

Základy složitosti a vyčíslitelnosti

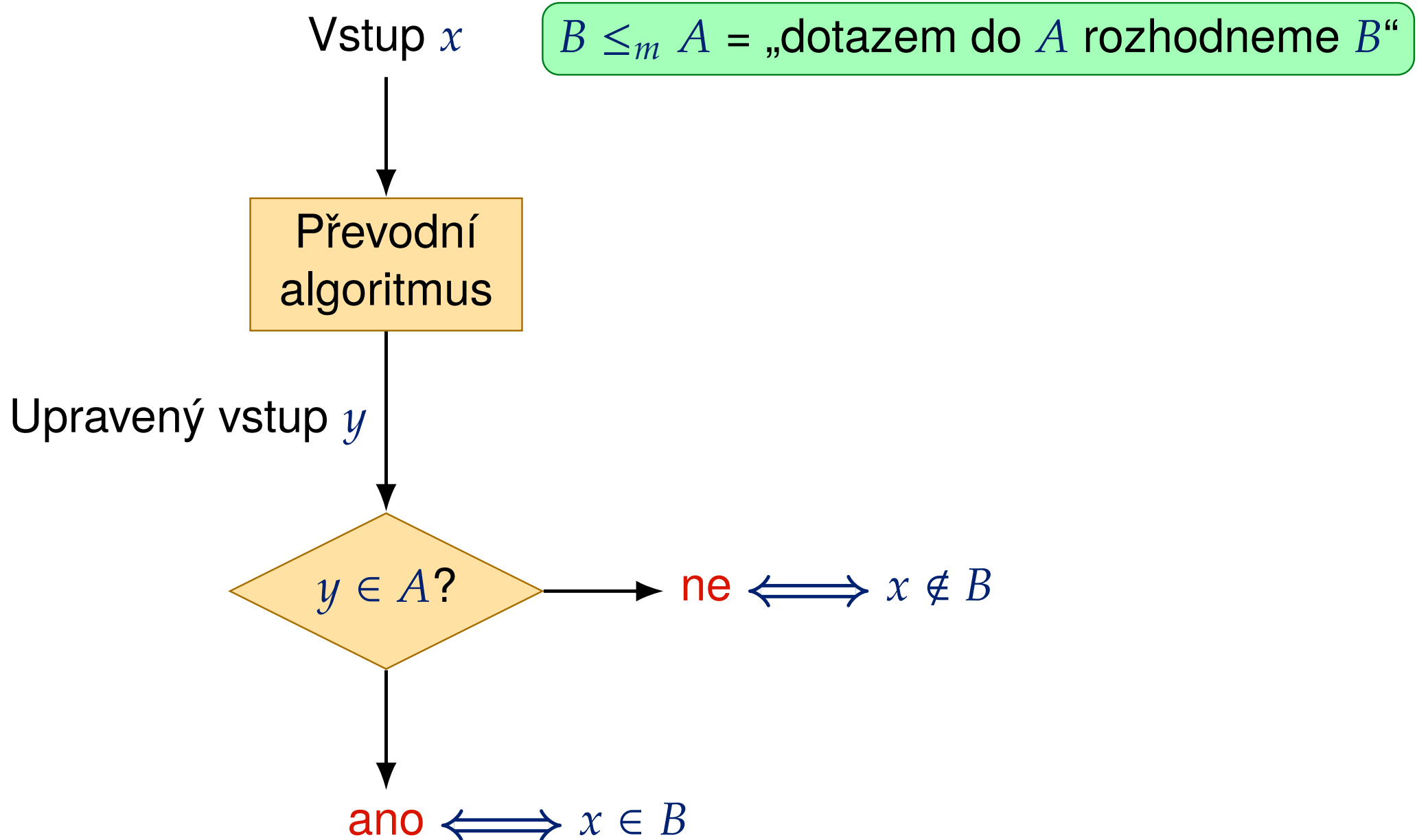
NTIN090

Petr Kučera

2024/25 (4. přednáška)

Převoditelnost (dokončení)

m -převoditelnost (princip)

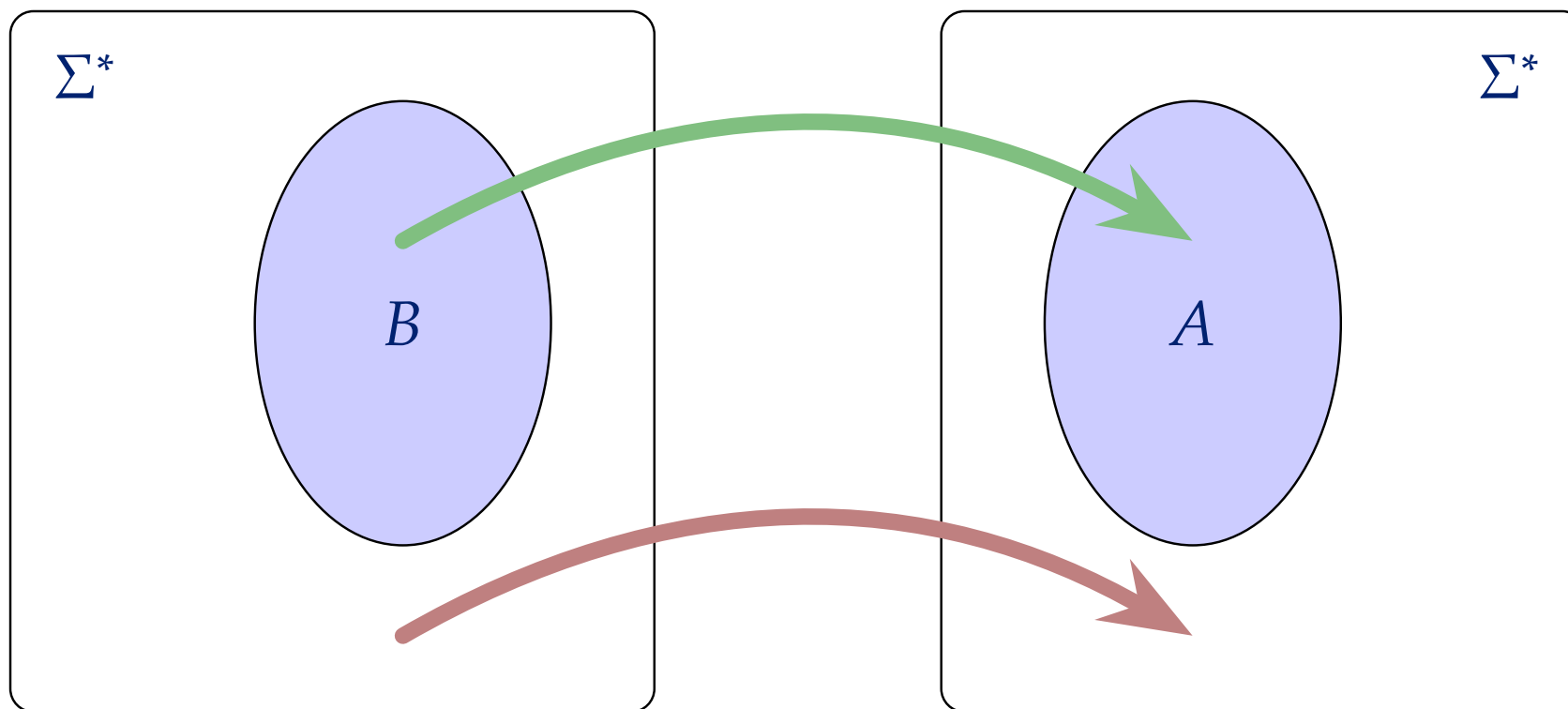


m -převoditelnost (definice)

Definice

Jazyk B je m -převoditelný na jazyk A , právě když existuje totální vyčíslitelná funkce f splňující

$$(\forall x \in \Sigma^*)[x \in B \Leftrightarrow f(x) \in A]$$



m -převoditelnost (definice)

Definice

Jazyk B je m -převoditelný na jazyk A ($B \leq_m A$), právě když existuje totální vyčíslitelná funkce f splňující

$$(\forall x \in \Sigma^*)[x \in B \Leftrightarrow f(x) \in A]$$

- $\leq_m$ je reflexivní a tranzitivní relace (**kvaziuspořádání**).
- Pokud $A \leq_m B$ a B je (částečně) rozhodnutelný jazyk, pak totéž lze říct o A .

$\overline{L_u}$	$\not\leq_m$	L_u
$\underbrace{\hspace{1.5cm}}$		$\underbrace{\hspace{1.5cm}}$
není částečně rozhodnutelný		částečně rozhodnutelný

Neprázdnost jazyka

PROBLÉM NEPRÁZDNÉHO JAZYKA

Instance: Kód Turingova stroje M

Otázka: Je $L(M) \neq \emptyset$?

$$NE = \{\langle M \rangle \mid L(M) \neq \emptyset\}$$

$\mathcal{L} \supset NonEmpty$

Věta

Jazyk NE je nerozhodnutelný.

- Nerozhodnutelnost ukážeme převodem L_u na NE

$$L_u \leq_m NE$$

$$L_u = \{\langle M, x \rangle \mid x \in L(M)\}$$

$$NE = \{\langle M \rangle \mid L(M) \neq \emptyset\}$$

K dvojici M a x definujeme Turingův stroj M'

→ samotný stroj od NE nepřijímá s nějakým vstupem

Výpočet M' se vstupem y

Puť $M(x)$ // M' ignoruje svůj vstup!

if M přijal **then**

přijmi

else

odmítni

tohle je na tom důležité

$$L(M') = \begin{cases} \Sigma^* & x \in L(M) \leadsto \text{nepřímý jazyk} \\ \emptyset & x \notin L(M) \leadsto \text{prázdný jazyk} \end{cases}$$

$$L_u \leq_m \text{NE}$$

- Pro každý kód dvojice $\langle M, x \rangle$ platí

$$\langle M, x \rangle \in L_u \iff x \in L(M) \iff L(M') \neq \emptyset \iff \langle M' \rangle \in \text{NE}$$

- Funkce $f(\langle M, x \rangle) = \langle M' \rangle$
 - je totální algoritmicky vyčíslitelná a
 - ukazuje, že $L_u \leq_m \text{NE}$

Riceova věta

Riceova věta

Věta (Riceova věta)

Nechť C je třída částečně rozhodnutelných jazyků a položme

$$L_C = \{\langle M \rangle \mid L(M) \in C\}$$

Jazyk L_C rozhodnutelný, právě když je třída C *triviální*, tedy buď je prázdná nebo obsahuje všechny částečně rozhodnutelné jazyky.

- Nechť P je netriviální vlastnost jazyka
- Následující otázka je nerozhodnutelná

Splňuje $L(M)$ vlastnost P pro daný program M ?

TEST TESTU PRVOČÍSELNOSTI

TEST TESTU PRVOČÍSELNOSTI

Instance: Zdrojový kód programu P .

Otázka: Přijímá P , právě když vstupní číslo je prvočíslo?

