP

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big).$$

Problem

Given $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$, consider the IVP

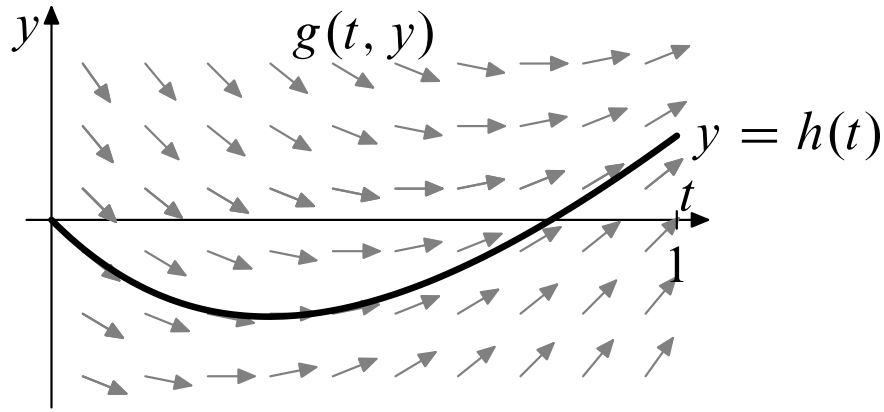
$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big).$$



Problem

Given $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$, consider the IVP

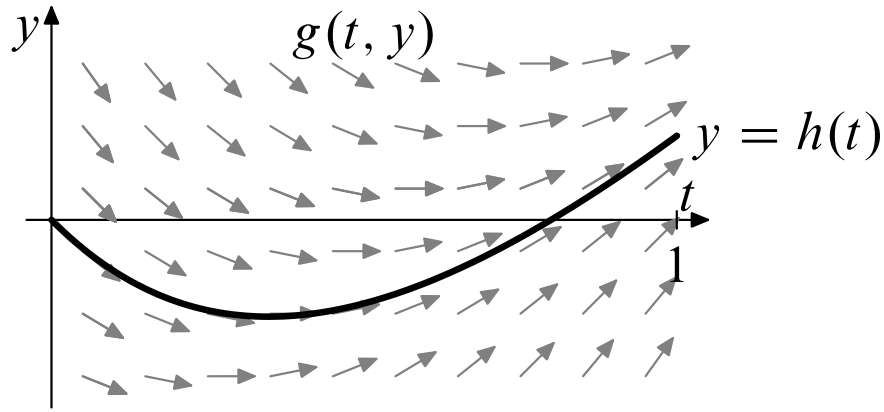
$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big).$$



Problem

Given $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$, consider the IVP

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big).$$



A sufficient condition for unique solution h is that

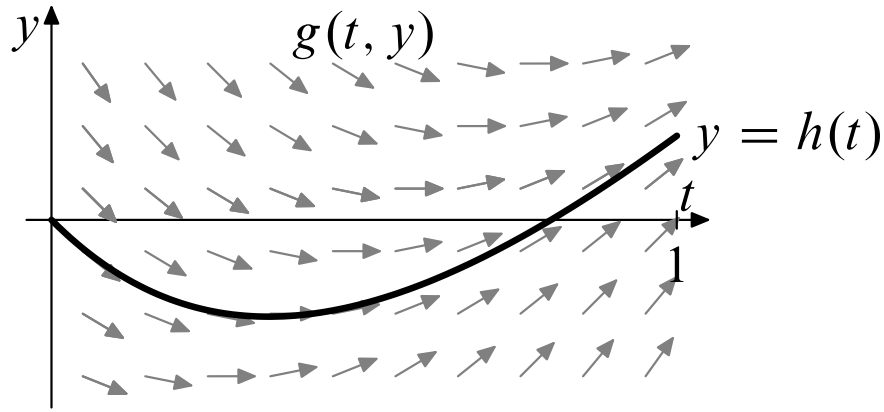
$$|g(t, y_0) - g(t, y_1)| \leq L|y_0 - y_1|, \qquad t \in [0,1],\ y_0, y_1 \in \mathbf{R}$$

for some $L \geq 0$ (Lipschitz continuity).

Problem

Given $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$, consider the IVP

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big).$$



A sufficient condition for unique solution h is that

$$|g(t, y_0) - g(t, y_1)| \leq L|y_0 - y_1|, \qquad t \in [0,1],\ y_0, y_1 \in \mathbf{R}$$

for some $L \geq 0$ (Lipschitz continuity).

Question

g is easy $\Longrightarrow$ h is easy? (in the sense of Computable Analysis)

Outline

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big)$$

1. Complexity of real functions
   How we define PTIME real functions
2. Warm-up: Integration
   Assuming g is PTIME and ignores the second argument, how complex can h be?
3. Lipschitz continuous IVP
   Assuming g is PTIME and Lipschitz continuous, how complex can h be?
4. Final remarks

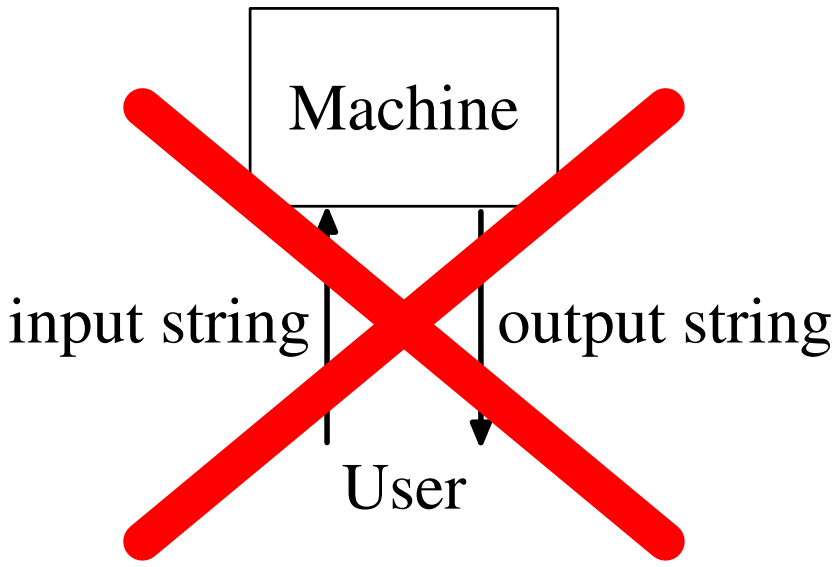
Outline

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big)$$

1. Complexity of real functions
   How we define PTIME real functions
2. Warm-up: Integration
   Assuming g is PTIME and ignores the second argument, how complex can h be?
3. Lipschitz continuous IVP
   Assuming g is PTIME and Lipschitz continuous, how complex can h be?
4. Final remarks

Read/write real numbers?

A real number cannot be encoded into a string.

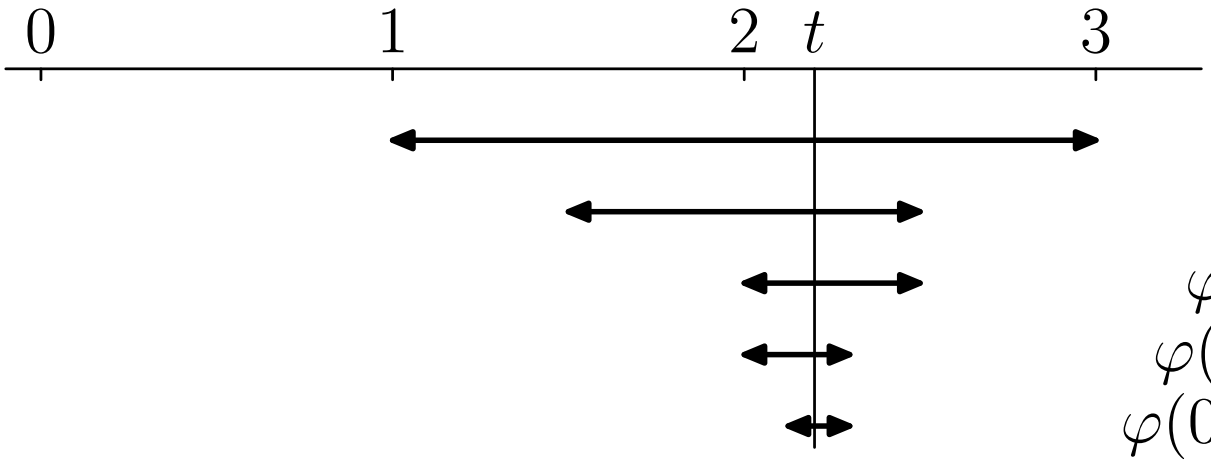


So we encode it into a function from strings to strings,
and give it to the machine as oracles.

Representation of real numbers

Definition

We say that $\varphi\colon \Sigma^* \to \Sigma^*$ is a name of $t \in \mathbf{R}$ if for each $n \in \mathbf{N}$, $\varphi(0^n)$ is (the binary expansion of) t rounded up or down at the nth bit below the point.



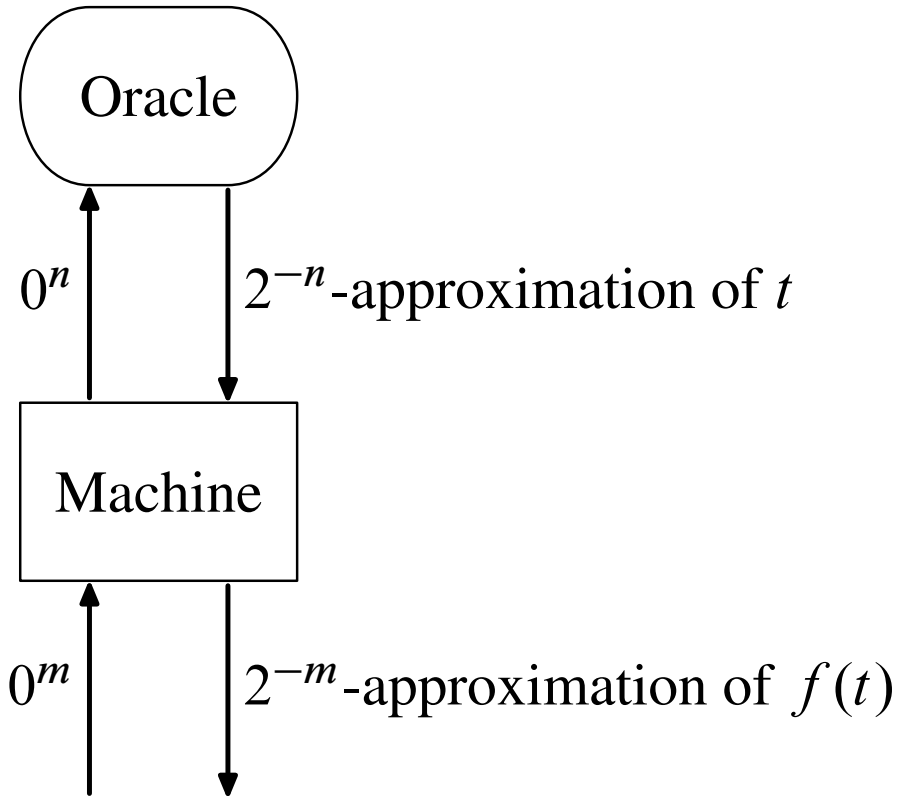
$$\begin{aligned}
\varphi(\varepsilon) &= +10. \\
\varphi(0) &= +10.0 \\
\varphi(00) &= +10.01 \\
\varphi(000) &= +10.001 \\
\varphi(0000) &= +10.0011 \\
&\vdots
\end{aligned}$$

(Each $\varphi(0^n)$ approximates t with precision 2^{-n})

Computing real functions

Definition (Grzegorzcyk 1955)

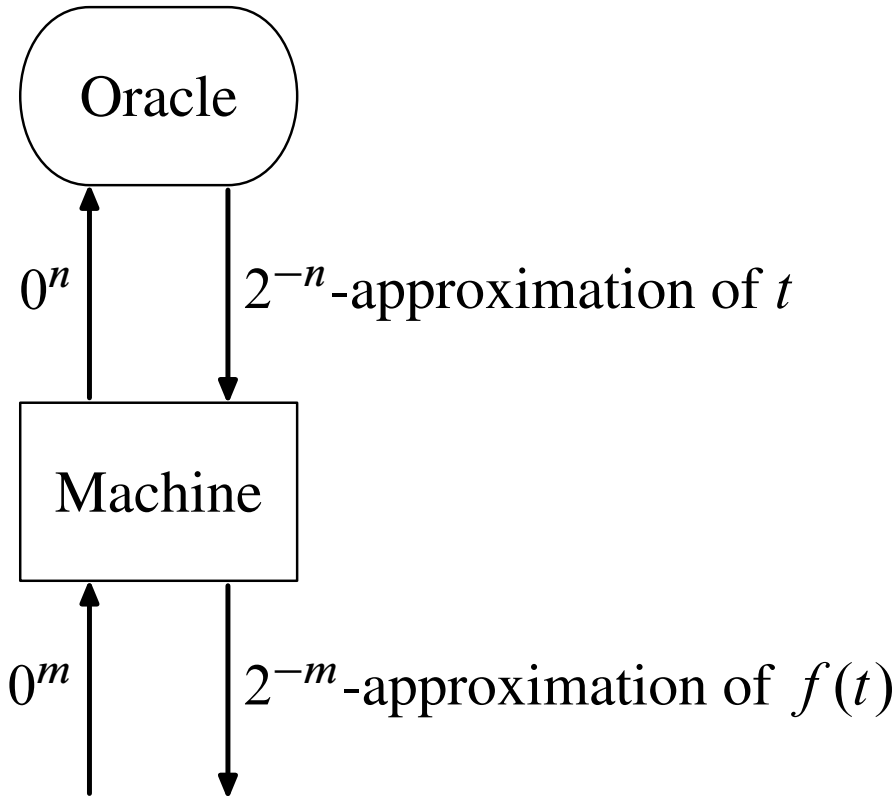
Machine M computes $f : [0,1] \to \mathbf{R}$ if, for any name φ of any $t \in [0,1]$, M^{φ} computes a name of $f(t)$.



Computing real functions

Definition (Grzegorzcyk 1955)

Machine M computes $f : [0,1] \to \mathbf{R}$ if, for any name φ of any $t \in [0,1]$, M^{φ} computes a name of $f(t)$.



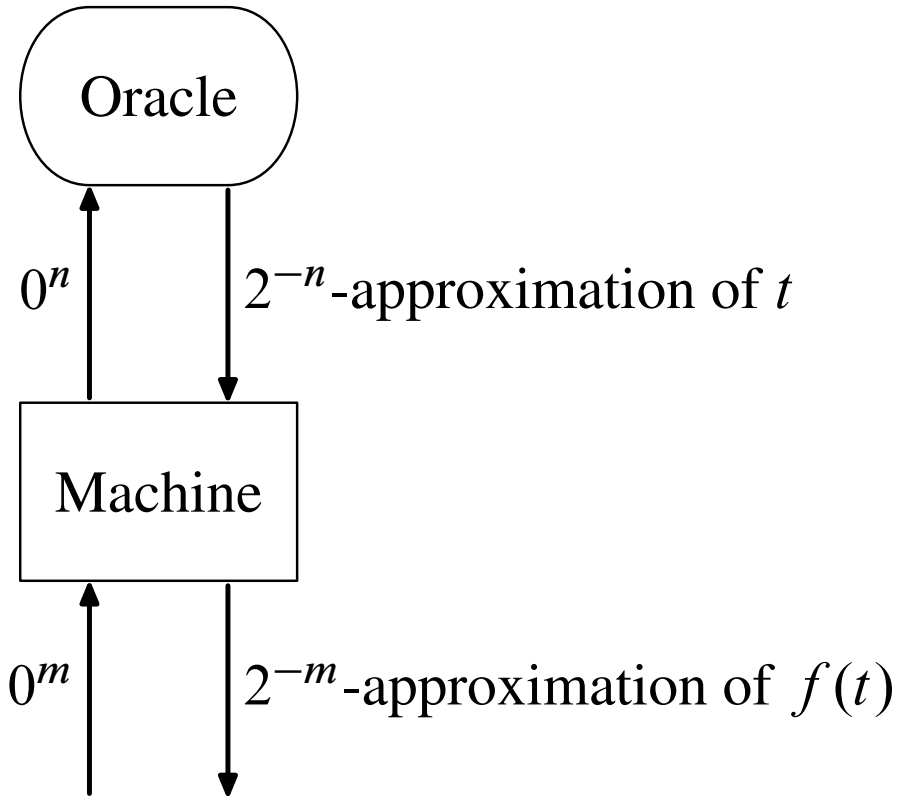
In other words,

- M Turing-reduces $f(t)$ to t.
- Given access to approximations of t to any precision, the machine yields $f(t)$ to any precision.