- Odpovídá mu jazyk

$$TTP = \{\langle P \rangle \mid L(P) = \text{PRIME}\},$$

kde

$$\text{PRIME} = \{\langle n \rangle \mid n \text{ je prvočíslo}\}$$

- Definujme $C = \{\text{PRIME}\}$
- Pak $TTP = L_C$

Podle Riceovy věty je jazyk TTP nerozhodnutelný

Neprázdnost jazyka

PROBLÉM NEPRÁZDNÉHO JAZYKA

Instance: Kód Turingova stroje M

Otázka: Je $L(M) \neq \emptyset$?

$$NE = \{\langle M \rangle \mid L(M) \neq \emptyset\}$$

- Definujme $C = \{L \mid L \text{ je částečně rozhodnutelný a } L \neq \emptyset\}$
- Pak $NE = L_C$

Podle Riceovy věty je jazyk NE nerozhodnutelný

Riceova věta (další důsledky)

Nerozhodnutelnost následujících jazyků plyne z Riceovy věty

- $\text{Fin} = \{\langle M \rangle \mid L(M) \text{ je konečný jazyk}\}$
 - C je třída konečných jazyků
- $\text{Inf} = \{\langle M \rangle \mid L(M) \text{ je nekonečný jazyk}\}$
 - C je třída nekonečných částečně rozhodnutelných jazyků
- $\text{Dec} = \{\langle M \rangle \mid L(M) \text{ je rozhodnutelný jazyk}\}$
 - C je třída rozhodnutelných jazyků
- $\text{Reg} = \{\langle M \rangle \mid L(M) \text{ je regulární jazyk}\}$
 - C je třída regulárních jazyků
- $\text{Hello} = \{\langle M \rangle \mid M \text{ přijímá řetězec Hello}\}$
 - $C = \{L \mid L \text{ je částečně rozhodnutelný a Hello} \in L\}$

Riceova věta (nepoužitelnost)

$\langle M_1 \rangle \in L(M_1), \langle M_2 \rangle \notin L(M_2)$ — tohle vlastnost *stihje*
 $L(M_1) = L(M_2)$ — $\Rightarrow$ tohle *je* vlastnost *jazyka*

Věta (Riceova věta)

Nechť C je třída částečně rozhodnutelných jazyků a položme

$$L_C = \{ \langle M \rangle \mid L(M) \in C \}$$

Jazyk L_C rozhodnutelný, právě když je třída C **triviální**, tedy buď je prázdná nebo obsahuje všechny částečně rozhodnutelné jazyky.

NELZE použít na jazyky

$$L_u = \{ \langle M, x \rangle \mid x \in L(M) \}$$

$$\text{HALT} = \{ \langle M, x \rangle \mid M(x) \downarrow \}$$

$$\text{DIAG} = \{ \langle M \rangle \mid \langle M \rangle \notin L(M) \}$$

Převod L_u na TTP

Důkaz Riceovy věty je zobecněním následujícího převodu

- $L_u = \{\langle M, x \rangle \mid x \in L(M)\}$
- $\text{TTP} = \{\langle P \rangle \mid L(P) = \text{PRIME}\}$

$$L_u \leq_m \text{TTP}$$

Idea:

- Definujeme $f(\langle M, x \rangle) = \langle P \rangle$ tak, aby platilo, že

$$x \in L(M) \iff L(P) = \text{PRIME}$$

- Algoritmus počítající f provede konstrukci P

$$L_u \leq_m \text{TTP}$$

$$L_u = \{\langle M, x \rangle \mid x \in L(M)\}$$

$$\text{TTP} = \{\langle P \rangle \mid L(P) = \text{PRIME}\}$$

K $\langle M, x \rangle$ definujeme program P takto

Výpočet P se vstupem n

Puť $M(x)$

if M přijal **and** n je prvočíslo **then**

 | **přijmi**

else

 | **odmítni**

- $M(x)$ přijme $\implies L(P) = \text{PRIME} \implies \langle P \rangle \in \text{TTP}$
- $M(x)$ nepřijme $\implies L(P) = \emptyset \implies \langle P \rangle \notin \text{TTP}$

Riceova věta (důkaz)

Věta (Riceova věta)

Nechť C je třída částečně rozhodnutelných jazyků a položme

$$L_C = \{ \langle M \rangle \mid L(M) \in C \}$$

*Jazyk L_C rozhodnutelný, právě když je třída C **triviální**, tedy buď je prázdná nebo obsahuje všechny částečně rozhodnutelné jazyky.*

- Nechť třída C je prázdná

$$\implies L_C = \emptyset$$

$$\implies L_C \text{ je rozhodnutelný}$$

Riceova věta (důkaz)

Věta (Riceova věta)

Nechť C je třída částečně rozhodnutelných jazyků a položme

$$L_C = \{\langle M \rangle \mid L(M) \in C\}$$

Jazyk L_C rozhodnutelný, právě když je třída C *triviální*, tedy buď je prázdná nebo obsahuje všechny částečně rozhodnutelné jazyky.

- Nechť třída C je prázdná
 - $\implies L_C = \emptyset$
 - $\implies L_C$ je rozhodnutelný
- Nechť třída C obsahuje všechny částečně rozhodnutelné jazyky
 - $\implies L_C = \Sigma^*$
 - $\implies L_C$ je rozhodnutelný

Riceova věta (důkaz)

- Dále předpokládáme, že C je netriviální
- Ukážeme, že
 - $L_u \leq_m \underline{L_C}$ pokud C neobsahuje prázdný jazyk
 - $L_u \leq_m \overline{L_C}$ pokud C obsahuje prázdný jazyk

Idea

Zobecnění převodu L_u na $\text{TTP} = \{\langle P \rangle \mid L(P) = \text{PRIME}\}$.

Riceova věta (důkaz)

- Předpokládejme, že C neobsahuje prázdný jazyk
 - tedy $\emptyset \notin C$

Riceova věta (důkaz)

- Předpokládejme, že C neobsahuje prázdný jazyk
 - tedy $\emptyset \notin C$
- Zvolíme Turingův stroj M_1 takový, že $L(M_1) \in C$
 - Speciálně $L(M_1) \neq \emptyset$

Riceova věta (důkaz)

- Předpokládejme, že C neobsahuje prázdný jazyk
 - tedy $\emptyset \notin C$
- Zvolíme Turingův stroj M_1 takový, že $L(M_1) \in C$
 - Speciálně $L(M_1) \neq \emptyset$
- Popíšeme totální vyčíslitelnou funkci $f(\langle M, x \rangle) = \langle M' \rangle$ splňující

$$L(M') = \begin{cases} L(M_1) & x \in L(M) \\ \emptyset & x \notin L(M) \end{cases}$$

Riceova věta (důkaz)

- Předpokládejme, že C neobsahuje prázdný jazyk
 - tedy $\emptyset \notin C$
- Zvolíme Turingův stroj M_1 takový, že $L(M_1) \in C$
 - Speciálně $L(M_1) \neq \emptyset$
- Popíšeme totální vyčíslitelnou funkci $f(\langle M, x \rangle) = \langle M' \rangle$ splňující

$$L(M') = \begin{cases} L(M_1) & x \in L(M) \\ \emptyset & x \notin L(M) \end{cases}$$

- Platí tedy

$$x \in L(M) \iff L(M') \in C \iff f(\langle M, x \rangle) = \langle M' \rangle \in L_C$$

Riceova věta (důkaz)

- Předpokládejme, že C neobsahuje prázdný jazyk
 - tedy $\emptyset \notin C$
- Zvolíme Turingův stroj M_1 takový, že $L(M_1) \in C$
 - Speciálně $L(M_1) \neq \emptyset$
- Popíšeme totální vyčíslitelnou funkci $f(\langle M, x \rangle) = \langle M' \rangle$ splňující

pro test testu prvočíselnosti:

$$L(M') = \begin{cases} L(M_1) & x \in L(M) \\ \emptyset & x \notin L(M) \end{cases}$$

PRIME : $x \in L(M)$

$\emptyset$: $x \notin L(M)$

$\hookrightarrow$ tímto simulujeme docyklení

- Platí tedy

$$x \in L(M) \iff L(M') \in C \iff f(\langle M, x \rangle) = \langle M' \rangle \in L_C$$

- Funkce f ukazuje, že $L_u \leq_m L_C$

Riceova věta (důkaz)

K dvojici M a x definujeme Turingův stroj M'

Výpočet M' se vstupem y

Simuluj $M(x)$

if M přijal **then** $// \quad x \in L(M)$

 Simuluj $M_1(y)$

if M_1 přijal **then**

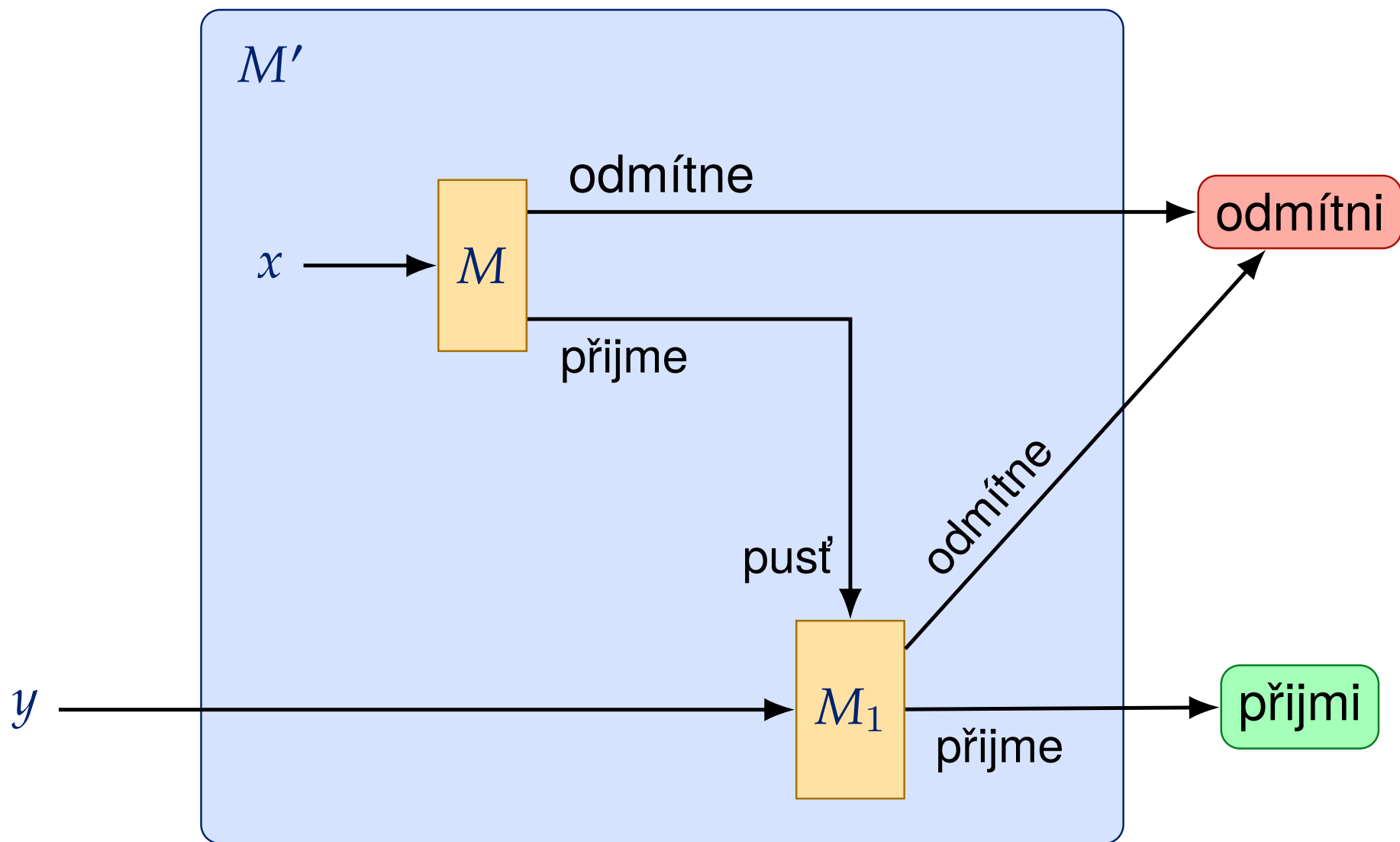
 přijmi

} pokud y je prvočíslo (u příkladu TTP)