Computing real functions

Definition (Grzegorzcyk 1955)

Machine M computes $f : [0,1] \to \mathbf{R}$ if, for any name φ of any $t \in [0,1]$, M^{φ} computes a name of $f(t)$.



In other words,

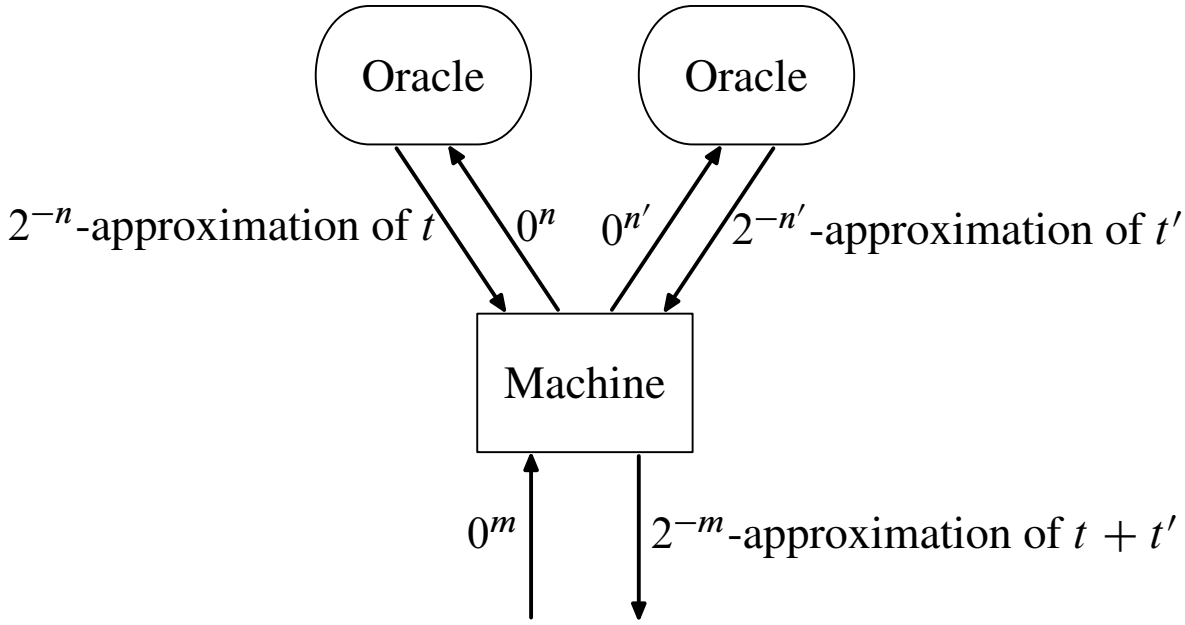
- M Turing-reduces $f(t)$ to t.
- Given access to approximations of t to any precision, the machine yields $f(t)$ to any precision.

Note: Saying that M is PTIME means that it halts in time poly(m). In particular, $n \leq$ poly(m).

Example 1

Addition $+\colon [0,1] \times [0,1] \to \mathbf{R}$ is PTIME.

To do: Given (names of) t and t' as oracles and 0^m as input, output a 2^{-m}-approximation of $t + t'$.



- Ask the oracles for 2^{-m-2}-approximations of t and t'.
- Add the answers (as rational numbers).
- Output the closest rational number to the sum that has m bits below the point.

Example 2

$\exp\colon [0,1] \to \mathbf{R}$ is PTIME.

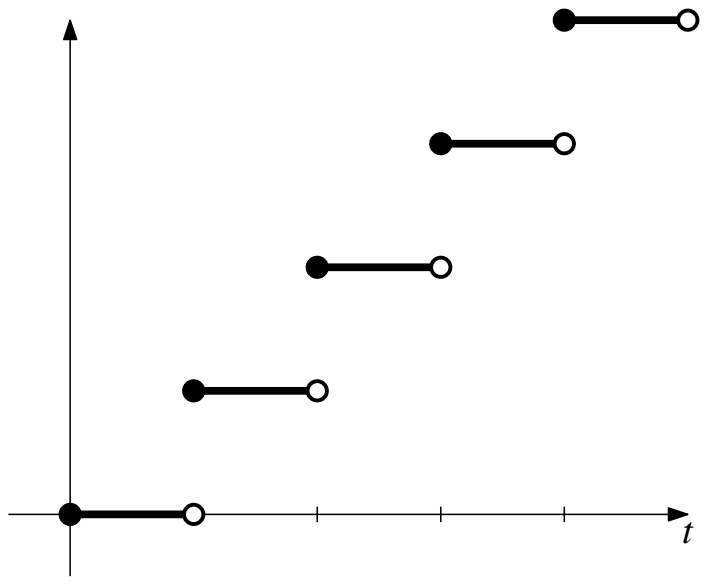
To get a 2^{-m}-approximation of the value

$$\exp t = \frac{1}{0!} + \frac{t}{1!} + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots,$$

it suffices to compute a $2^{-m}/2$-approximation of the sum of the first m terms (because the remaining terms add up to at most $2^{-m}/2$).

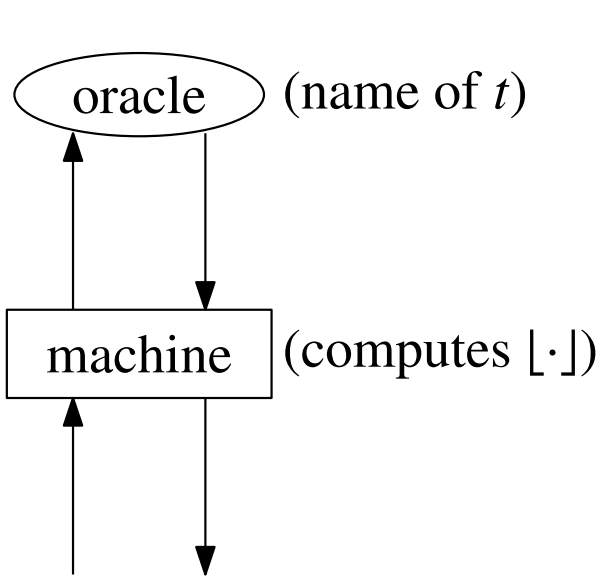
Example 3

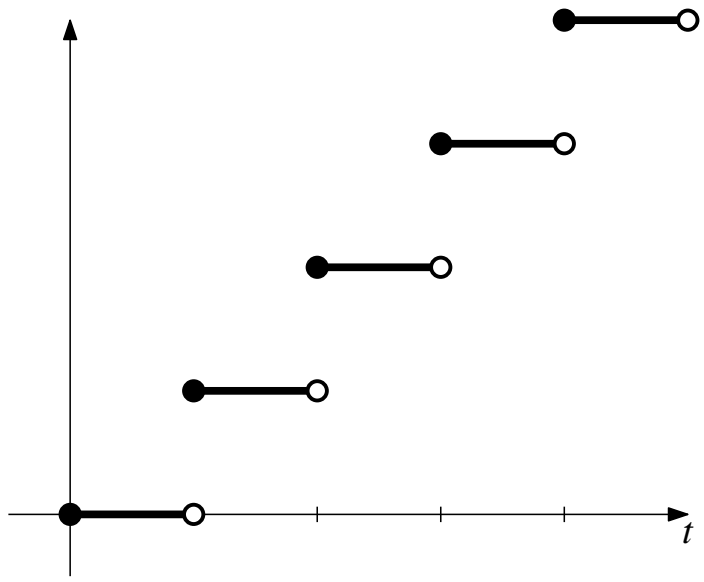
$\lfloor \cdot \rfloor : [0, 5] \longrightarrow \mathbf{R}$ is not computable.



Example 3

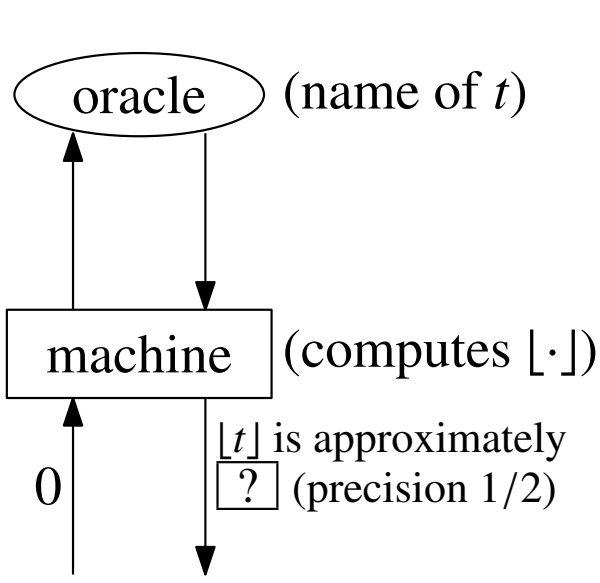
$\lfloor \cdot \rfloor : [0, 5] \to \mathbf{R}$ is not computable.

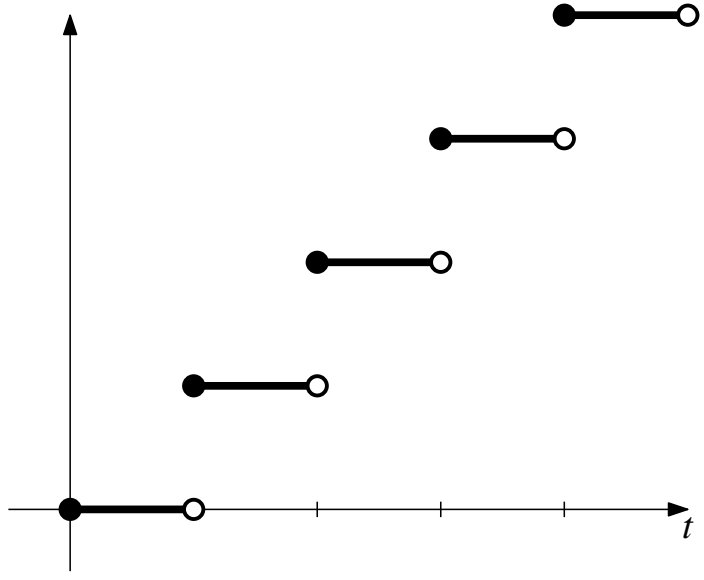




Example 3

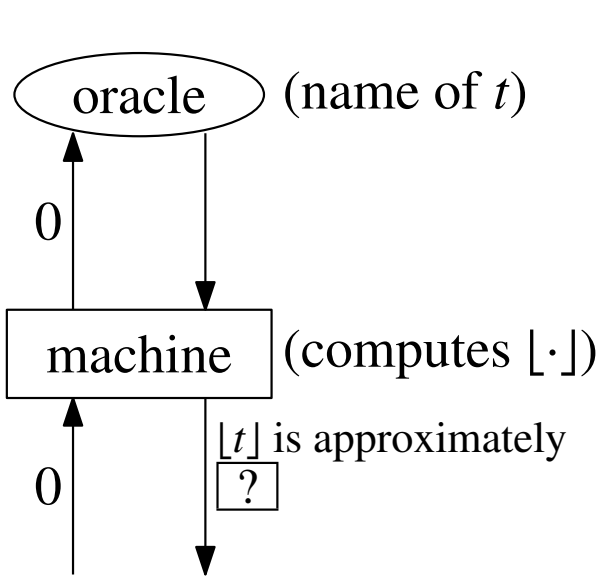
$\lfloor \cdot \rfloor : [0,5] \to \mathbf{R}$ is not computable.

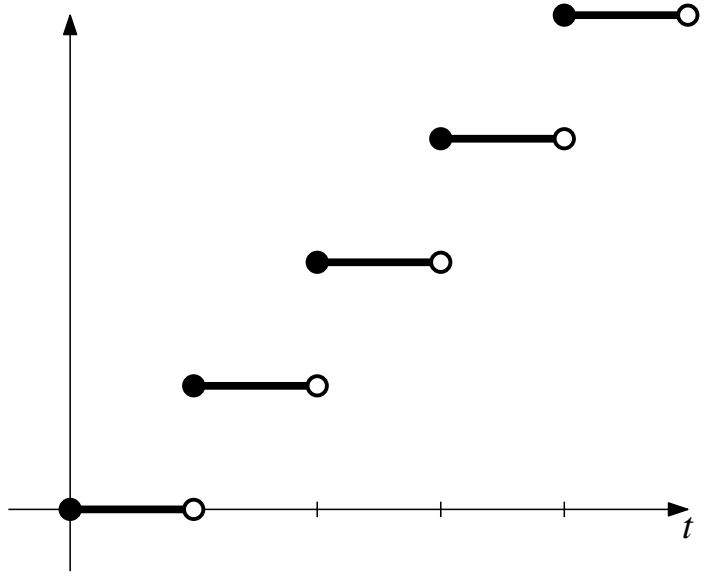




Example 3

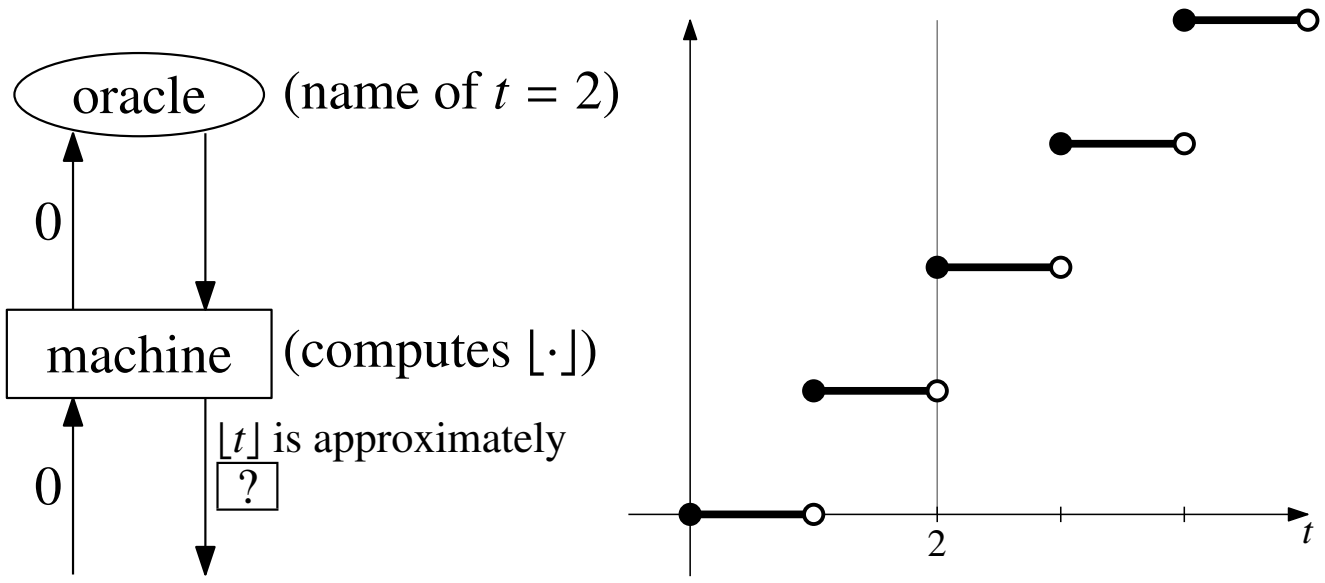
$\lfloor \cdot \rfloor : [0, 5] \to \mathbf{R}$ is not computable.





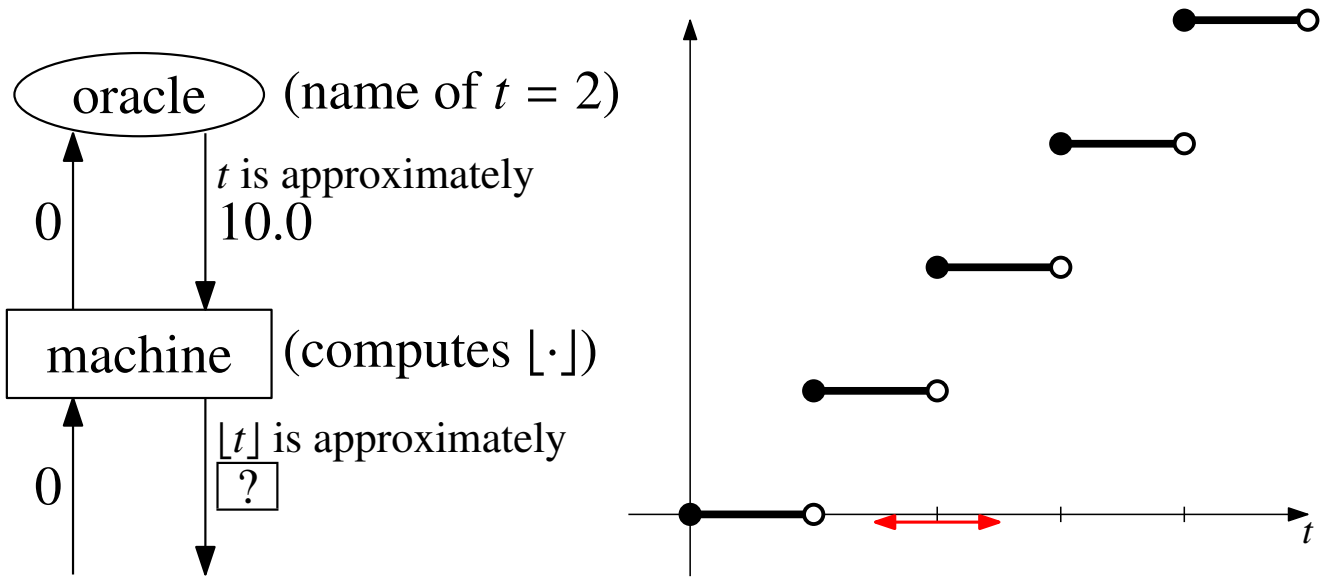
Example 3

$\lfloor \cdot \rfloor : [0, 5] \to \mathbf{R}$ is not computable.



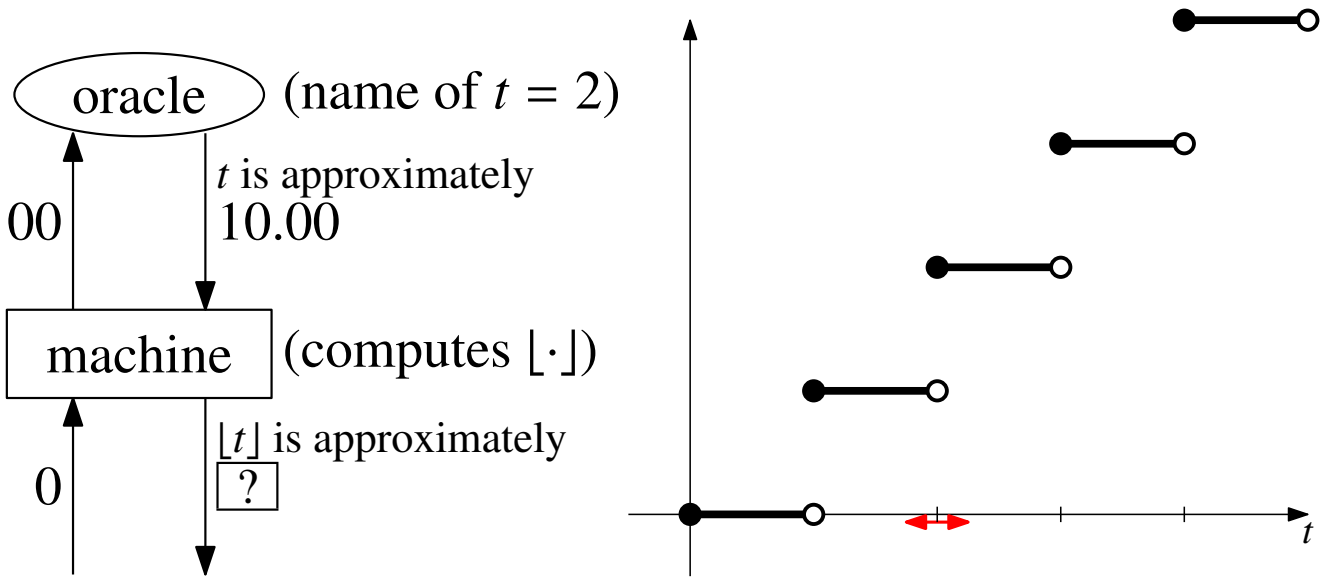
Example 3

$\lfloor \cdot \rfloor : [0,5] \to \mathbf{R}$ is not computable.



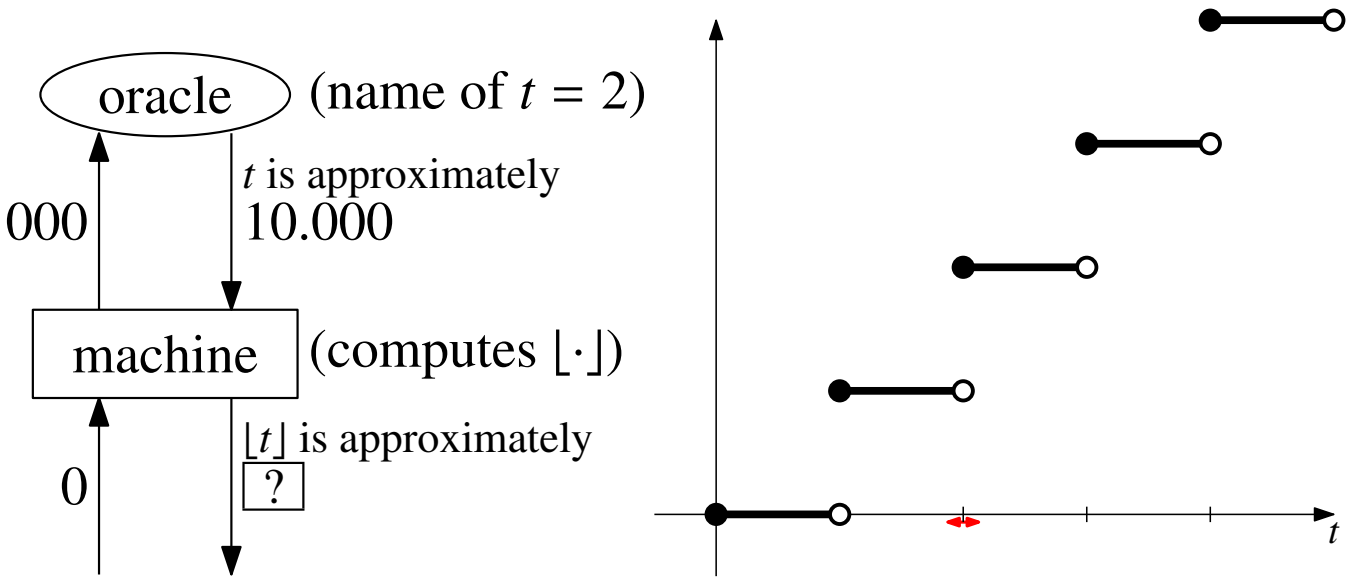
Example 3

$\lfloor \cdot \rfloor : [0, 5] \to \mathbf{R}$ is not computable.



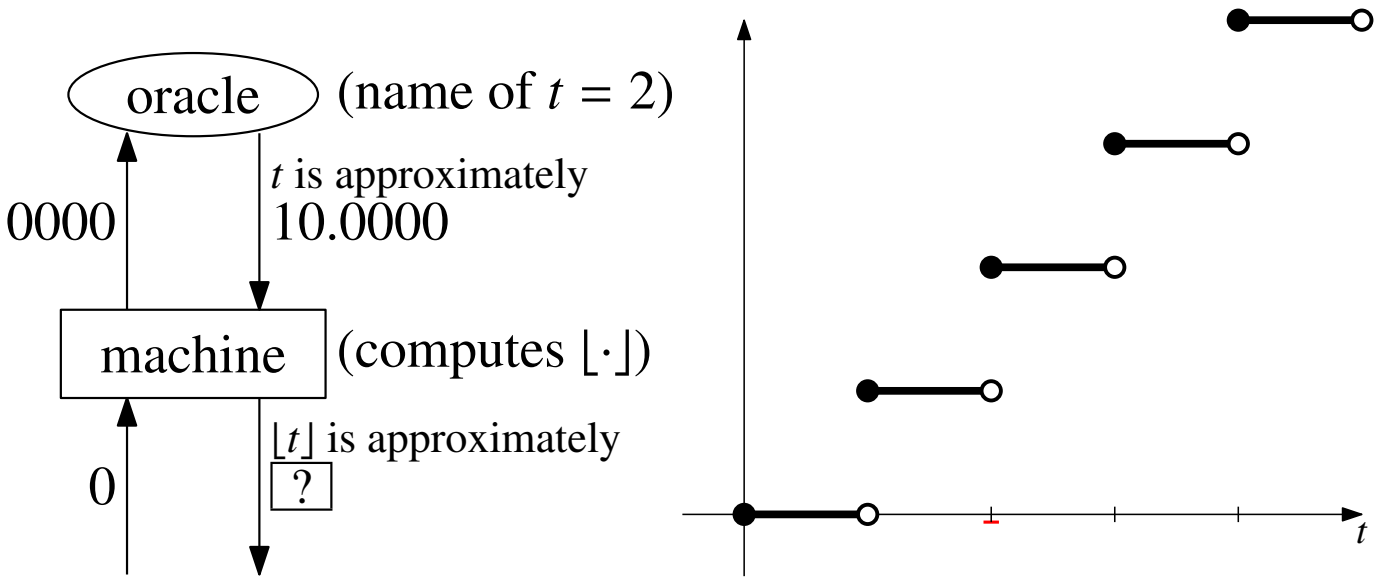
Example 3

$\lfloor \cdot \rfloor : [0,5] \to \mathbf{R}$ is not computable.



Example 3

$\lfloor \cdot \rfloor : [0,5] \to \mathbf{R}$ is not computable.



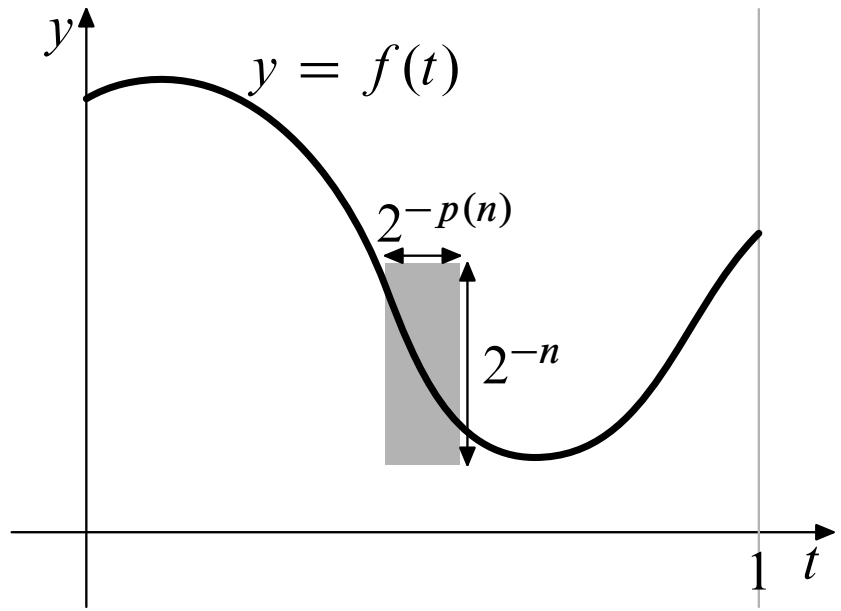
Computable functions are continuous

Theorem

- A computable real function is continuous.
- A PTIME real function has a polynomial modulus of continuity.

Modulus of continuity p:

$$|t - t'| < 2^{-p(n)} \implies |f(t) - f(t')| < 2^{-n}.$$



Outline

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big)$$

1. Complexity of real functions
   How we define PTIME real functions
2. Warm-up: Integration
   Assuming g is PTIME and ignores the second argument, how complex can h be?
3. Lipschitz continuous IVP
   Assuming g is PTIME and Lipschitz continuous, how complex can h be?
4. Final remarks

Complexity of integration

Theorem (essentially [Friedman 1984])

There are $g\colon [0,1] \to \mathbf{R}$ and $h\colon [0,1] \to \mathbf{R}$ such that

- g is PTIME;
- $h(0) = 0,\ \ h'(t) = g(t)$;
- h is #PTIME-hard.

Complexity of integration

Theorem (essentially [Friedman 1984])

There are $g\colon [0,1] \to \mathbf{R}$ and $h\colon [0,1] \to \mathbf{R}$ such that

- g is PTIME;
- $h(0) = 0,\ \ h'(t) = g(t)$; — special case of $h'(t) = g(t, h(t))$ where g ignores its second argument
- h is #PTIME-hard.

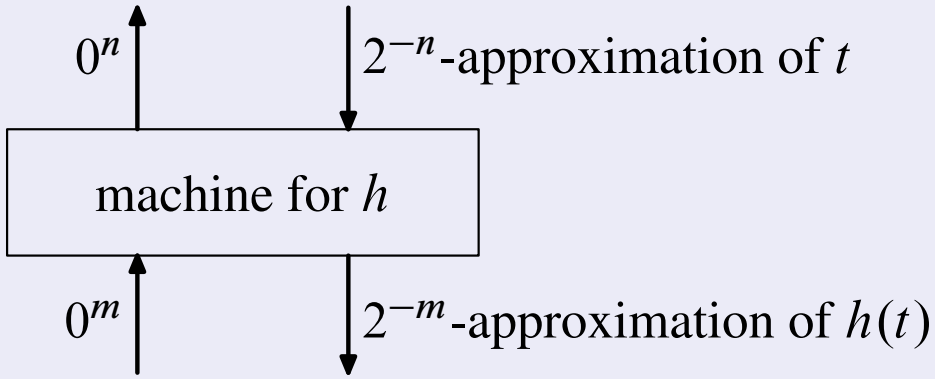
Complexity of integration

Theorem (essentially [Friedman 1984])

There are $g\colon [0,1] \to \mathbf{R}$ and $h\colon [0,1] \to \mathbf{R}$ such that