odmítni

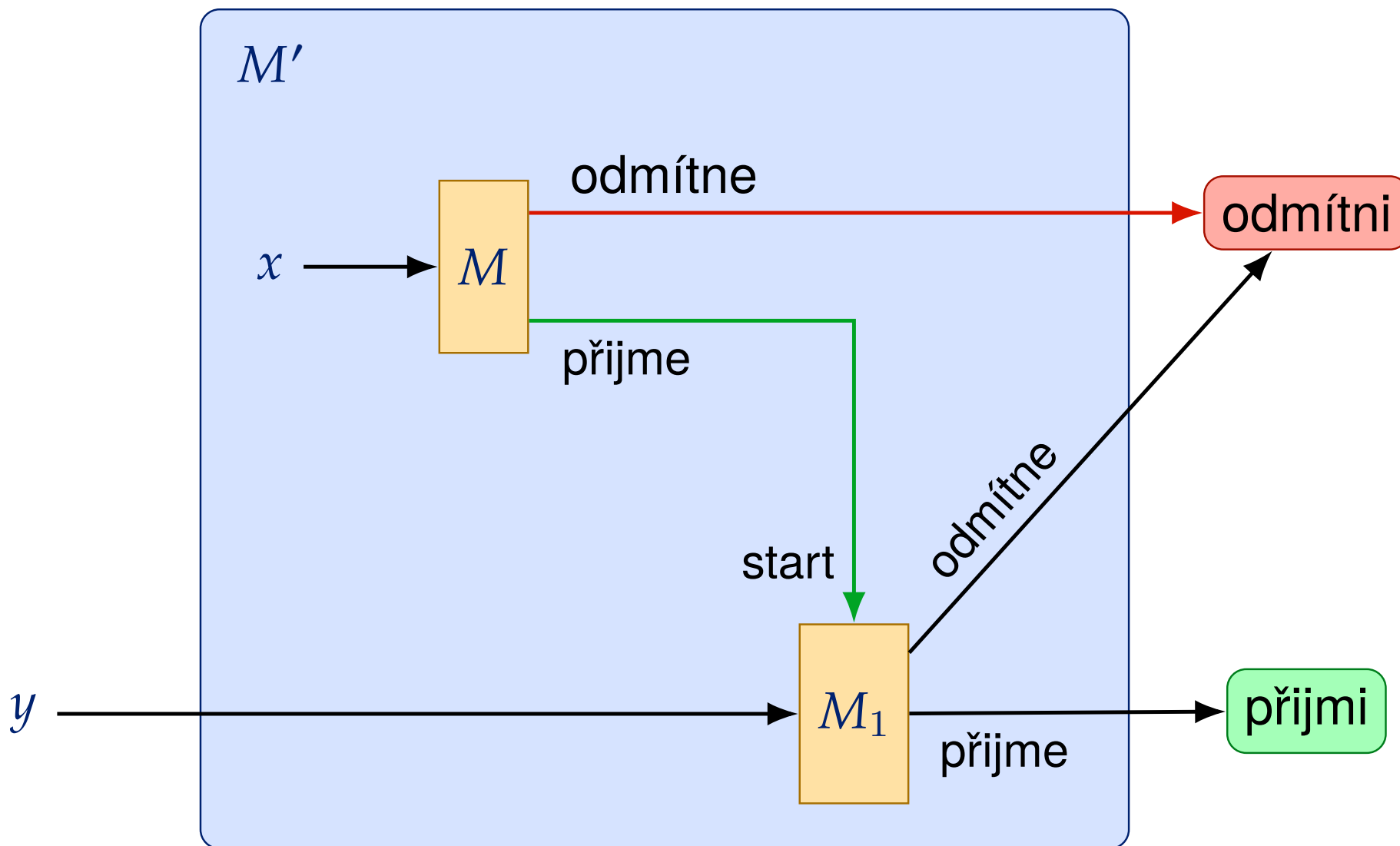
Riceova věta (důkaz)

K dvojici M a x definujeme Turingův stroj M'



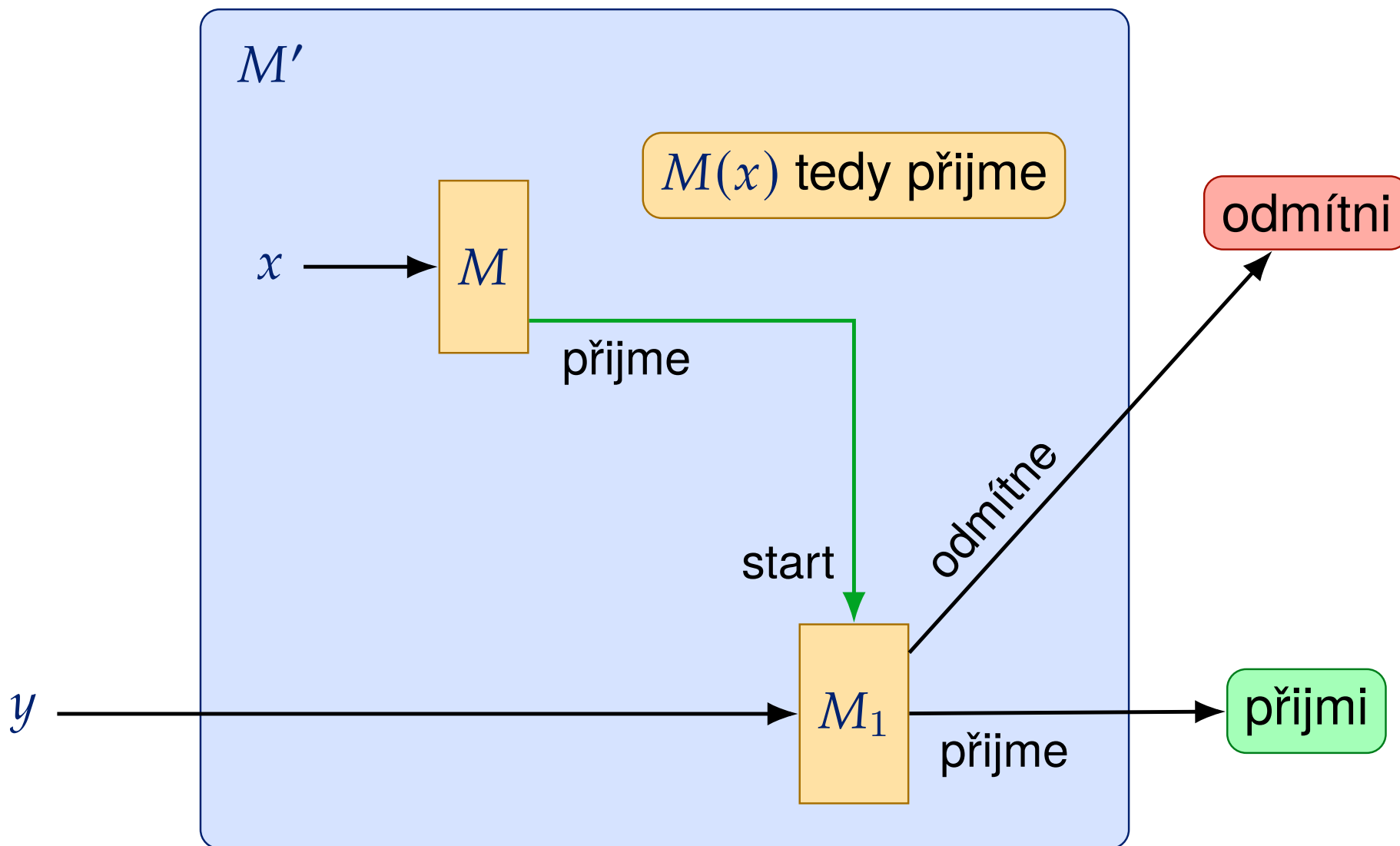
Riceova věta (důkaz, $x \in L(M)$)

Předpokládejme $x \in L(M)$



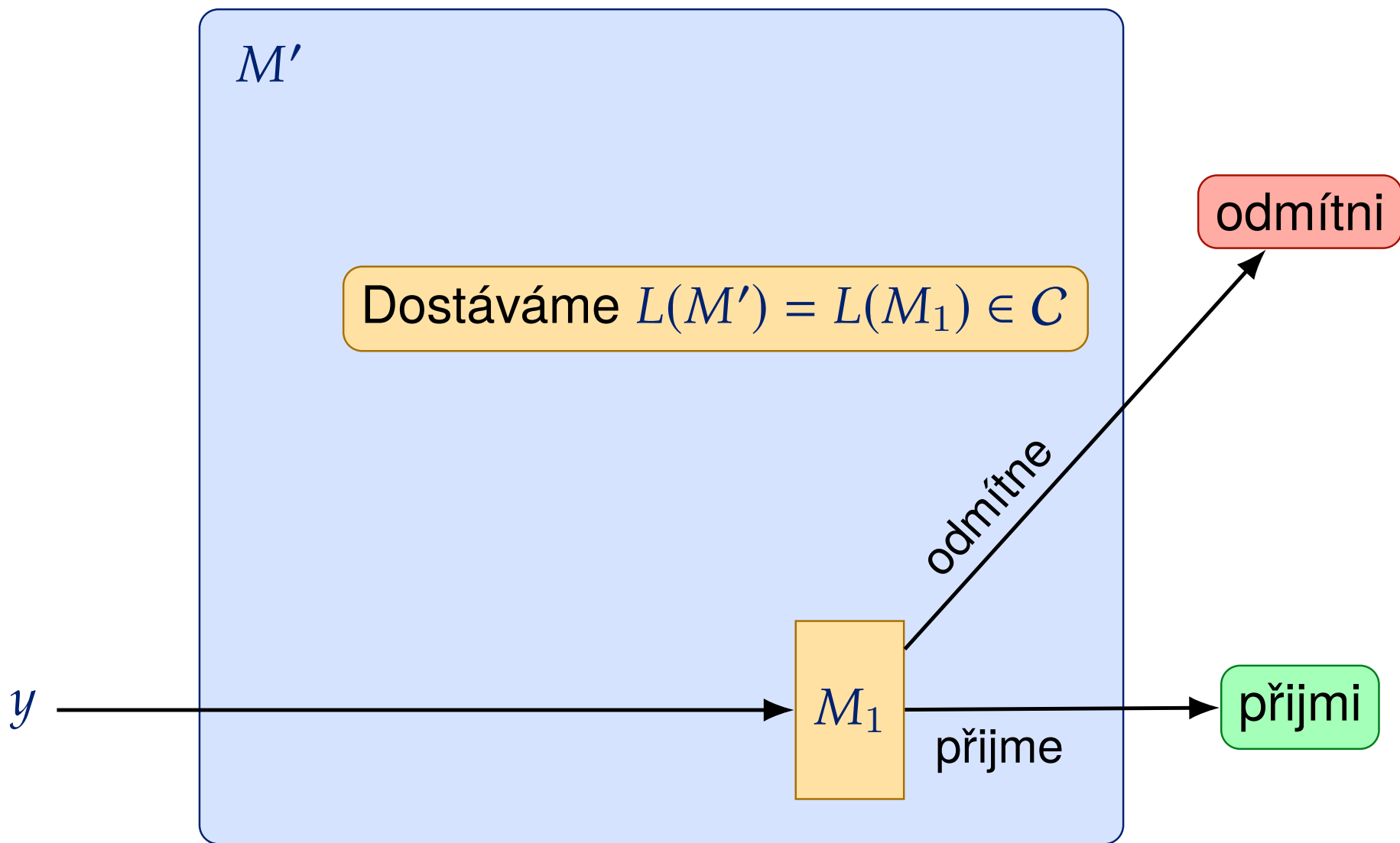
Riceova věta (důkaz, $x \in L(M)$)

Předpokládejme $x \in L(M)$



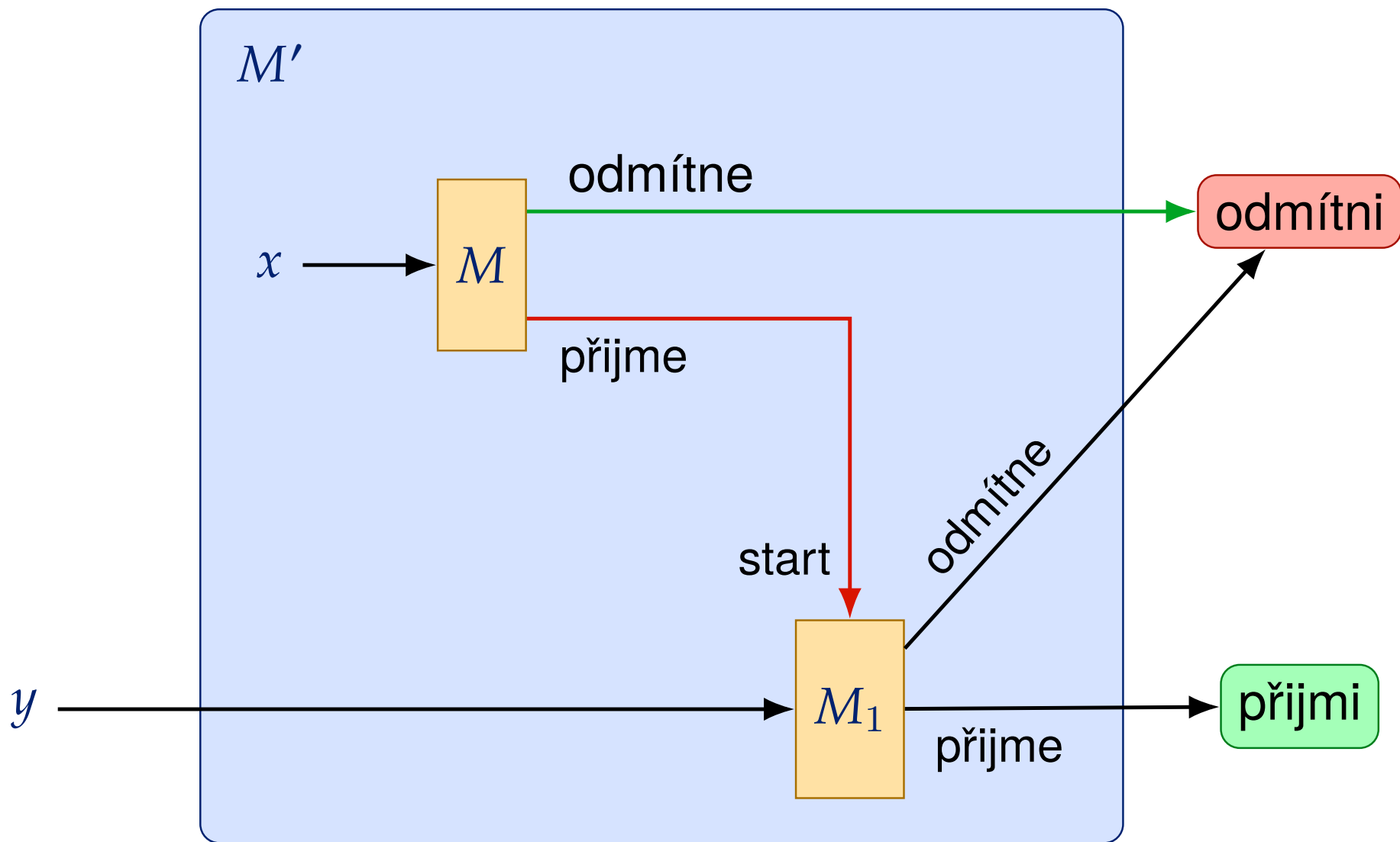
Riceova věta (důkaz, $x \in L(M)$)

Předpokládejme $x \in L(M)$



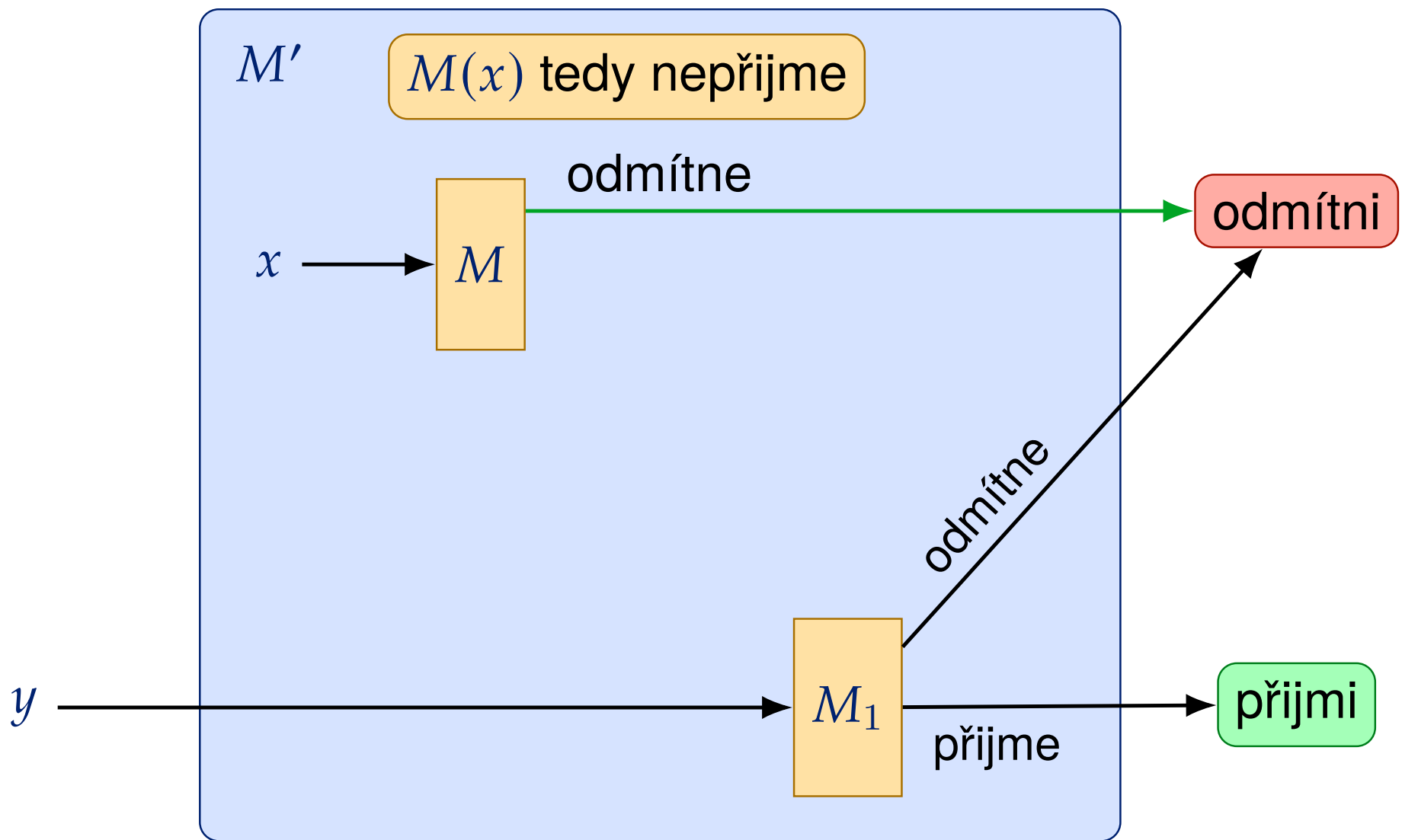
Riceova věta (důkaz, $x \notin L(M)$)

Předpokládejme $x \notin L(M)$



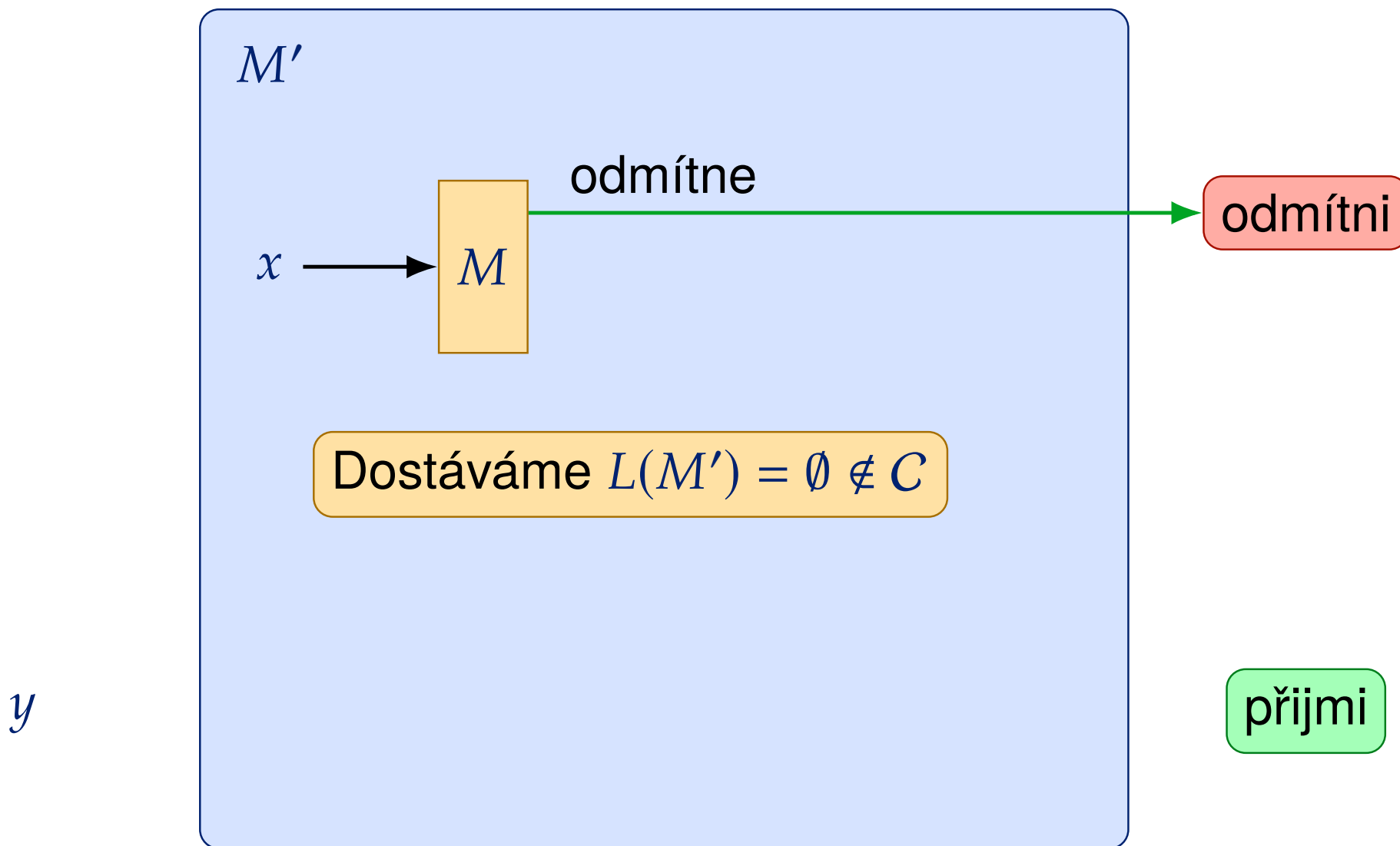
Riceova věta (důkaz, $x \notin L(M)$)

Předpokládejme $x \notin L(M)$



Riceova věta (důkaz, $x \notin L(M)$)

Předpokládejme $x \notin L(M)$



Riceova věta (důkaz, dokončení)

$$\blacksquare \quad x \in L(M) \implies L(M') = L(M_1) \in C \implies \langle M' \rangle \in L_C$$

Riceova věta (důkaz, dokončení)

- $x \in L(M) \implies L(M') = L(M_1) \in C \implies \langle M' \rangle \in L_C$
- $x \notin L(M) \implies L(M') = \emptyset \notin C \implies \langle M' \rangle \notin L_C$