- g is PTIME;
- $h(0) = 0,\ \ h'(t) = g(t)$;
- h is #PTIME-hard, in the sense that

special case of $h'(t) = g(t, h(t))$ where g ignores its second argument

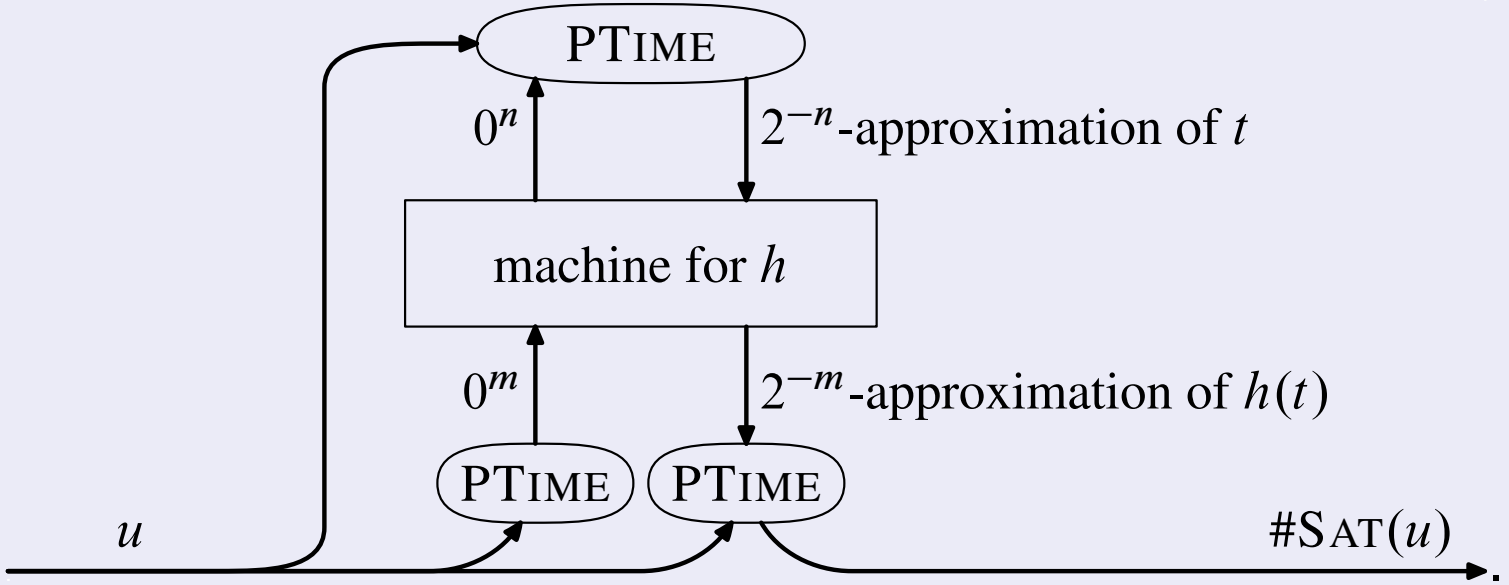


Complexity of integration

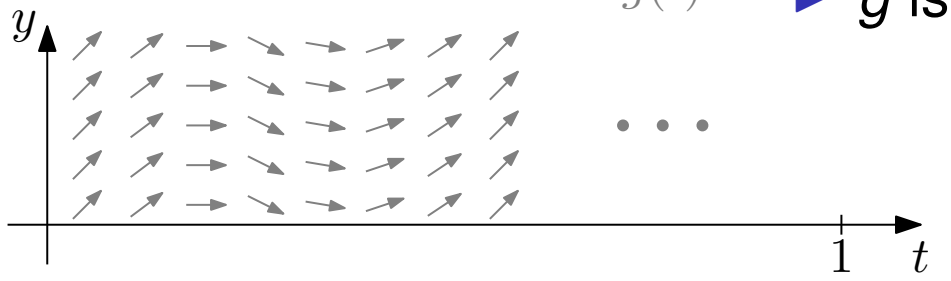
Theorem (essentially [Friedman 1984])

There are $g\colon [0,1] \to \mathbf{R}$ and $h\colon [0,1] \to \mathbf{R}$ such that

- g is PTIME;
- $h(0) = 0,\ \ h'(t) = g(t)$; — special case of $h'(t) = g(t, h(t))$ where g ignores its second argument
- h is #PTIME-hard, in the sense that



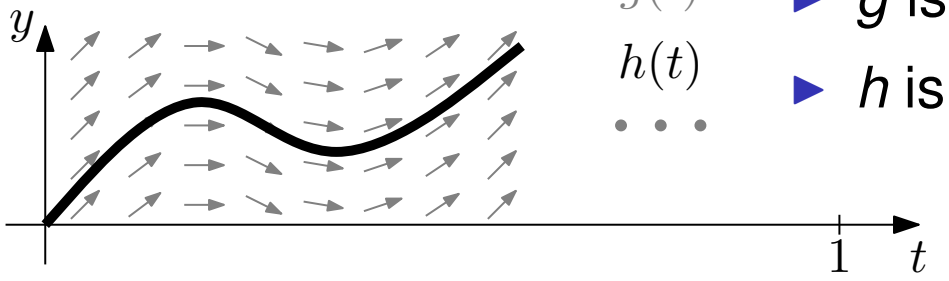
Reducing #SAT to integration



What we want:

- g is easy (PTIME);

Reducing #SAT to integration



$g(t)$

$h(t)$

• • •

What we want:

- g is easy (PTIME);
- h is hard (#SAT reduces to it).

Reducing #SAT to integration

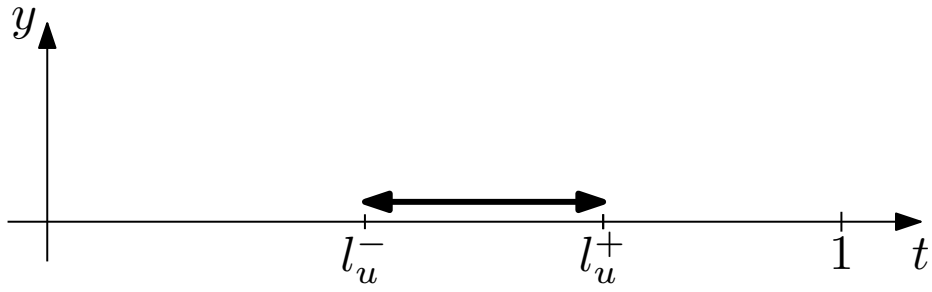
Reducing #SAT to integration

$\#\text{SAT}(u) =$ number of v
with $(u, v) \in R$
(for some $R \in \text{PTIME}$).

Reducing #SAT to integration

$\#\text{SAT}(u) =$ number of v
with $(u, v) \in R$
(for some $R \in \text{PTIME}$).

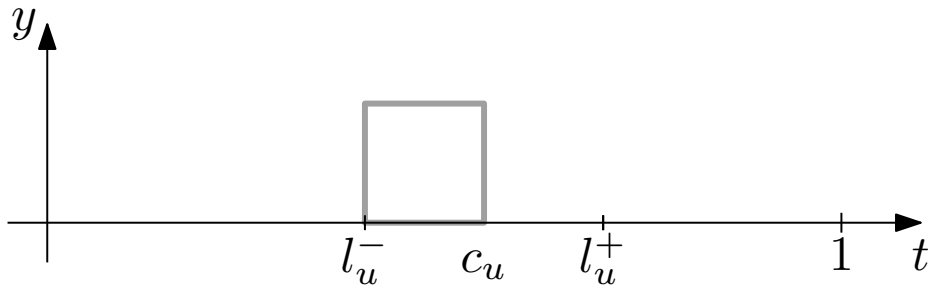
For each $u \in \Sigma^*$,
assign interval $[l_u^-, l_u^+]$.



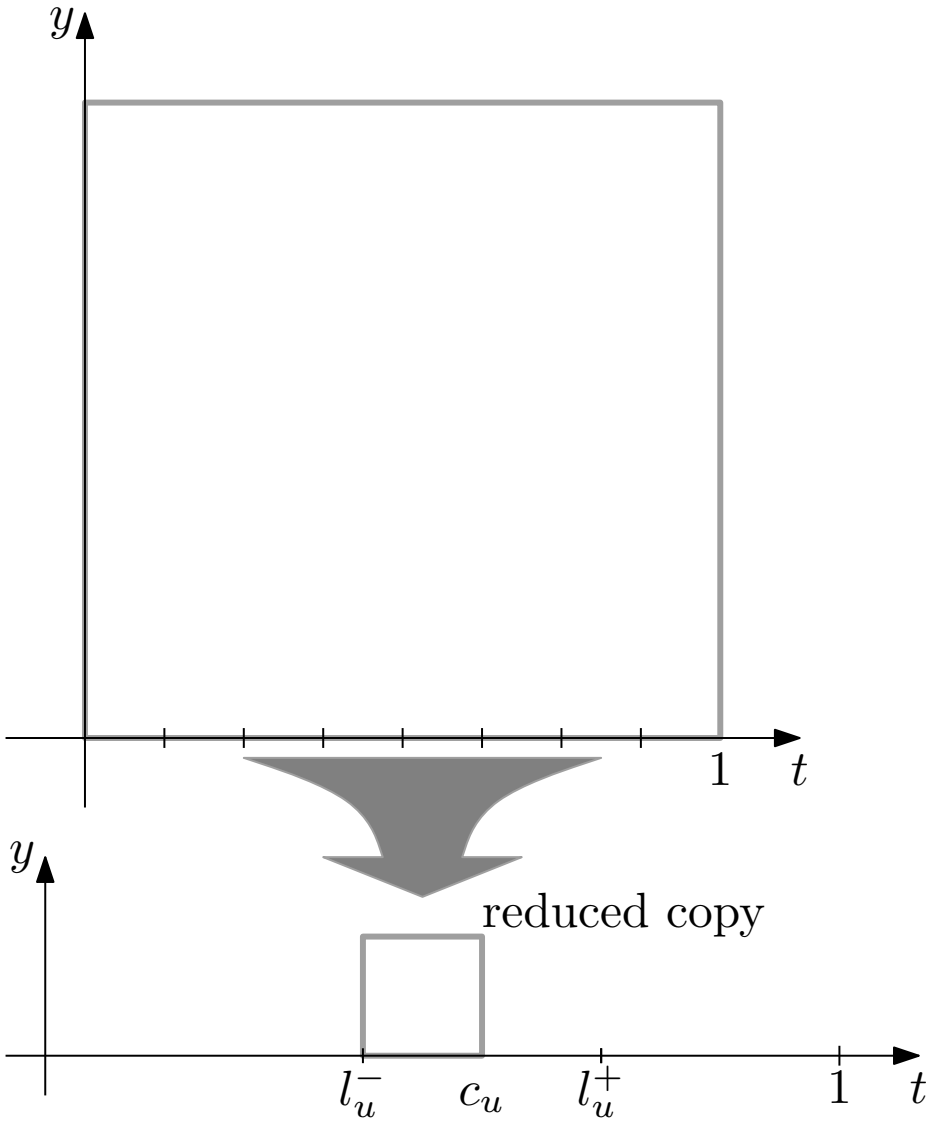
Reducing #SAT to integration

$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$.



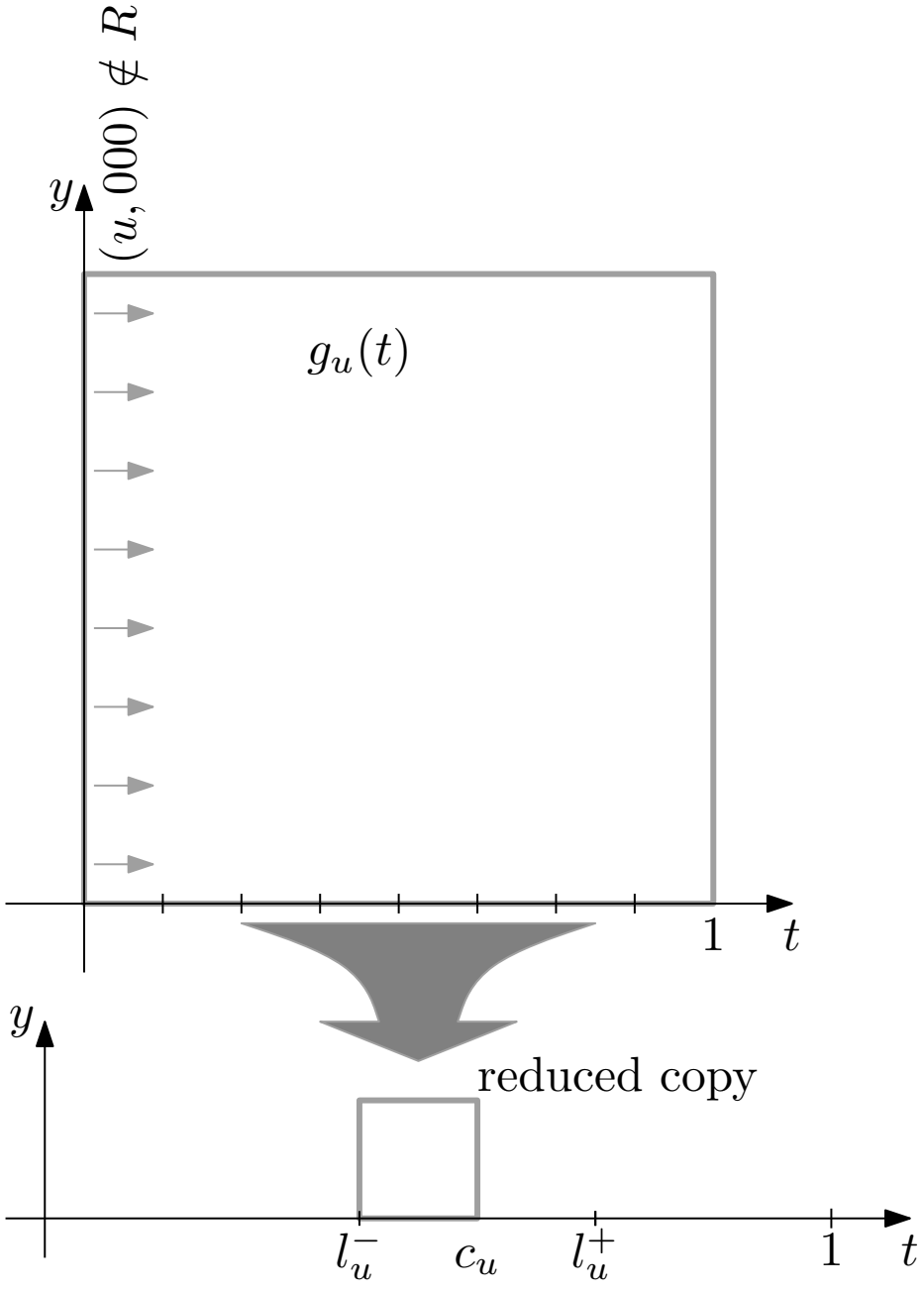
Reducing #SAT to integration



$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$. Put there a pair of reduced copies of g_u.

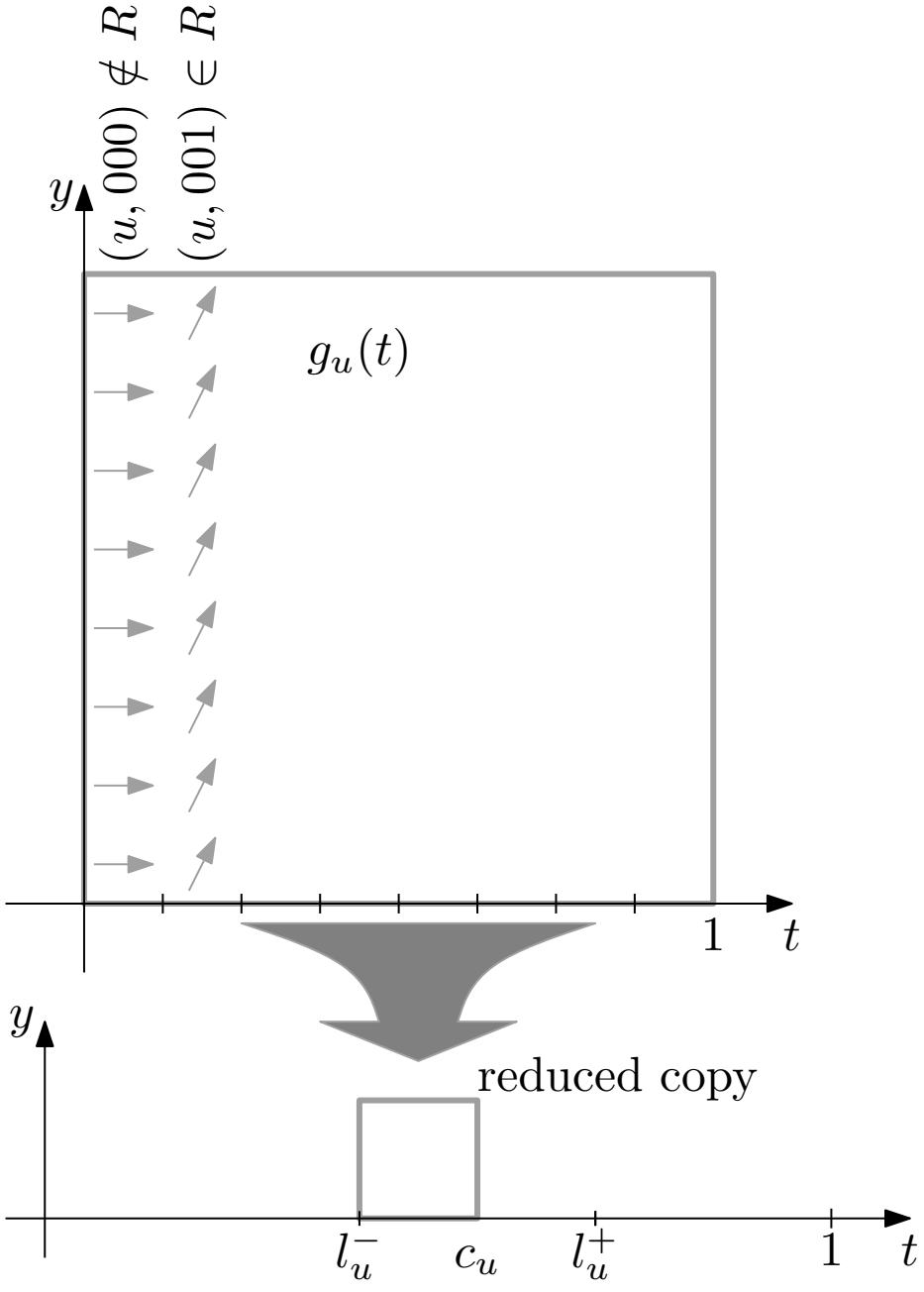
Reducing #SAT to integration



$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[I_u^-, I_u^+]$. Put there a pair of reduced copies of g_u.