Riceova věta (důkaz, dokončení)

- $x \in L(M) \implies L(M') = L(M_1) \in C \implies \langle M' \rangle \in L_C$
- $x \notin L(M) \implies L(M') = \emptyset \notin C \implies \langle M' \rangle \notin L_C$
- Funkce $f(\langle M, x \rangle) = \langle M' \rangle$ je totální algoritmicky vyčíslitelná a platí

$$\langle M, x \rangle \in L_u \iff x \in L(M) \iff f(\langle M, x \rangle) = \langle M' \rangle \in L_C$$

Riceova věta (důkaz, dokončení)

- $x \in L(M) \implies L(M') = L(M_1) \in C \implies \langle M' \rangle \in L_C$
- $x \notin L(M) \implies L(M') = \emptyset \notin C \implies \langle M' \rangle \notin L_C$
- Funkce $f(\langle M, x \rangle) = \langle M' \rangle$ je totální algoritmicky vyčíslitelná a platí

$$\langle M, x \rangle \in L_u \iff x \in L(M) \iff f(\langle M, x \rangle) = \langle M' \rangle \in L_C$$

- Dostáváme $L_u \leq_m L_C$

Riceova věta (důkaz, dokončení)

- $x \in L(M) \implies L(M') = L(M_1) \in C \implies \langle M' \rangle \in L_C$
- $x \notin L(M) \implies L(M') = \emptyset \notin C \implies \langle M' \rangle \notin L_C$
- Funkce $f(\langle M, x \rangle) = \langle M' \rangle$ je totální algoritmicky vyčíslitelná a platí

$$\langle M, x \rangle \in L_u \iff x \in L(M) \iff f(\langle M, x \rangle) = \langle M' \rangle \in L_C$$

- Dostáváme $L_u \leq_m L_C$
- Pokud $\emptyset \in C$
 - stačí vyměnit role C a doplňku C
 - tím ukážeme $L_u \leq_m \overline{L_C}$

Riceova věta (funkce)

Pomocí f_M označíme funkci, kterou počítá Turingův stroj M .

Věta (Riceova věta (funkce))

Nechť C je třída algoritmicky vyčíslitelných funkcí a položme

$$A_C = \{\langle M \rangle \mid f_M \in C\}$$

*Jazyk A_C rozhodnutelný, právě když je třída C **triviální**, tedy buď je prázdná nebo obsahuje všechny algoritmicky vyčíslitelné funkce.*

- Důkaz analogický verzi pro jazyky

Riceova věta (důsledky)

Následující jazyky jsou nerozhodnutelné dle Riceovy věty

- $\text{Tot} = \{\langle M \rangle \mid f_M \text{ je totální, tedy } \text{dom } f_M = \Sigma^*\}$
- $\text{Sum} = \{\langle M \rangle \mid M \text{ počítá součet dvou čísel}\}$
- $\text{Inc} = \{\langle M \rangle \mid f_M \text{ je rostoucí}\}$
- $\text{Zero} = \{\langle M \rangle \mid f_M(0) \downarrow = 0\}$
- $\text{ToUpper} = \{\langle M \rangle \mid M \text{ změní ve vstupu malá písmena na velká}\}$
- $\text{HW} = \{\langle M \rangle \mid M \text{ vypíše } \text{Hello, world} \text{ jako prvních } 12 \text{ znaků výstupu}\}$

Postův korespondenční problém

POSTŮV KORESPONDENČNÍ PROBLÉM

Instance: Množina „dominových kostek“ P :

$$P = \left\{ \left[\frac{t_1}{b_1} \right], \left[\frac{t_2}{b_2} \right], \dots, \left[\frac{t_k}{b_k} \right] \right\}$$

kde $t_1, \dots, t_k, b_1, \dots, b_k \in \Sigma^*$ jsou řetězce.

Otázka: Existuje **párovací posloupnost** $i_1, i_2, \dots, i_l$, kde $l \geq 1$
a $t_{i_1} t_{i_2} \dots t_{i_l} = b_{i_1} b_{i_2} \dots b_{i_l}$?

Věta (Bez důkazu)

POSTŮV KORESPONDENČNÍ PROBLÉM je nerozhodnutelný.

Postova věta

Uzavřenost na jazykové operace

Doplňěk jazyka L označíme pomocí $\bar{L} = \Sigma^* \setminus L$.

Konkatenací dvou jazyků L_1 a L_2 vznikne jazyk
$$L_1 \cdot L_2 = \{w_1w_2 \mid w_1 \in L_1, w_2 \in L_2\}.$$

Kleeneho uzávěrem jazyka L je jazyk

$$L^* = \{w \mid (\exists k \in \mathbb{N})(\exists w_1, \dots, w_k \in L)[w = w_1w_2 \dots w_k]\}.$$

Věta

Jsou-li L_1 a L_2 (částečně) rozhodnutelné jazyky, pak $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \cdot L_2$, L_1^ jsou (částečně) rozhodnutelné jazyky.*

Jak je to s doplňkem?

Postova věta

Věta (Postova věta)

Jazyk L je rozhodnutelný $\iff$ L i $\bar{L}$ jsou částečně rozhodnutelné jazyky.

→ pokud mi oba dostaní, pokud daný slovo přijímá.
Takže vždy musí na stejném vstupu alespoň jeden zastavit.

Postova věta

Věta (Postova věta)

Jazyk L je rozhodnutelný $\iff L$ i $\bar{L}$ jsou částečně rozhodnutelné jazyky.

Důkaz.

Dva kroky

„ $\implies$ “ L je rozhodnutelný $\implies L$ i $\bar{L}$ jsou částečně rozhodnutelné

„ $\impliedby$ “ L i $\bar{L}$ jsou částečně rozhodnutelné $\implies L$ je rozhodnutelný



Postova věta (důkaz „ $\Rightarrow$ “)

- Předpokládáme, že $L \subseteq \Sigma^*$ je rozhodnutelný jazyk

Postova věta (důkaz „ $\implies$ “)

- Předpokládáme, že $L \subseteq \Sigma^*$ je rozhodnutelný jazyk
- $\implies$ Existuje Turingův stroj M rozhodující L
- $L = L(M)$ a $M(x) \downarrow$ pro každý vstup $x \in \Sigma^*$

Postova věta (důkaz „ $\implies$ “)

- Předpokládáme, že $L \subseteq \Sigma^*$ je rozhodnutelný jazyk
- $\implies$ Existuje Turingův stroj M rozhodující L
 - $L = L(M)$ a $M(x) \downarrow$ pro každý vstup $x \in \Sigma^*$
- Následující Turingův stroj M' rozhoduje $\bar{L}$

Výpočet M' se vstupem x

```
1 Pust'  $M(x)$ 
2 if  $M(x)$  přijal then
3   | odmítni
4 else
5   | přijmi
```

Postova věta (důkaz „ $\implies$ “)

- Předpokládáme, že $L \subseteq \Sigma^*$ je rozhodnutelný jazyk
- $\implies$ Existuje Turingův stroj M rozhodující L
 - $L = L(M)$ a $M(x) \downarrow$ pro každý vstup $x \in \Sigma^*$
- Následující Turingův stroj M' rozhoduje $\bar{L}$

Výpočet M' se vstupem x

```
1 Pust'  $M(x)$ 
2 if  $M(x)$  přijal then
3   | odmítni
4 else
5   | přijmi
```

$\implies \bar{L}$ je rozhodnutelný jazyk

Postova věta (důkaz „ $\implies$ “)

- Předpokládáme, že $L \subseteq \Sigma^*$ je rozhodnutelný jazyk
- $\implies$ Existuje Turingův stroj M rozhodující L
 - $L = L(M)$ a $M(x) \downarrow$ pro každý vstup $x \in \Sigma^*$
- Následující Turingův stroj M' rozhoduje $\bar{L}$

Výpočet M' se vstupem x

```
1 Pust'  $M(x)$ 
2 if  $M(x)$  přijal then
3   | odmítni
4 else
5   | přijmi
```

$\implies \bar{L}$ je rozhodnutelný jazyk

$\implies L$ i $\bar{L}$ jsou částečně rozhodnutelné jazyky

Postova věta (důkaz „ $\Leftarrow$ “)

- Předpokládáme, že
 - $L = L(M_1)$ pro nějaký Turingův stroj $M_1 = (Q_1, \Sigma_1, \delta_1, q_0^1, F_1)$
 - $\bar{L} = L(M_2)$ pro nějaký Turingův stroj $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$

Postova věta (důkaz „ $\Leftarrow$ “)

- Předpokládáme, že
 - $L = L(M_1)$ pro nějaký Turingův stroj $M_1 = (Q_1, \Sigma_1, \delta_1, q_0^1, F_1)$
 - $\bar{L} = L(M_2)$ pro nějaký Turingův stroj $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$
- Každý řetězec $x \in \Sigma^*$ je přijat právě jedním ze strojů M_1 a M_2

Postova věta (důkaz „ $\Leftarrow$ “)

- Předpokládáme, že
 - $L = L(M_1)$ pro nějaký Turingův stroj $M_1 = (Q_1, \Sigma_1, \delta_1, q_0^1, F_1)$
 - $\bar{L} = L(M_2)$ pro nějaký Turingův stroj $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$
- Každý řetězec $x \in \Sigma^*$ je přijat právě jedním ze strojů M_1 a M_2
- Následující Turingův stroj M rozhoduje L

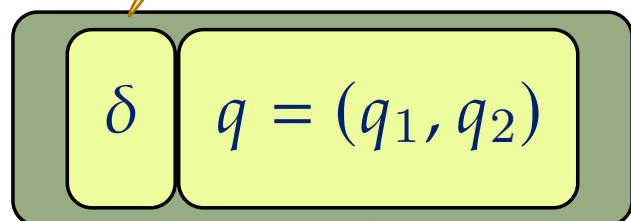
Výpočet M se vstupem x

- 1 Pusť výpočty $M_1(x)$ a $M_2(x)$ paralelně
 - 2 **if** $M_1(x)$ přijal **then přijmi**
 - 3 **if** $M_2(x)$ přijal **then odmítni**
-

vždy alespoň jeden zastaví

Možná implementace M

Přechodová funkce M
složená z δ_1 a δ_2



Stav M reprezentuje
stav q_1 stroje M_1 a
stav q_2 stroje M_2

Pozice hlavy stroje M_1



Páska stroje M_1

Pozice hlavy stroje M_2



Páska stroje M_2

Uzavřenost na doplněk

Důsledek

- *Třída rozhodnutelných jazyků je uzavřená na operaci doplňku*
- *Třída částečně rozhodnutelných jazyků není uzavřená na operaci doplňku*

Uzavřenost na doplněk

Důsledek

- *Třída rozhodnutelných jazyků je uzavřená na operaci doplňku*
- *Třída částečně rozhodnutelných jazyků není uzavřená na operaci doplňku*
- Jazyk L_u je částečně rozhodnutelný, ale není rozhodnutelný

Uzavřenost na doplněk

Důsledek

- *Třída rozhodnutelných jazyků je uzavřená na operaci doplňku*
- *Třída částečně rozhodnutelných jazyků není uzavřená na operaci doplňku*
- Jazyk L_u je částečně rozhodnutelný, ale není rozhodnutelný
- $\overline{L_u}$ není částečně rozhodnutelný dle Postovy věty

Uzavřenost na doplněk

Důsledek

- *Třída rozhodnutelných jazyků je uzavřená na operaci doplňku*
- *Třída částečně rozhodnutelných jazyků není uzavřená na operaci doplňku*
- Jazyk L_u je částečně rozhodnutelný, ale není rozhodnutelný
- $\overline{L_u}$ není částečně rozhodnutelný dle Postovy věty
- $\text{DIAG} = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\}$ není částečně rozhodnutelný

Uzavřenost na doplněk

Důsledek

- *Třída rozhodnutelných jazyků je uzavřená na operaci doplněku*
- *Třída částečně rozhodnutelných jazyků není uzavřená na operaci doplněku*
- Jazyk L_u je částečně rozhodnutelný, ale není rozhodnutelný
- $\overline{L_u}$ není částečně rozhodnutelný dle Postovy věty
- $\text{DIAG} = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\}$ není částečně rozhodnutelný
- $\overline{\text{DIAG}} = \{\langle M \rangle \mid \langle M \rangle \in L(M)\}$ je částečně rozhodnutelný
 - Plyne z existence univerzálního Turingova stroje

Vztahy tříd jazyků

PD částečně rozhodnutelné jazyky

- *partially decidable*

co-PD doplňky částečně
rozhodnutelných jazyků

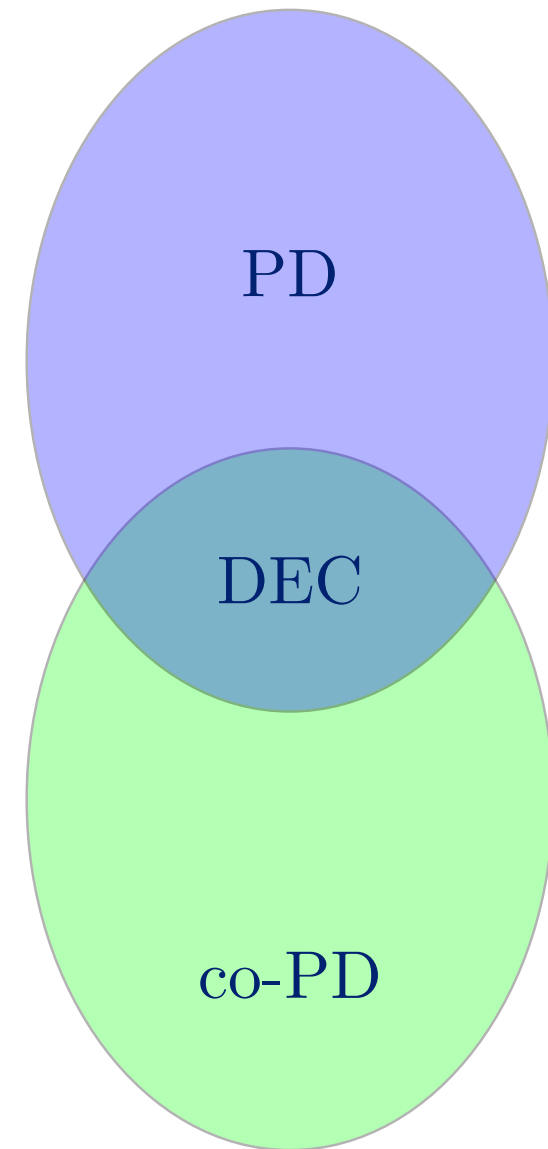
- $L \in \text{co-PD} \Leftrightarrow \bar{L} \in \text{PD}$
- *co-partially decidable*

DEC rozhodnutelné jazyky

- *decidable*

Postova věta: $\text{DEC} = \text{PD} \cap \text{co-PD}$

Všechny jazyky



Algoritmicky vyčíslitelné funkce

Funkce — značení

Pro částečnou funkci $f : \Sigma^* \rightarrow \Sigma^*$ definujeme:

Doména f je množina vstupů, pro něž je hodnota f definovaná

$$\text{dom } f = \{x \in \Sigma^* \mid f(x) \downarrow\}$$

Totální funkce f je definovaná pro každý vstup x , tedy $\text{dom } f = \Sigma^*$

Obor hodnot f je množina možných hodnot f

$$\text{rng } f = \{y \in \Sigma^* \mid (\exists x \in \Sigma^*)[f(x) \downarrow = y]\}$$

Značení používáme i pro jiné než řetězcové funkce

- například funkce $f : \mathbb{N} \rightarrow \mathbb{N}$

Algoritmicky vyčíslitelné funkce (definice)

Intuitivně

Algoritmicky vyčíslitelná funkce jsou právě ty, jejichž hodnoty lze vyčíslit nějakým algoritmem

Definice

Částečná funkce $f : \Sigma^* \rightarrow \Sigma^*$ je **algoritmicky vyčíslitelná** pokud existuje Turingův stroj M , který ji počítá.

Pro každý vstup $x \in \Sigma^*$ platí

- Je-li $f(x) \uparrow$, pak $M(x) \uparrow$
- Je-li $f(x) \downarrow = y$, pak
 - $M(x) \downarrow$ a
 - na výstupní pásce M je po ukončení výpočtu $M(x)$ řetězec y

Algoritmicky vyčíslitelné funkce

- Vyčíslitelné funkce = **částečně rekurzivní funkce**
- Totální vyčíslitelné funkce = **obecně rekurzivní funkce**
- Uvažujeme i funkce jiných typů, například
 - aritmetické funkce
 - funkce více parametrů

Příklad

Například funkce

$$f(x, y) = x^2 + y^2$$

může být realizována řetězcovou funkcí

$$f'(\langle x, y \rangle) = \langle x^2 + y^2 \rangle$$

Ne všechny funkce jsou vyčíslitelné

- Vyčíslitelných funkcí je jen spočetně mnoho
⇒ ne všechny funkce jsou vyčíslitelné

Příklad

Charakteristická funkce jazyka L_u

$$\chi_u(\langle M, x \rangle) = \begin{cases} 1 & x \in L(M) \\ 0 & x \notin L(M) \end{cases}$$

není algoritmicky vyčíslitelná, protože jazyk

$$L_u = \{ \langle M, x \rangle \mid x \in L(M) \}$$

je algoritmicky nerozhodnutelný

Vlastnosti (částečně) rozhodnutelných jazyků

Charakteristická funkce rozhodnutelného jazyka

Věta

Jazyk $L \subseteq \Sigma^*$ je rozhodnutelný, právě když jeho *charakteristická funkce*

$$\chi_L(x) = \begin{cases} 1 & x \in L \\ 0 & x \notin L \end{cases}$$

je algoritmicky vyčíslitelná.

Charakteristická funkce rozhodnutelného jazyka

Věta

Jazyk $L \subseteq \Sigma^*$ je rozhodnutelný, právě když jeho *charakteristická funkce*

$$\chi_L(x) = \begin{cases} 1 & x \in L \\ 0 & x \notin L \end{cases}$$

je algoritmicky vyčíslitelná.

Důkaz.

Důkaz ve dvou krocích

„ $\Rightarrow$ “ L je rozhodnutelný $\Rightarrow \chi_L$ je algoritmicky vyčíslitelná

„ $\Leftarrow$ “ χ_L je algoritmicky vyčíslitelná $\Rightarrow L$ je rozhodnutelný



Důkaz „ $\Rightarrow$ “

- Předpokládáme, že L je rozhodnutelný jazyk
- Existuje Turingův stroj M , který
 - přijímá L ($L = L(M)$)
 - $M(x) \downarrow$ pro každý vstup $x \in \Sigma^*$
- Následující Turingův stroj M' počítá χ_L

Výpočet M' se vstupem x

- 1 Simuluj $M(x)$
 - 2 **if** M přijal **then**
 - 3 | Zapiš na výstup 1
 - 4 **else**
 - 5 | Zapiš na výstup 0
-

Důkaz „ $\Leftarrow$ “

- Předpokládáme, že funkce χ_L je algoritmicky vyčíslitelná
- Existuje Turingův stroj M , který počítá χ_L
- $M(x) \downarrow$ pro každý vstup $x \in \Sigma^*$
 - protože $\chi_L(x) \downarrow$ pro každý vstup $x \in \Sigma^*$
- $M(x)$ vypíše na výstup hodnotu $\chi_L(x)$ (1 pokud $x \in L$, jinak 0)
- Následující Turingův stroj M' rozhoduje L

Výpočet M' se vstupem x

```
1 Simuluj  $M(x)$ 
2 if  $M$  vypsál 1 then
3   | přijmi
4 else
5   | odmítni
```

Přijetí nebo zastavení

Věta

Jazyk L je částečně rozhodnutelný, právě když existuje Turingův stroj M splňující

$$L = \{x \in \Sigma^* \mid M(x) \downarrow\} \quad (1)$$

Přijetí nebo zastavení

Věta

Jazyk L je částečně rozhodnutelný, právě když existuje Turingův stroj M splňující

$$L = \{x \in \Sigma^* \mid M(x) \downarrow\} \quad (1)$$

Důkaz.

Ve dvou krocích

„ $\Rightarrow$ “ L je částečně rozhodnutelný $\Rightarrow$ existuje M splňující (1)

„ $\Leftarrow$ “ Existuje M splňující (1) $\Rightarrow L$ je částečně rozhodnutelný



Důkaz „ $\implies$ “

- Předpokládáme, že L je částečně rozhodnutelný
- Existuje Turingův stroj M' , který přijímá L ($L = L(M')$)
- Popíšeme Turingův stroj M , který splňuje

$$L = \{x \in \Sigma^* \mid M(x) \downarrow\}$$

Výpočet M se vstupem x

- 1 Simuluj $M'(x)$
 - 2 **if** $M'(x)$ odmítl **then**
 - 3 └ vstup do nekonečného cyklu
-

- Pro každý řetězec $x \in \Sigma^*$

$$x \in L \iff M'(x) \text{ přijme} \iff M(x) \downarrow$$

Důkaz „ $\Leftarrow$ “

- Předpokládejme, že M je Turingův stroj splňující

$$L = \{x \in \Sigma^* \mid M(x) \downarrow\}$$

- Popíšeme Turingův stroj M' , který přijímá L ($L = L(M')$)

Výpočet M' se vstupem x

- 1 Simuluj $M(x)$
 - 2 **přijmi**
-

- Platí

$$x \in L \iff M(x) \downarrow \iff x \in L(M')$$

- Tedy $L = L(M')$

Domény algoritmicky vyčíslitelných funkcí

Věta

Jazyk L je částečně rozhodnutelný, právě když existuje algoritmicky vyčíslitelná funkce f splňující

$$L = \text{dom } f = \{x \in \Sigma^* \mid f(x) \downarrow\} \quad (2)$$

Domény algoritmicky vyčíslitelných funkcí

Věta

Jazyk L je částečně rozhodnutelný, právě když existuje algoritmicky vyčíslitelná funkce f splňující

$$L = \text{dom } f = \{x \in \Sigma^* \mid f(x) \downarrow\} \quad (2)$$

Důkaz.

- L je částečně rozhodnutelný, právě když existuje TS M splňující

$$L = \{x \in \Sigma^* \mid M(x) \downarrow\} \quad (3)$$

(2) $\implies$ (3) M počítá funkci f

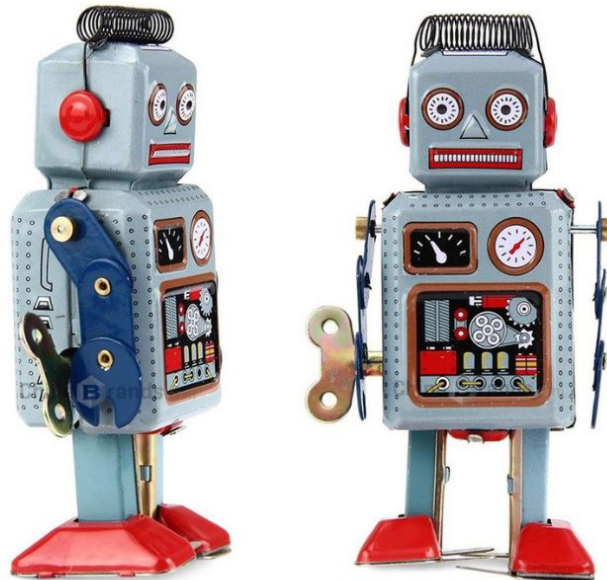
(3) $\implies$ (2) f je funkce počítaná strojem M



Soutěž pro středně pokročilé (připomenutí)

Soutěž

Pošlete nám program, který testuje, zda číslo na vstupu je prvočíslo a my vám pošleme robota!