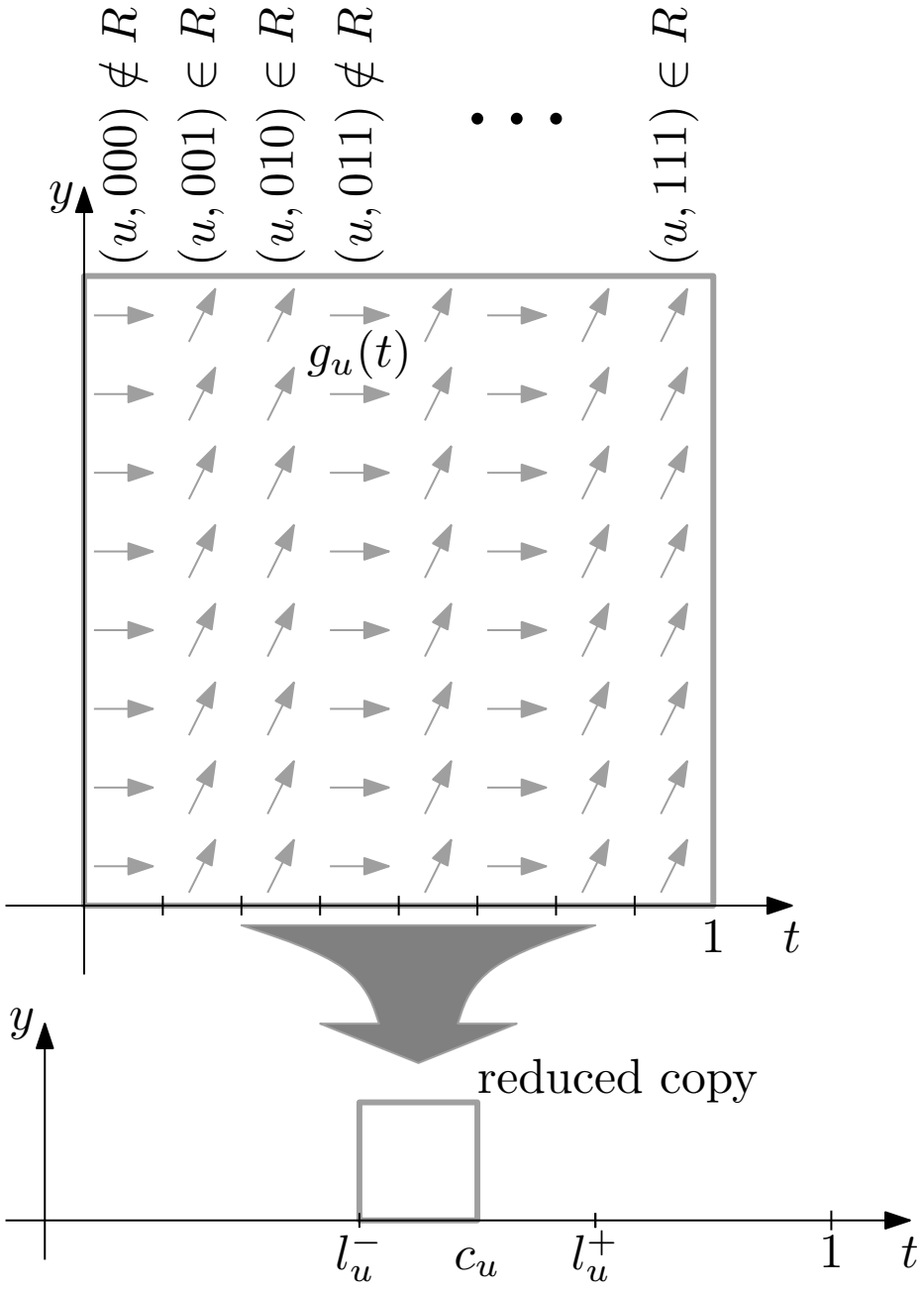
Reducing #SAT to integration



$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[I_u^-, I_u^+]$. Put there a pair of reduced copies of g_u.

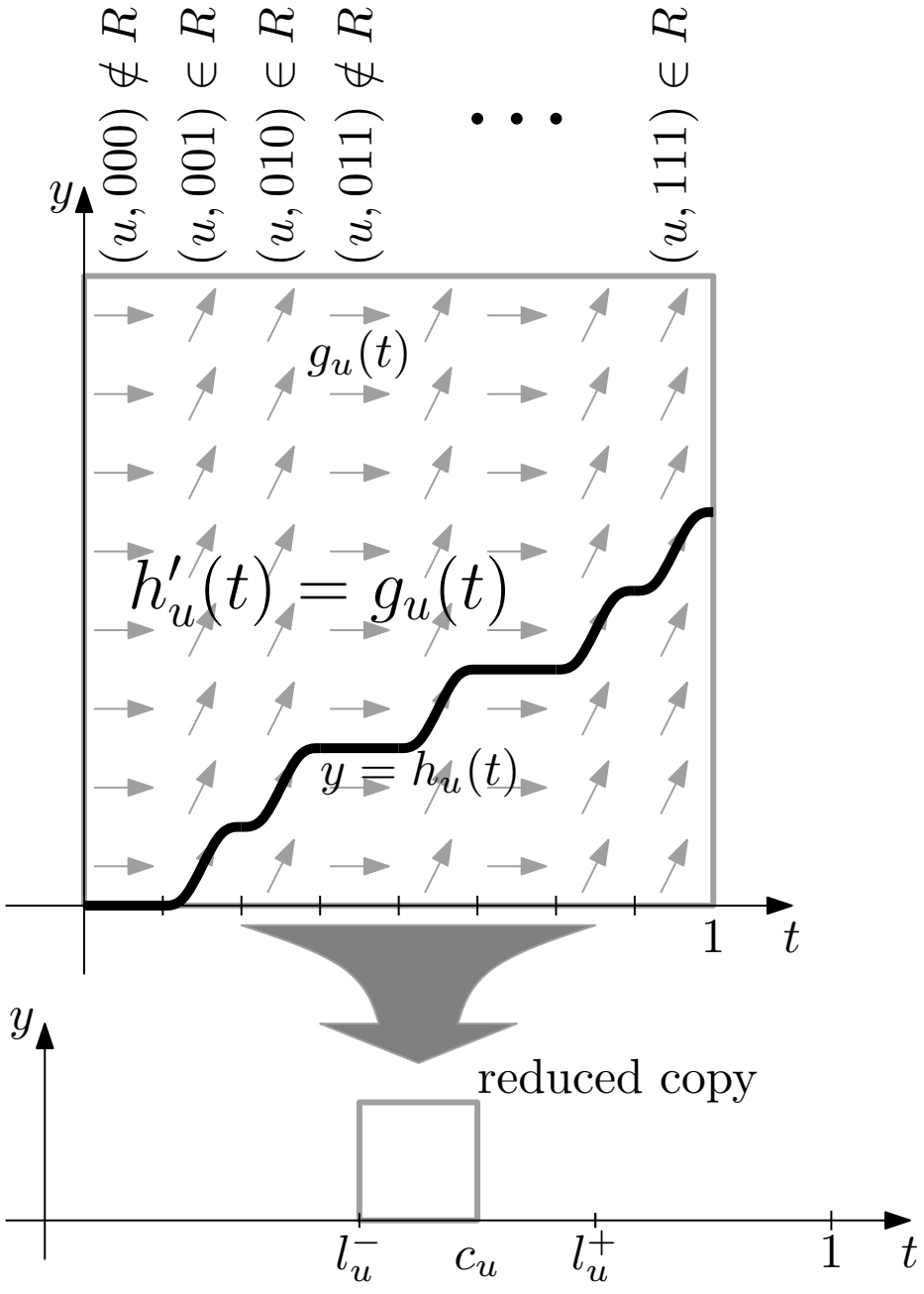
Reducing #SAT to integration



$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[I_u^-, I_u^+]$. Put there a pair of reduced copies of g_u.

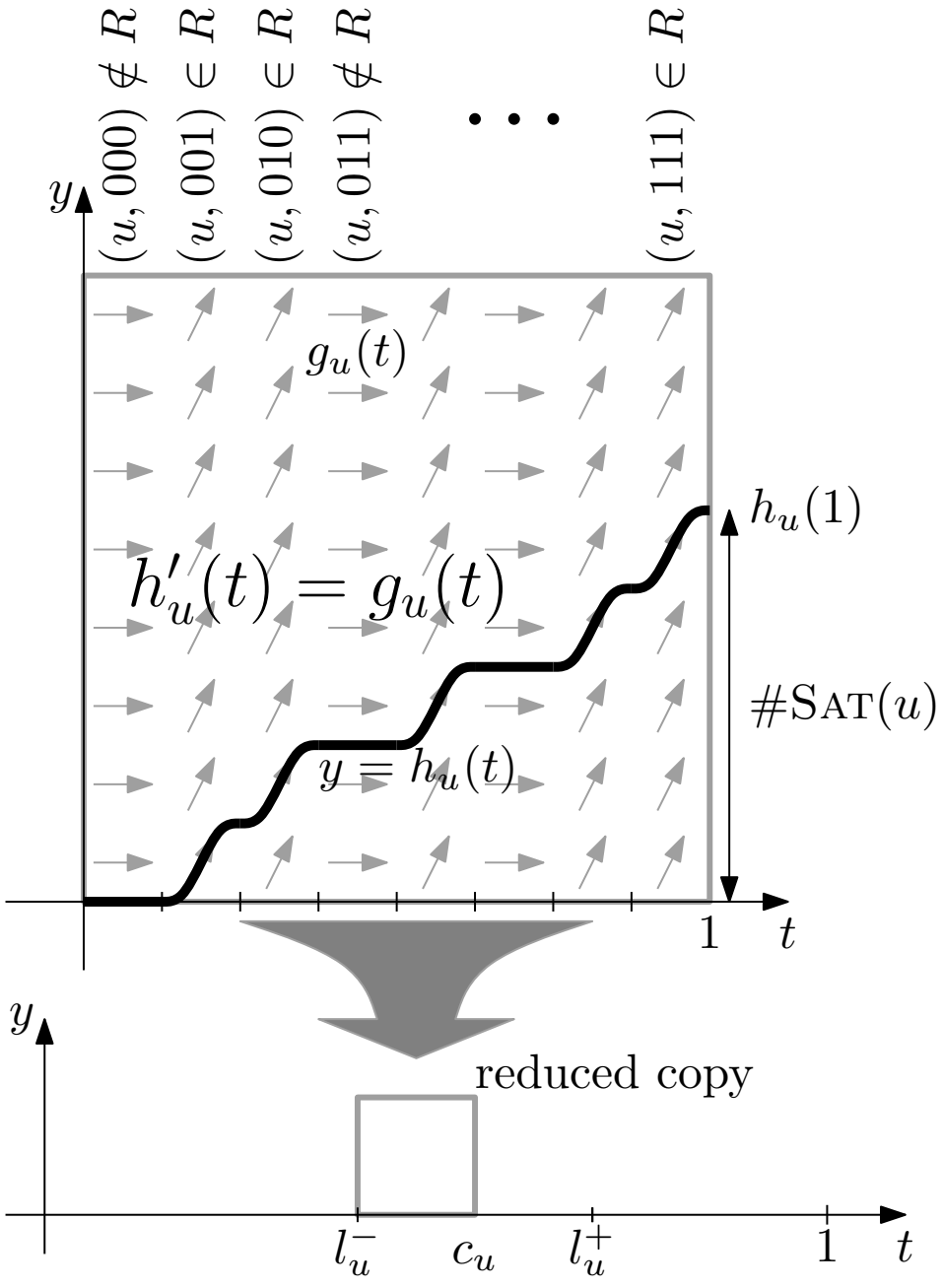
Reducing #SAT to integration



#SAT(u) = number of v with $(u, v) \in R$ (for some $R \in$ PTIME).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$. Put there a pair of reduced copies of g_u.

Reducing #SAT to integration

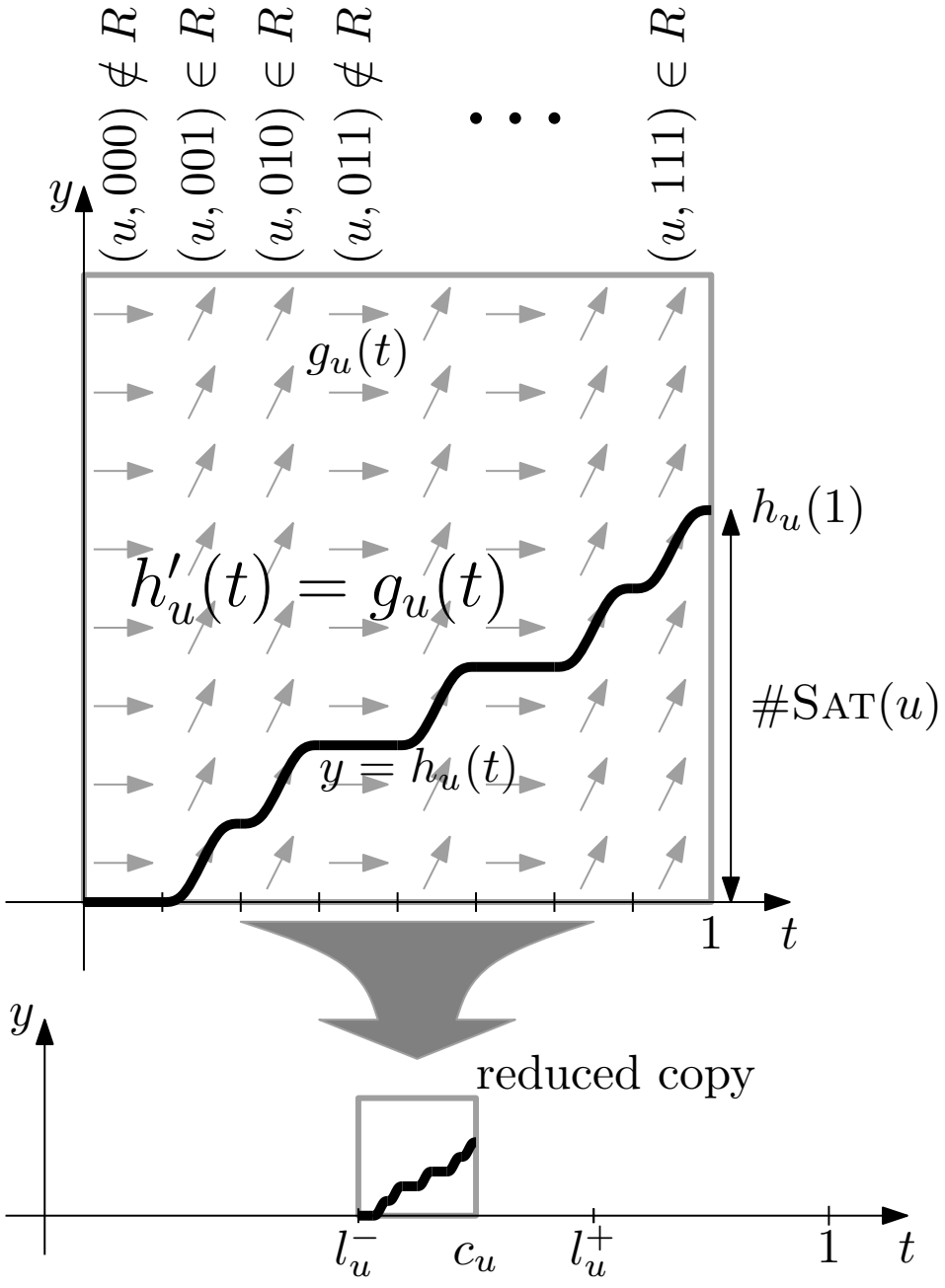


$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$. Put there a pair of reduced copies of g_u.