Nekorektní programy

Kdy program P nerozhoduje prvočíselnost?

Možné protipříklady:

Typ 1 P o nějakém složeném čísle n prohlásí, že jde o prvočíslo

Typ 2 P o nějakém prvočísle n prohlásí, že nejde o prvočíslo

Typ 3 P o nějakém čísle n neprohlásí nic (výpočet se nezastaví)

Náš cíl

Chceme najít protipříklad pro P , pokud existuje.

Typ 3

Typ 3 P o nějakém čísle n neprohlásí nic (výpočet se nezastaví)

- Pro dané n nelze určit, zda se P se vstupem n zastaví — problém zastavení

Protipříklady tohoto typu nelze efektivně hledat.

Typ 1

Typ 1 P o nějakém složeném čísle n prohlásí, že jde o prvočíslo

PROTIPŘÍKLAD TYPU 1

Instance: Zdrojový kód programu P .

Otázka: Existuje složené číslo n , které P přijme?

- Odpovídá jazyku

$$T1 = \{ \langle P \rangle \mid (\exists n \in \mathbb{N})[n \text{ je složené číslo a } P(n) \text{ přijme}] \}$$

Je jazyk $T1$ částečně rozhodnutelný?

Existence protipříkladu typu 1

$$T1 = \{\langle P \rangle \mid (\exists n \in \mathbb{N})[n \text{ je složené číslo a } P(n) \text{ přijme}]\}$$

Je jazyk $T1$ částečně rozhodnutelný?

- Chceme algoritmus $\mathcal{A}$, který
 - hledá složené číslo n , které by P přijal,
 - pokud takové číslo neexistuje, algoritmus $\mathcal{A}$ nikdy neukončí činnost

Algoritmus $\mathcal{A}$

Výpočet $\mathcal{A}$ se vstupem P

$t \leftarrow 2$

while true do

for $a \leftarrow 2$ **to** t **do**

for $b \leftarrow 2$ **to** t **do**

$n \leftarrow a \cdot b$

 Simuluj $P(n)$ po nejvýš t kroků

if P přijal **then**

 └ **přijmi**

$t \leftarrow t + 1$

$L(\mathcal{A}) = \text{T1}$, jazyk T1 je tedy částečně rozhodnutelný.

Existenční kvantifikace

Věta

Jazyk L je částečně rozhodnutelný, právě když existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\} \quad (4)$$

$\hookrightarrow$ nemusím přijmout všechny, jen pro ty, které platí, zastavím.

Existenční kvantifikace

Věta

Jazyk L je částečně rozhodnutelný, právě když existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\} \quad (4)$$

Důkaz.

Důkaz ve dvou krocích

„ $\Rightarrow$ “ L je částečně rozhodnutelný $\Rightarrow$ existuje rozhodnutelný jazyk B splňující (4)

„ $\Leftarrow$ “ existuje rozhodnutelný jazyk B splňující (4) $\Rightarrow L$ je částečně rozhodnutelný



Důkaz „ $\Rightarrow$ “

- Předpokládáme, že L je částečně rozhodnutelný

Důkaz „ $\Rightarrow$ “

- Předpokládáme, že L je částečně rozhodnutelný
- Existuje Turingův stroj M přijímající L ($L = L(M)$)

Důkaz „ $\Rightarrow$ “

- Předpokládáme, že L je částečně rozhodnutelný
- Existuje Turingův stroj M přijímající L ($L = L(M)$)
- Platí

$$L = \{x \mid (\exists n \in \mathbb{N})[M(x) \text{ přijme do } n \text{ kroků}]\}$$

Rozhodnutelná podmínka,
stačí simulovat $M(x)$ po n kroků

Důkaz „ $\Rightarrow$ “

- Předpokládáme, že L je částečně rozhodnutelný
- Existuje Turingův stroj M přijímající L ($L = L(M)$)
- Platí

$$L = \{x \mid (\exists n \in \mathbb{N})[M(x) \text{ přijme do } n \text{ kroků}]\}$$

Rozhodnutelná podmínka,
stačí simulovat $M(x)$ po n kroků

- Stačí tedy definovat

$$B = \{\langle x, \langle n \rangle \rangle \mid M(x) \text{ přijme do } n \text{ kroků}\}$$

Důkaz „ $\Rightarrow$ “

- Předpokládáme, že L je částečně rozhodnutelný
- Existuje Turingův stroj M přijímající L ($L = L(M)$)
- Platí

$$L = \{x \mid (\exists n \in \mathbb{N})[M(x) \text{ přijme do } n \text{ kroků}]\}$$

Rozhodnutelná podmínka,
stačí simulovat $M(x)$ po n kroků

- Stačí tedy definovat

$$B = \{\langle x, \langle n \rangle \rangle \mid M(x) \text{ přijme do } n \text{ kroků}\}$$

- Jazyk B je rozhodnutelný a splňuje

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\underbrace{y = \langle n \rangle}_{y = \langle n \rangle} \quad \underbrace{\langle x, y \rangle \in B}_{M(x) \text{ přijme do } n \text{ kroků}}] \}$$

Důkaz „ $\Leftarrow$ “

- Předpokládáme, že existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

- Popíšeme Turingův stroj M přijímající L ($L = L(M)$)

Výpočet M se vstupem x

```
1 forall  $y \in \Sigma^*$  v shortlex uspořádání do  
2   | if  $\langle x, y \rangle \in B$  then  
3   |   | přijmi
```

Důkaz „ $\Leftarrow$ “

- Předpokládáme, že existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

- Popíšeme Turingův stroj M přijímající L ($L = L(M)$)

Výpočet M se vstupem x

```
1 forall  $y \in \Sigma^*$  v shortlex uspořádání do
2   |   if  $\langle x, y \rangle \in B$  then
3   |   |   přijmi
```

- $x \in L \implies (\exists y \in \Sigma^*)[\langle x, y \rangle \in B] \implies M(x)$ přijme

Důkaz „ $\Leftarrow$ “

- Předpokládáme, že existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

- Popíšeme Turingův stroj M přijímající L ($L = L(M)$)

Výpočet M se vstupem x

```
1 forall  $y \in \Sigma^*$  v shortlex uspořádání do
2   if  $\langle x, y \rangle \in B$  then
3     přijmi
```

- $x \in L \implies (\exists y \in \Sigma^*)[\langle x, y \rangle \in B] \implies M(x)$ přijme
- $x \notin L \implies (\forall y \in \Sigma^*)[\langle x, y \rangle \notin B] \implies M(x) \uparrow$

Důkaz „ $\Leftarrow$ “

- Předpokládáme, že existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

- Popíšeme Turingův stroj M přijímající L ($L = L(M)$)

Výpočet M se vstupem x

```
1 forall  $y \in \Sigma^*$  v shortlex uspořádání do
2   if  $\langle x, y \rangle \in B$  then
3     přijmi
```

- $x \in L \implies (\exists y \in \Sigma^*)[\langle x, y \rangle \in B] \implies M(x)$ přijme
- $x \notin L \implies (\forall y \in \Sigma^*)[\langle x, y \rangle \notin B] \implies M(x) \uparrow$
- Dohromady $L = L(M)$

Existenční kvantifikace (příklad)

$$\begin{aligned} L_u &= \{ \langle M, x \rangle \mid x \in L(M) \} \\ &= \{ \langle M, x \rangle \mid (\exists n \in \mathbb{N}) [\underbrace{M(x) \text{ přijme do } n \text{ kroků}}] \} \end{aligned}$$

Rozhodnutelná podmínka,
stačí simulovat $M(x)$ po n kroků

Existenční kvantifikace (příklad)

$$\begin{aligned} L_u &= \{ \langle M, x \rangle \mid x \in L(M) \} \\ &= \{ \langle M, x \rangle \mid (\exists n \in \mathbb{N}) \underbrace{[M(x) \text{ přijme do } n \text{ kroků}]} \} \end{aligned}$$

Rozhodnutelná podmínka,
stačí simulovat $M(x)$ po n kroků

- Následující jazyk je rozhodnutelný

$$B = \{ \langle M, x, n \rangle \mid M(x) \text{ přijme do } n \text{ kroků} \}$$

Existenční kvantifikace (příklad)

$$\begin{aligned} L_u &= \{ \langle M, x \rangle \mid x \in L(M) \} \\ &= \{ \langle M, x \rangle \mid (\exists n \in \mathbb{N}) [\underbrace{M(x) \text{ přijme do } n \text{ kroků}}] \} \end{aligned}$$

Rozhodnutelná podmínka,
stačí simulovat $M(x)$ po n kroků

- Následující jazyk je rozhodnutelný

$$B = \{ \langle M, x, n \rangle \mid M(x) \text{ přijme do } n \text{ kroků} \}$$

- Částečně rozhodnutelný jazyk L_u můžeme zapsat jako

$$L_u = \{ \langle M, x \rangle \mid (\exists n \in \mathbb{N}) [\langle M, x, n \rangle \in B] \}$$

Uzavřenost na existenční kvantifikaci

Důsledek

Je-li B částečně rozhodnutelný jazyk, pak jazyk

$$A = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

je též částečně rozhodnutelný.

Uzavřenost na existenční kvantifikaci

Důsledek

Je-li B částečně rozhodnutelný jazyk, pak jazyk

$$A = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

je též částečně rozhodnutelný.

Důkaz.

- Existuje rozhodnutelný jazyk C splňující

$$B = \{\langle x, y \rangle \in \Sigma^* \mid (\exists z \in \Sigma^*)[\langle x, y, z \rangle \in C]\}$$



Uzavřenost na existenční kvantifikaci

Důsledek

Je-li B částečně rozhodnutelný jazyk, pak jazyk

$$A = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

je též částečně rozhodnutelný.

Důkaz.

- Existuje rozhodnutelný jazyk C splňující

$$B = \{\langle x, y \rangle \in \Sigma^* \mid (\exists z \in \Sigma^*)[\langle x, y, z \rangle \in C]\}$$

- Platí $A = \{x \in \Sigma^* \mid (\exists \langle y, z \rangle \in \Sigma^*)[\langle x, y, z \rangle \in C]\}$



Uzavřenost na existenční kvantifikaci

Důsledek

Je-li B částečně rozhodnutelný jazyk, pak jazyk

$$A = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

je též částečně rozhodnutelný.

Důkaz.

- Existuje rozhodnutelný jazyk C splňující

$$B = \{\langle x, y \rangle \in \Sigma^* \mid (\exists z \in \Sigma^*)[\langle x, y, z \rangle \in C]\}$$

- Platí $A = \{x \in \Sigma^* \mid (\exists \langle y, z \rangle \in \Sigma^*)[\langle x, y, z \rangle \in C]\}$
- A je částečně rozhodnutelný dle předchozí věty



Protipříklad typu 1

PROTIPŘÍKLAD TYPU 1

Instance: Zdrojový kód programu P .

Otázka: Existuje složené číslo n , které P přijme?

$$\begin{aligned} T1 &= \{ \langle P \rangle \mid (\exists n \in \mathbb{N})[n \text{ je složené číslo a } P(n) \text{ přijme}] \} \\ &= \{ \langle P \rangle \mid (n \in \mathbb{N})[n \text{ je složené číslo} \wedge \langle P, n \rangle \in L_u] \} \\ &= \{ \langle P \rangle \mid (\exists n \in \mathbb{N})[\langle P, n \rangle \in C \cap L_u] \}, \text{ kde} \end{aligned}$$

- $C = \{ \langle P, n \rangle \mid n \text{ je složené číslo} \}$ je rozhodnutelný jazyk
- L_u je částečně rozhodnutelný jazyk
- $C \cap L_u$ je tedy částečně rozhodnutelný jazyk

Jazyk $T1$ je tedy částečně rozhodnutelný.