$\#\text{SAT}(u)$ is encoded in $h_u(1)$.

Reducing #SAT to integration

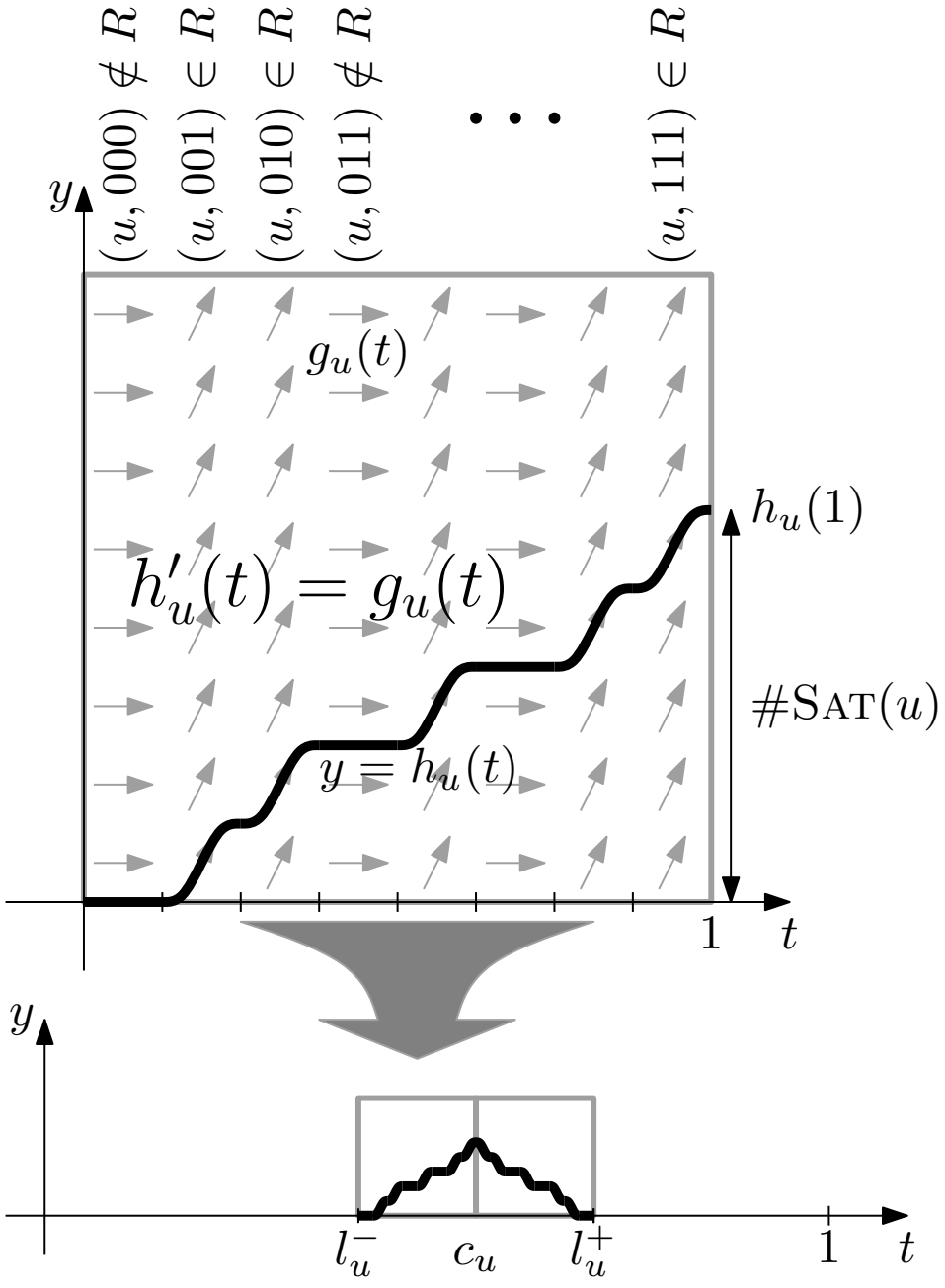


$\#\textsc{Sat}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \textsc{PTime}$).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$. Put there a pair of reduced copies of g_u.

$\#\textsc{Sat}(u)$ is encoded in $h_u(1)$, or in $h(c_u)$.

Reducing #SAT to integration

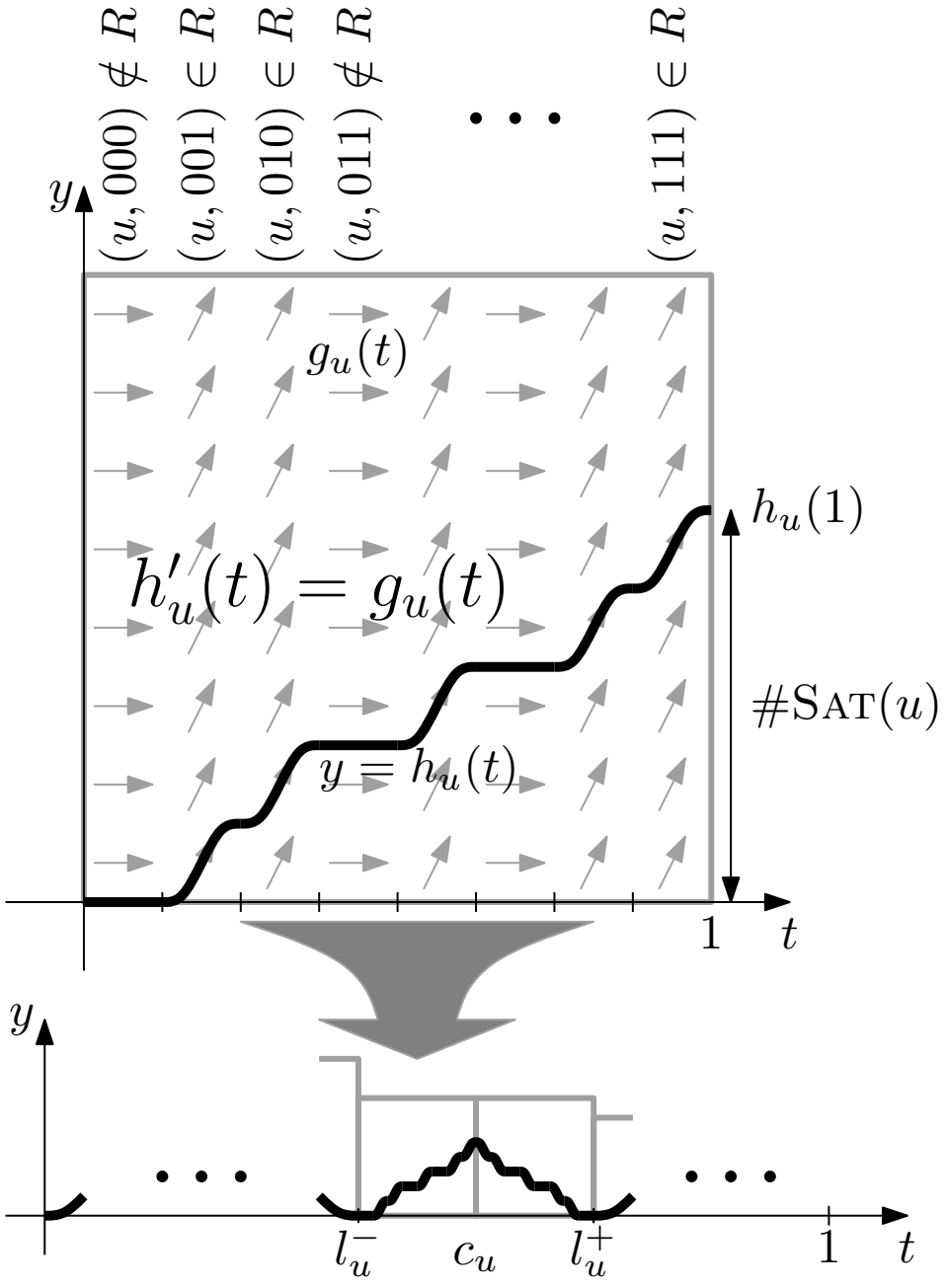


$\#\textsc{Sat}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \textsc{PTime}$).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$. Put there a pair of reduced copies of g_u.

$\#\textsc{Sat}(u)$ is encoded in $h_u(1)$, or in $h(c_u)$.

Reducing #SAT to integration



$\#\text{SAT}(u) =$ number of v with $(u, v) \in R$ (for some $R \in \text{PTIME}$).

For each $u \in \Sigma^*$, assign interval $[l_u^-, l_u^+]$. Put there a pair of reduced copies of g_u.

$\#\text{SAT}(u)$ is encoded in $h_u(1)$, or in $h(c_u)$.

Outline

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big)$$

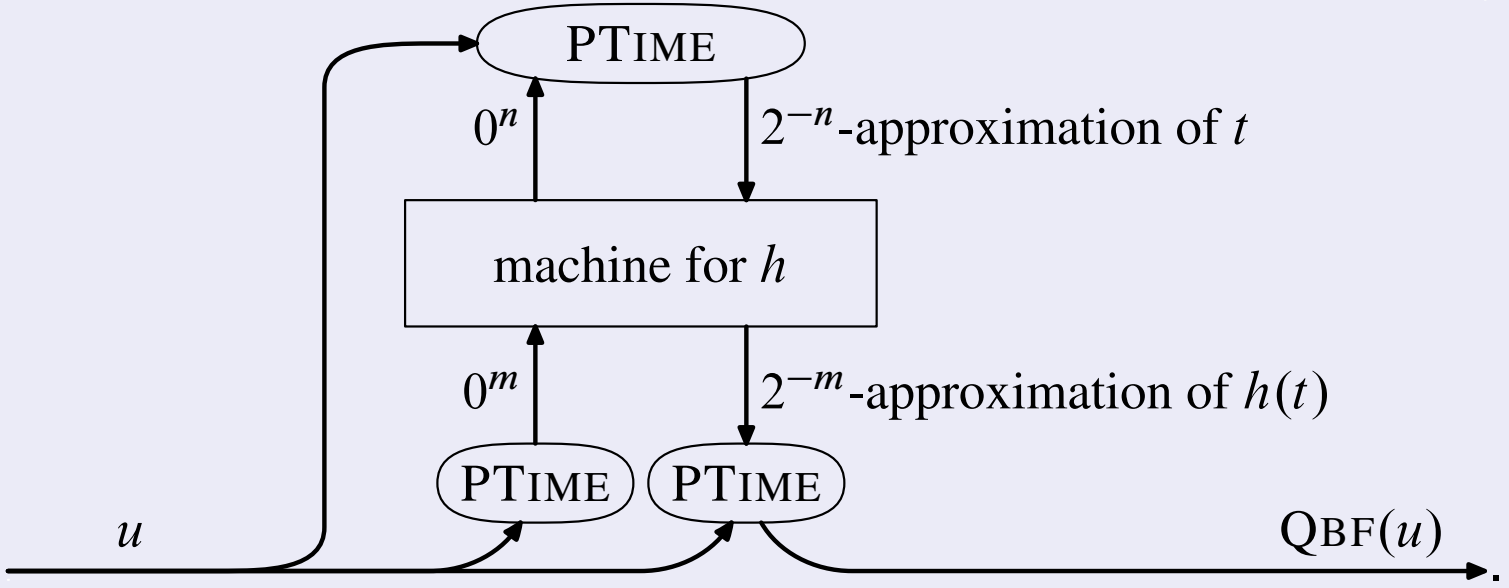
1. Complexity of real functions
   How we define PTIME real functions
2. Warm-up: Integration
   Assuming g is PTIME and ignores the second argument, how complex can h be?
3. Lipschitz continuous IVP
   Assuming g is PTIME and Lipschitz continuous, how complex can h be?
4. Final remarks

Complexity of Lipschitz IVPs

Main Theorem

There are $g\colon [0,1]\times[-1,1]\to \mathbf{R}$ and $h\colon [0,1]\to[-1,1]$ such that

- g is PTIME and Lipschitz continuous;
- $h(0)=0, \quad h'(t)=g\big(t,h(t)\big)$;
- h is PSPACE-hard in the sense that

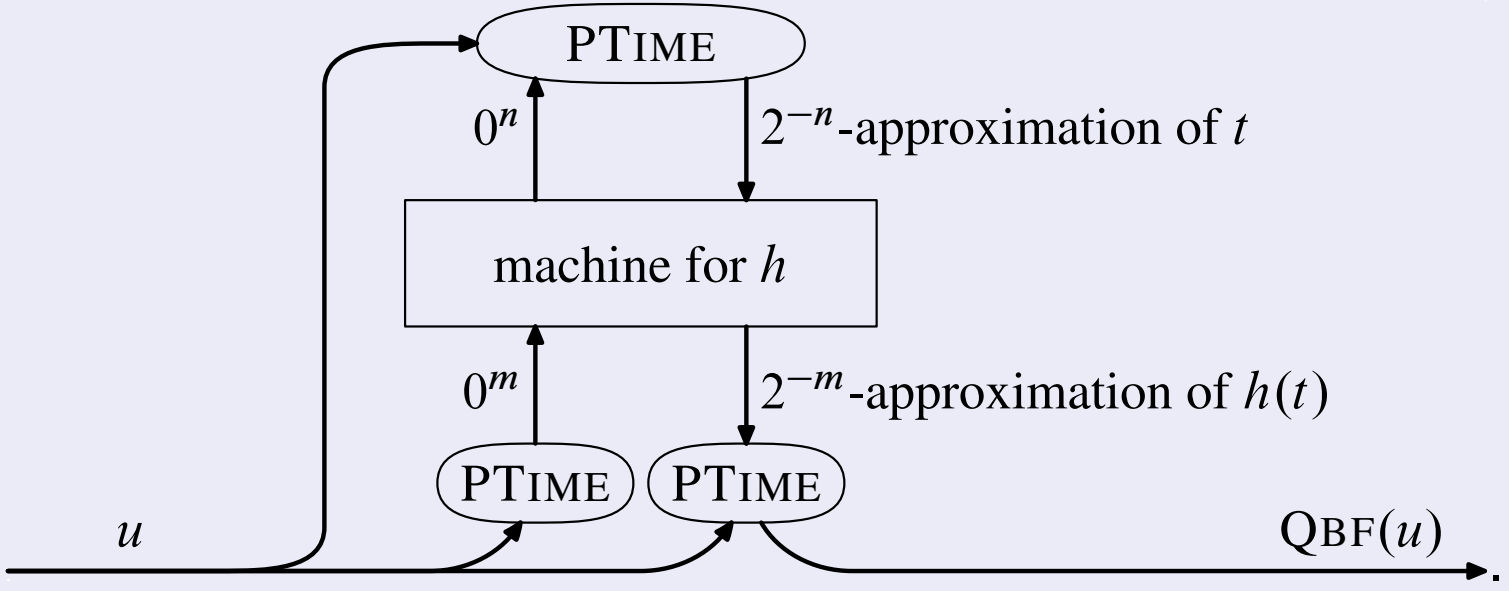


Complexity of Lipschitz IVPs

Main Theorem

There are $g\colon [0,1]\times[-1,1]\to \mathbf{R}$ and $h\colon [0,1]\to[-1,1]$ such that

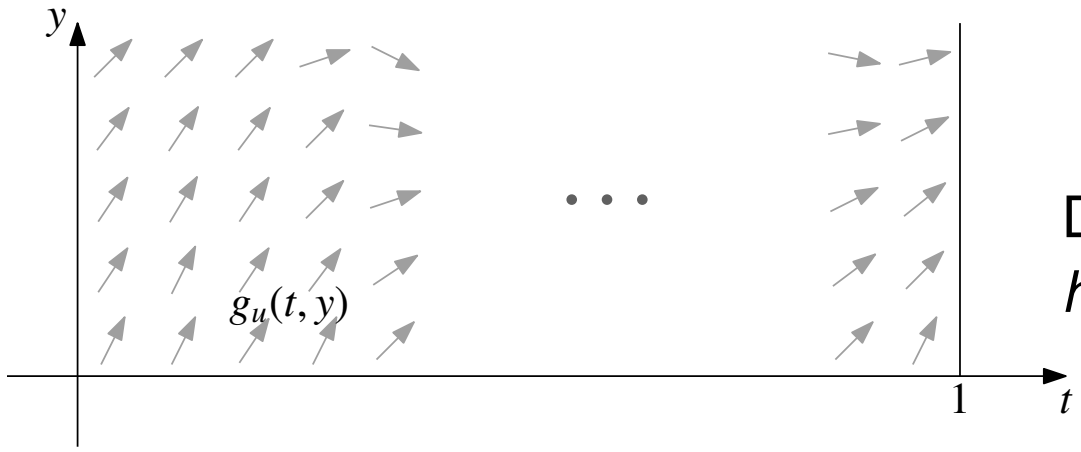
- g is PTIME and Lipschitz continuous;
- $h(0)=0, \quad h'(t)=g\big(t,h(t)\big)$;
- h is PSPACE-hard in the sense that



Cf. Upper bound: h is in PSPACE by the Euler method [Ko 1983].

Proof (1/3): An attempt to reduce PSPACE to IVP

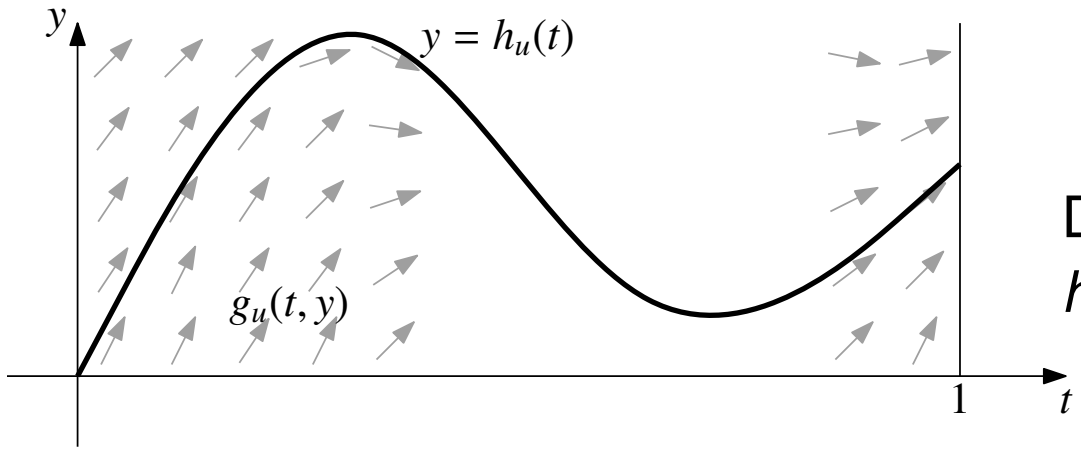
As before, we need blocks g_u such that $h_u(1)$ indicates if $u \in$ QBF.



Differential equation:
$h'_u(t) = g_u\big(t, h(t)\big)$

Proof (1/3): An attempt to reduce PSPACE to IVP

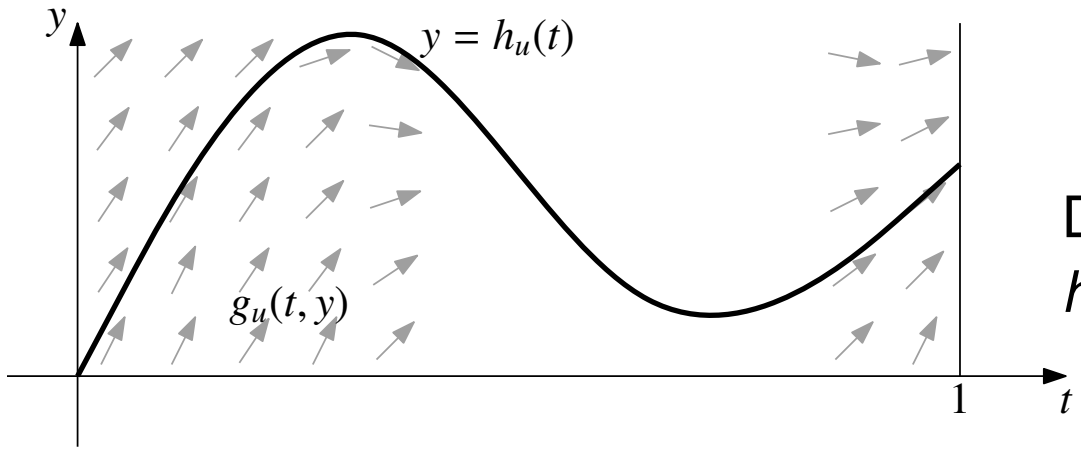
As before, we need blocks g_u such that $h_u(1)$ indicates if $u \in$ QBF.



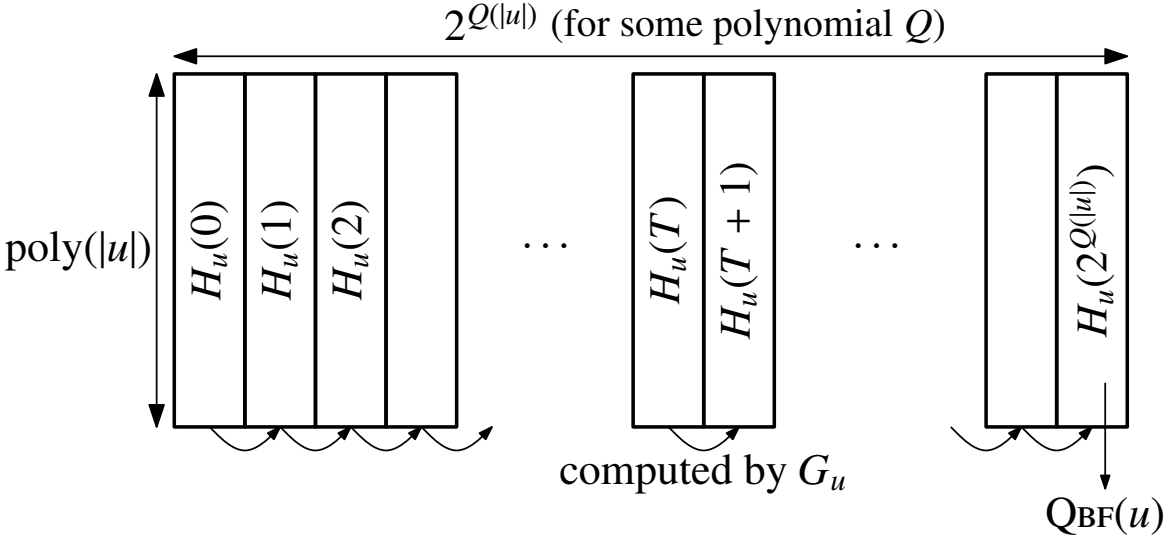
Differential equation:
$h'_u(t) = g_u\big(t, h(t)\big)$

Proof (1/3): An attempt to reduce PSPACE to IVP

As before, we need blocks g_u such that $h_u(1)$ indicates if $u \in$ QBF.



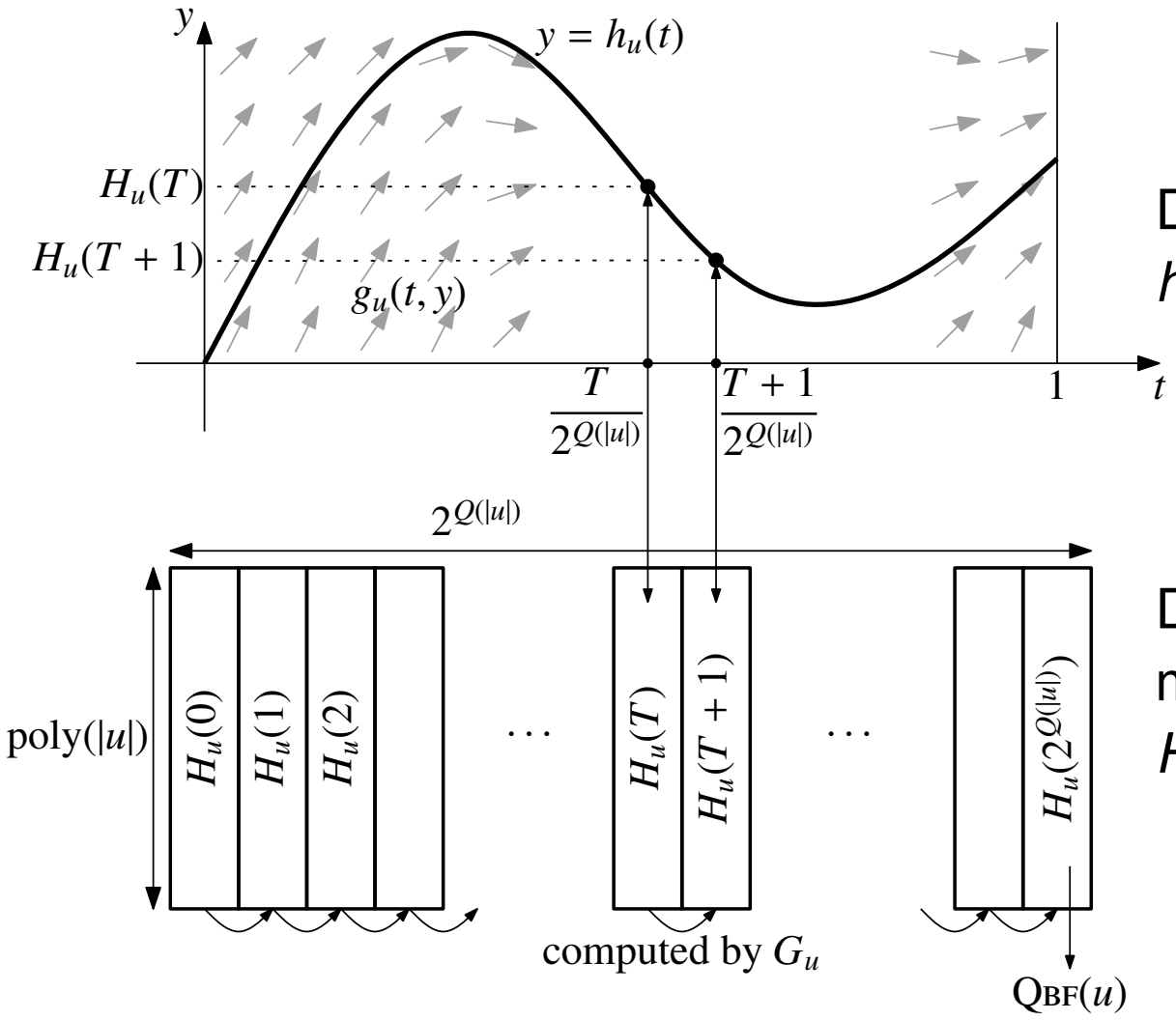
Differential equation:
$h'_u(t) = g_u\big(t, h(t)\big)$



Description of a PSPACE machine on input u:
$H_u(T+1) = G_u(T, H_u(T))$

Proof (1/3): An attempt to reduce PSPACE to IVP

As before, we need blocks g_u such that $h_u(1)$ indicates if $u \in$ QBF.



Differential equation:

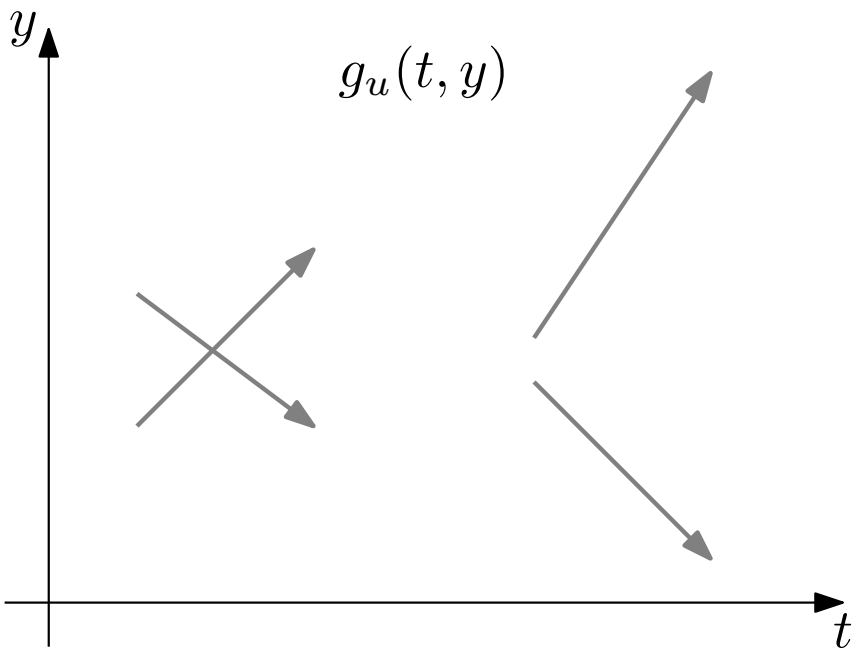
$$h'_u(t) = g_u\big(t, h(t)\big)$$

Description of a PSPACE machine on input u:

$$H_u(T+1) = G_u(T, H_u(T))$$

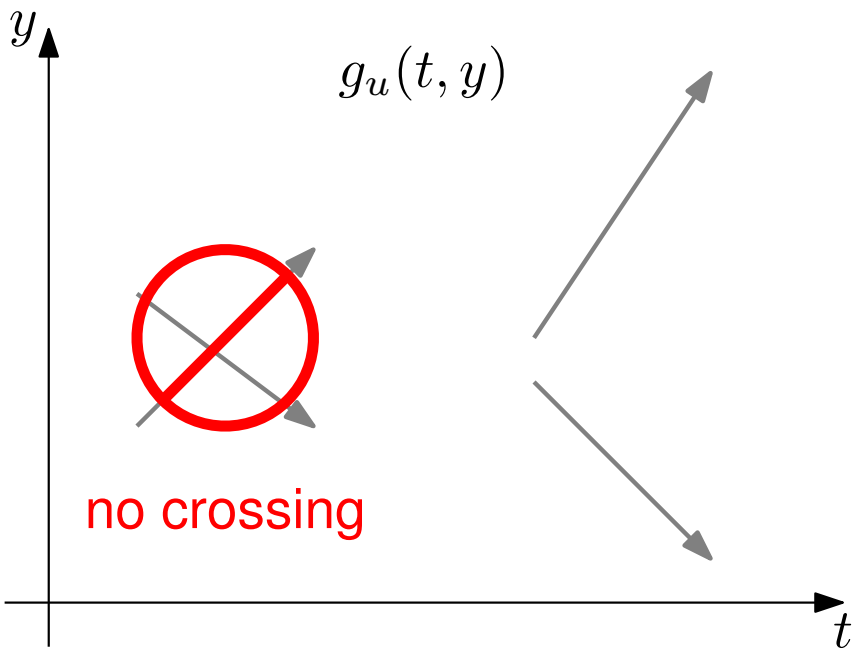
Proof (2/3): Why this attempt does not work

Not all G_u can be translated to g_u.