Neprázdnost jazyka

PROBLÉM NEPRÁZDNÉHO JAZYKA

Instance: Kód Turingova stroje M

Otázka: Je $L(M) \neq \emptyset$?

$$\begin{aligned} \text{NE} &= \{ \langle M \rangle \mid L(M) \neq \emptyset \} \\ &= \{ \langle M \rangle \mid (\exists x \in \Sigma^*) [x \in L(M)] \} \\ &= \{ \langle M \rangle \mid (\exists x \in \Sigma^*) [\langle M, x \rangle \in L_u] \} \end{aligned}$$

- Jazyk L_u je částečně rozhodnutelný

Jazyk NE je tedy částečně rozhodnutelný.

Vyčíslitelnost jazyků

Výpis všech protipříkladů typu 1

PROTIPŘÍKLAD TYPU 1

Instance: Zdrojový kód programu P .

Otázka: Existuje složené číslo n , které P přijme?

- Odpovídá jazyku

$$T1 = \{\langle P \rangle \mid (\exists n \in \mathbb{N})[n \text{ je složené číslo a } P(n) \text{ přijme}]\}$$

Chceme program $\mathcal{E}$, který bude vypisovat všechny protipříklady typu 1 k danému programu P .

Program vypisující protipříklady typu 1

Výpočet $\mathcal{E}$ se vstupem P

$t \leftarrow 2$

while true do

for $a \leftarrow 2$ **to** t **do**

for $b \leftarrow 2$ **to** t **do**

$n \leftarrow a \cdot b$

 Simuluj $P(n)$ po nejvýš t kroků

if P přijal **then**

$\text{print}(n)$

$t \leftarrow t + 1$

-
- Pro pevné P jde o **enumerátor** jazyka

$$\text{T1P} = \{n \mid n \text{ je složené číslo a } P(n) \text{ přijme}\}$$

- Lze upravit tak, aby každé n bylo vypsáno jen jednou

Enumerátor

Enumerátorem pro jazyk L je Turingův stroj E , který

- ignoruje svůj vstup,
- vypisuje řetězce $w \in L$ na vyhrazenou výstupní pásku
 - například oddělené $\#$
- každý řetězec $w \in L$ je někdy vypsán TS E
- Je-li L nekonečný, E svou činnost nikdy neskončí

Enumerátor pro jazyk NE

$$NE = \{\langle M \rangle \mid L(M) \neq \emptyset\}$$

- Enumerátor pro jazyk NE řeší následující úlohu:
 - Vypiš kódy Turingových strojů, které přijímají alespoň jedno slovo

Enumerátor pro jazyk NE

```
1 forall  $\langle M, x, n \rangle \in \Sigma^*$  v shortlex uspořádání do  
2   |   Simuluj výpočet  $M(x)$  po nejvýš  $n$  kroků  
3   |   if  $M(x)$  přijal then  
4   |   |   Zapiš  $\langle M \rangle$  na výstup
```

- Každý kód $\langle M \rangle \in NE$ je vypsán nekonečný počet krát
- Stroje jsou vypisovány v neurčeném pořadí

Enumerátor pro jazyk NE

$$NE = \{\langle M \rangle \mid L(M) \neq \emptyset\}$$

Upravíme enumerátor tak, aby každý kód stroje M s neprázdným jazykem byl vypsán právě jednou

Enumerátor jazyka NE

```
1  $S \leftarrow$  prázdný seznam řetězců
2 forall  $\langle M, x, n \rangle \in \Sigma^*$  v shortlex uspořádání do
3   Simuluj výpočet  $M(x)$  po nejvýš  $n$  kroků
4   if  $M(x)$  přijal and  $\langle M \rangle \notin S$  then
5     Zapiš  $\langle M \rangle$  na výstup
6     Přidej  $\langle M \rangle$  do seznamu  $S$ 
```

Vyčíslitelnost částečně rozhodnutelných jazyků

Věta

Jazyk L je částečně rozhodnutelný, právě když pro něj existuje enumerátor E .

Vyčíslitelnost částečně rozhodnutelných jazyků

Věta

Jazyk L je částečně rozhodnutelný, právě když pro něj existuje enumerátor E .

Důkaz.

Důkaz ve dvou krocích

„ $\Rightarrow$ “ L je částečně rozhodnutelný $\Rightarrow$ existuje enumerátor E pro L

„ $\Leftarrow$ “ Existuje enumerátor E pro $L \Rightarrow L$ je částečně rozhodnutelný



Důkaz „ $\Rightarrow$ “

- L je částečně rozhodnutelný

Důkaz „ $\Rightarrow$ “

- L je částečně rozhodnutelný
- Existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

Důkaz „ $\Rightarrow$ “

- L je částečně rozhodnutelný
- Existuje rozhodnutelný jazyk B splňující

$$L = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y \rangle \in B]\}$$

Enumerátor E jazyka L

```
1 forall  $\langle x, y \rangle \in \Sigma^*$  v shortlex uspořádání do
2   if  $\langle x, y \rangle \in B$  then
3     do Zapiš  $x$  na výstup
```

- Lze upravit tak, aby se slova na výstupu neopakovala
- Prvky L jsou vypisovány v neurčeném pořadí

Důkaz „ $\Leftarrow$ “

- Máme enumerátor E pro jazyk L
- Následující Turingův stroj M přijímá L

Výpočet M se vstupem x

- 1 Simuluj E a sleduj výstup
 - 2 **if** E vypsál x **then přijmi**
-

$x \in L \implies E$ někdy vypíše x a $M(x)$ přijme

$x \notin L \implies E$ nikdy nevypíše x a $M(x)$ se nezastaví

Dohromady $L = L(M)$

Vyčíslitelnost jazyků a funkce

Důsledek

Nekonečný jazyk L je částečně rozhodnutelný, právě když je oborem hodnot nějaké totální algoritmicky vyčíslitelné funkce f (tj. $L = \text{rng } f$).

„ $\implies$ “ L částečně rozhodnutelný

- máme enumerátor E pro L
- Pro jednoduchost uvažujeme parametry f typu $\mathbb{N}$
- pro $i \in \mathbb{N}$ definujeme

$$f(i) = (i + 1)\text{-ní řetězec vypsáný } E$$

- $f(i)$ definováno pro každé $i \in \mathbb{N}$, protože jazyk L je nekonečný
- $\text{rng } f = L$, protože E vypisuje právě řetězce z L

Vyčíslitelnost jazyků a funkce

Důsledek

Nekonečný jazyk L je částečně rozhodnutelný, právě když je oborem hodnot nějaké totální algoritmicky vyčíslitelné funkce f (tj. $L = \text{rng } f$).

„ $\Leftarrow$ “ Máme funkci f

- Následující stroj E je enumerátor pro L

Výpočet E

1 **forall** $y \in \Sigma^*$ v shortlex pořadí **do**

2 $\perp$ Zapiš $f(y)$ na výstup

- $x \in L$
 - $\Leftrightarrow$ existuje y pro nějž $f(y) = x$
 - $\Leftrightarrow E$ vypíše x

Enumerátor pro jazyk prvočísel

$$\text{PRIME} = \{ \langle p \rangle \mid p \text{ je prvočíslo} \}$$

Úloha: vypisuj prvočísla v rostoucím pořadí

Enumerátor prvočísel

```
1 forall  $p \in \mathbb{N}$  v rostoucím pořadí do  
2   | if  $p$  je prvočíslo then  
3   |   | Zapiš  $\langle p \rangle$  na výstup
```

Lze zkonstruovat proto, že jazyk **PRIME** je rozhodnutelný.

Vyčíslitelnost rozhodnutelných jazyků

Věta

Jazyk L je rozhodnutelný, právě když pro něj existuje enumerátor E , který navíc vypisuje prvky L v shortlex pořadí.

Vyčíslitelnost rozhodnutelných jazyků

Věta

Jazyk L je rozhodnutelný, právě když pro něj existuje enumerátor E , který navíc vypisuje prvky L v shortlex pořadí.

Důkaz.

Důkaz ve dvou krocích

- „ $\implies$ “ L je rozhodnutelný $\implies$ existuje enumerátor E pro L , který vypisuje prvky L v shortlex pořadí
- „ $\impliedby$ “ Existuje enumerátor E pro L , který vypisuje prvky L v shortlex pořadí $\implies L$ je rozhodnutelný



Důkaz „ $\Rightarrow$ “

- L je rozhodnutelný
- Následující enumerátor E vypisuje slova L v shortlex pořadí

Enumerátor E jazyka L

```
1 forall  $x \in \Sigma^*$  v shortlex uspořádání do  
    // lze ověřit díky rozhodnutelnosti  $L$   
2     if  $x \in L$  then  
3         Zapiš  $x$  na výstup
```

V případě, že L je konečný jazyk, E se po vypsání posledního slova z L zacyklí.

Důkaz „ $\Leftarrow$ “

- Máme enumerátor E pro jazyk L
- E vypisuje prvky L v shortlex pořadí
- Rozlišíme dva případy
 - 1 L je konečný jazyk $\implies L$ je rozhodnutelný
 - Všechny konečné jazyky jsou rozhodnutelné
 - 2 L je nekonečný jazyk $\implies$ následující stroj M rozhoduje L

Výpočet M se vstupem x

- 1 Simuluj E a sleduj výstup
 - 2 **if** E vypsál x **then přijmi**
 - 3 **if** E vypsál řetězec $y > x$ **then odmítni**
-

L je nekonečný $\implies$ vždy existuje $y > x \implies$ algoritmus skončí

Vyčíslitelnost rozhodnutelných jazyků a funkce

Definice

Funkce $f : \Sigma^* \rightarrow \Sigma^*$ je **rostoucí**, pokud platí, že $u < v$ implikuje $f(u) < f(v)$ pro každé dva řetězce $u, v \in \Sigma^*$, kde $f(u) \downarrow$ a $f(v) \downarrow$.

Důsledek

Nekonečný jazyk L je rozhodnutelný, právě když je oborem hodnot nějaké rostoucí totální algoritmicky vyčíslitelné funkce f (tj. $L = \text{rng } f$).

Vyčíslitelnost rozhodnutelných jazyků a funkce

Důsledek

Nekonečný jazyk L je rozhodnutelný, právě když je oborem hodnot nějaké rostoucí totální algoritmicky vyčíslitelné funkce f (tj. $L = \text{rng } f$).

„ $\implies$ “ L je rozhodnutelný

- Máme enumerátor E , který vypisuje prvky L v shortlex pořadí
- Pro jednoduchost uvažujeme parametry f typu $\mathbb{N}$
- Pro $i \in \mathbb{N}$ definujeme

$$f(i) = (i + 1)\text{-ní řetězec vypsáný } E$$

- $f(i)$ definováno pro každé $i \in \mathbb{N}$, protože jazyk L je nekonečný
- $\text{rng } f = L$, protože E vypisuje právě řetězce z L
- f je rostoucí, protože E vypisuje prvky L v shortlex pořadí

Vyčíslitelnost rozhodnutelných jazyků a funkce

Důsledek

Nekonečný jazyk L je rozhodnutelný, právě když je oborem hodnot nějaké rostoucí totální algoritmicky vyčíslitelné funkce f (tj. $L = \text{rng } f$).

„ $\Leftarrow$ “ Máme funkci f

- Popíšeme enumerátor E pro L

Výpočet E

```
1 forall  $y \in \Sigma^*$  v shortlex pořadí do  
2   | Zapiš  $f(y)$  na výstup
```

- E vypisuje právě prvky L v shortlex pořadí, protože f je rostoucí
- E tedy ukazuje, že L je rozhodnutelný