Proof (2/3): Why this attempt does not work

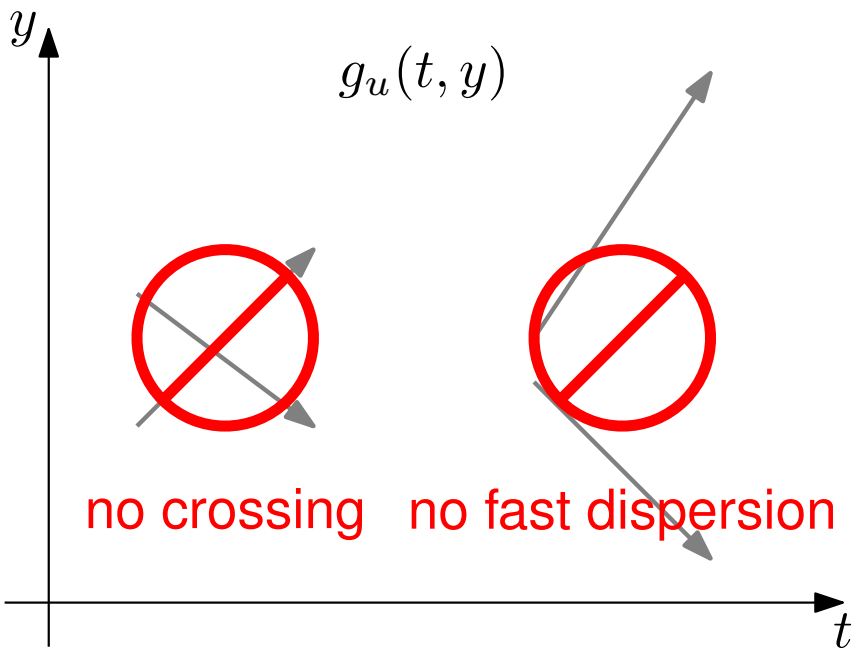
Not all G_u can be translated to g_u.



Flows can never cross.

Proof (2/3): Why this attempt does not work

Not all G_u can be translated to g_u.

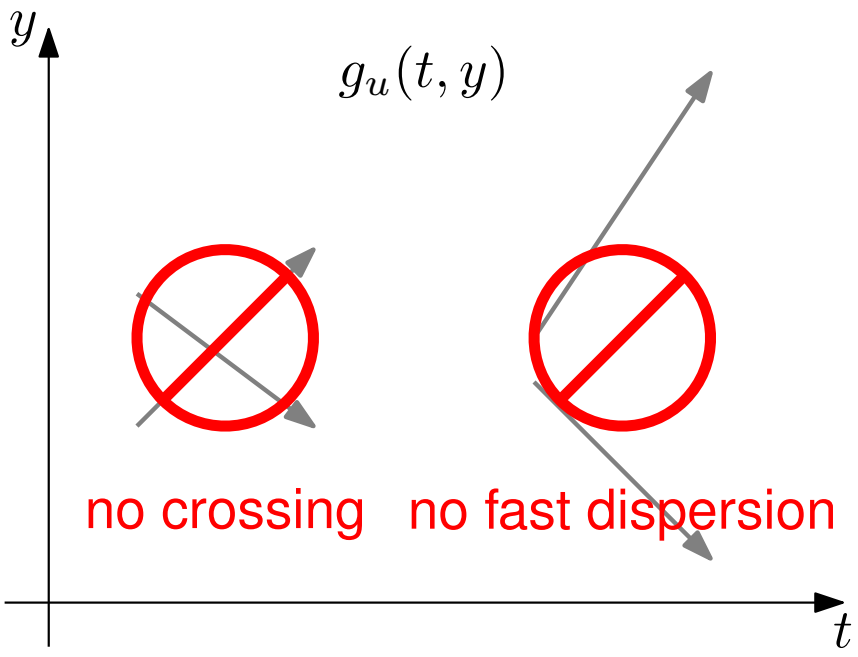


Flows can never cross.

In fact, the Lipschitz condition keeps them from widening or narrowing fast.

Proof (2/3): Why this attempt does not work

Not all G_u can be translated to g_u.

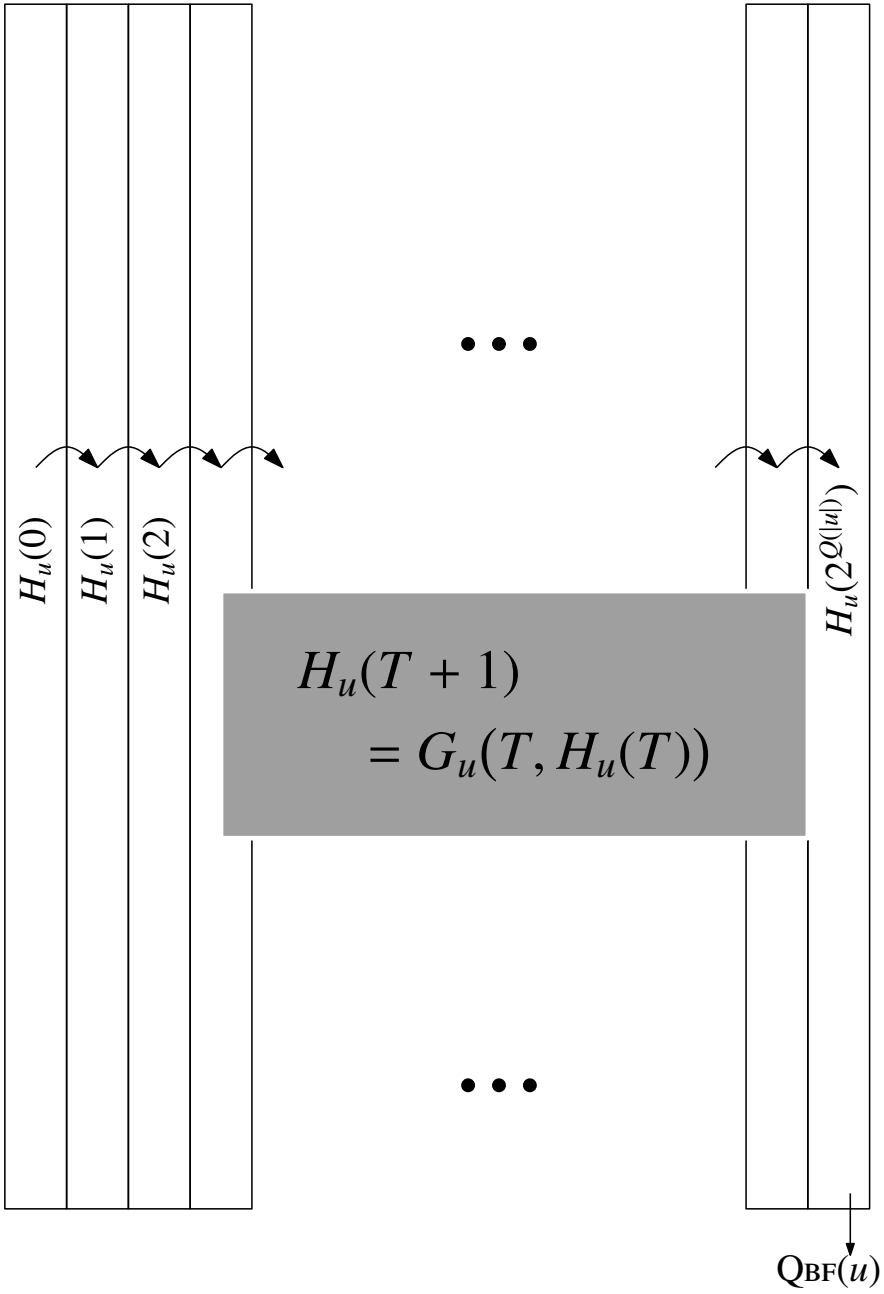


Flows can never cross.

In fact, the Lipschitz condition keeps them from widening or narrowing fast.

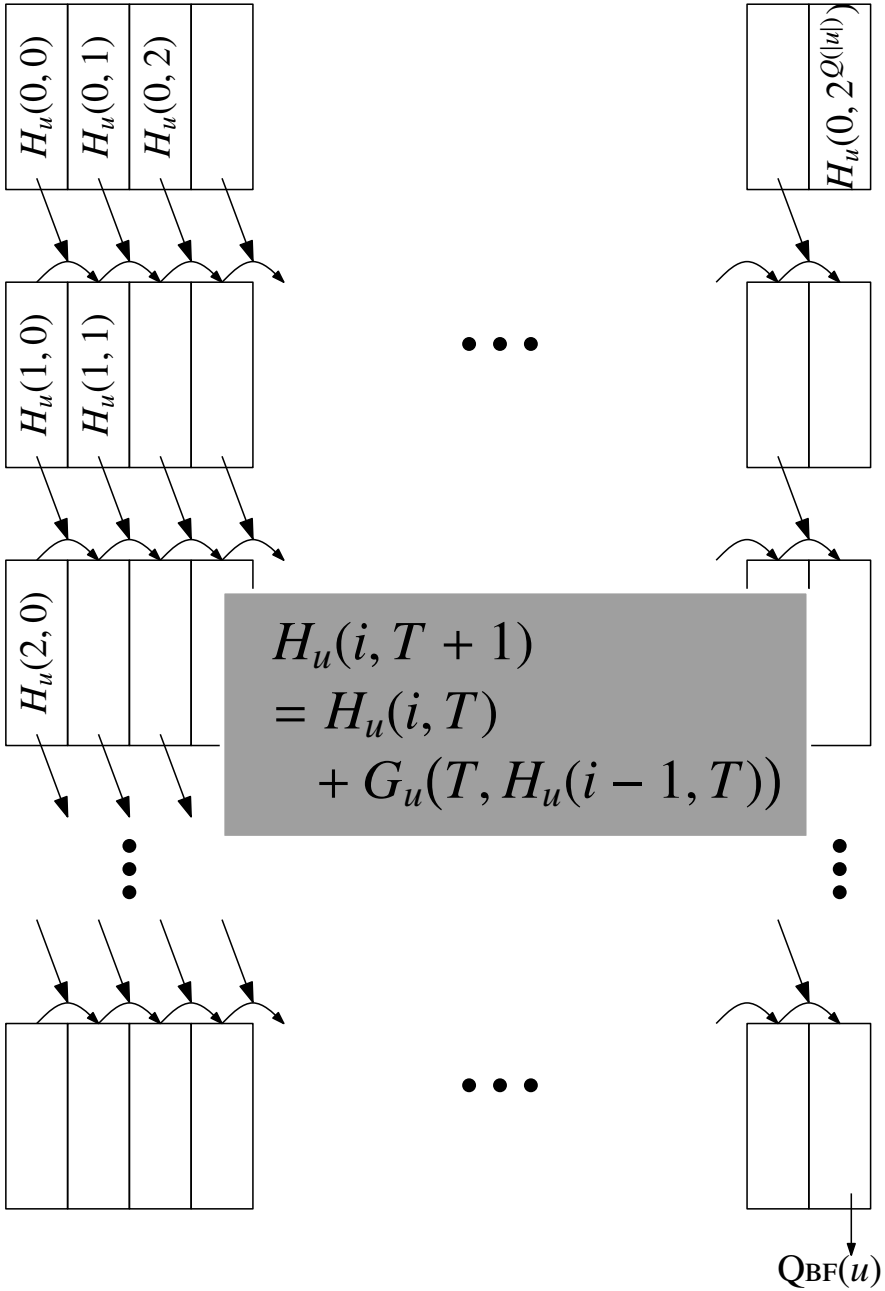
The 'feedback' (on h_u) of equation $h'_u(t) = g_u\big(t, h_u(t)\big)$ is very weak under the Lipschitz condition.

Proof (3/3): Layered PSPACE tableaux



General PSPACE computation tableaux cannot be simulated by the differential equation.

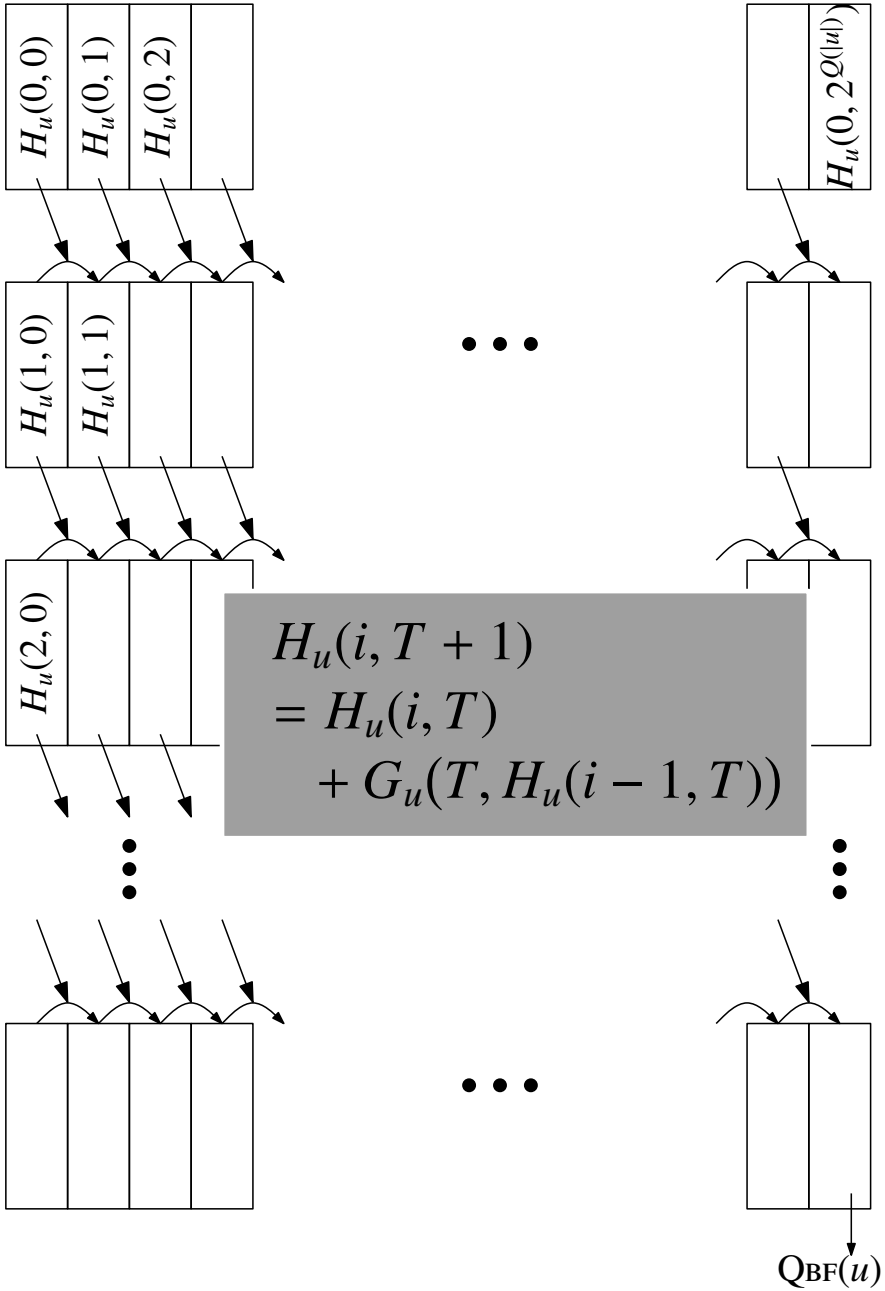
Proof (3/3): Layered PSPACE tableaux



General PSPACE computation tableaux cannot be simulated by the differential equation.

Split each cell into parts, and put restrictions on which part can affect which.

Proof (3/3): Layered PSPACE tableaux



General PSPACE computation tableaux cannot be simulated by the differential equation.

Split each cell into parts, and put restrictions on which part can affect which.

This tableau can be simulated by the equation, and is PSPACE-complete despite the restriction.

Outline

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big)$$

1. Complexity of real functions
   How we define PTIME real functions
2. Warm-up: Integration
   Assuming g is PTIME and ignores the second argument, how complex can h be?
3. Lipschitz continuous IVP
   Assuming g is PTIME and Lipschitz continuous, how complex can h be?
4. Final remarks

Related results

$$h(0) = 0, \qquad h'(t) = g\big(t, h(t)\big)$$

Assuming g is PTIME, how complex can h be?

Assumptions	Positive results on h	Negative results on h
None	——	can be (non-unique and) all non-computable [Pour-El 1979]
h is the unique solution	computable [implicit in Osgood 1898]	time (or space) cannot be bounded [Miller 1970]
g is Lipschitz continuous	PSPACE [Ko 1983]	can be PSPACE-hard
g is analytic	PTIME [Ko 1988, K.]	——

Open problems

- What happens between the Lipschitz case (PSPACE) and the analytic case (PTIME)?
- Effective versions? (Complexity of "computing h from g")
- Other differential equations