# Věta o rekurzi a její aplikace

Jan Konečný

2. listopadu 2013

# Self-Reference

Nejdříve sestavíme stroj, který vypisuje svůj vlastní kód (při lib. vstupu).
Pomocné tvrzení:

## Lemma

*Existuje vyčíslitelná funkce $q : \Sigma^* \to \Sigma^*$, která jakýkoli řetězec $w$ zobrazí na $[P_w]$, kde $P_w$ je TS, který zapíše na pásku $w$ a pak zastaví.*

## Důkaz.

Následující stroj $Q$ vyčisluje $q(w)$:

*TS $Q$ pro slovo $w$:*

1. *Sestav TS $P_w$ který pro lib. vstup provádí:*
   - *smaž vstup,*
   - *zapiš $w$ na pásku,*
   - *zastav.*
2. *Vypiš $[P_w]$.*

□

# Self-Reference

TS $SELF$, který bude vypisovat sám sebe rozdělíme na dva moduly $A$,$B$.
Nejříve bude spuštěn $A$, pak $B$.
Budeme brát, že kód $SELF$ je kombinací kódů $A$ a $B$.

$$[SELF] = [AB]$$

$A$ bude rovno $P_{[B]}$, což je $q([B])$ (z předchozího lematu a jeho důkazu).

$B$ vypočítá $\langle A \rangle$: definovali jsme $A$ jako $P_{[B]}$, takže v okamžiku, kdy se spustí $B$ (tedy po skončení činnosti $A$), bude na pásce zapsáno $[B]$. $B$ tedy má k dispozici svůj vlastní kód a může $[A]$ vypočítat jako $q([B])$, pak má k dispozici $[A]$ i $[B]$, může tedy vypsat $[SELF]$.

# ... podrobněji

TS $A = P_{[B]}$

TS $B$ pro $[M]$, kde $M$ je modul TS

- Vypočítej $q([M])$.
- Zkombinuj výsledek z předchozícho kroku s $[M]$ abys vytvořil celkový popis TS.
- Výsledek z předchozího kroku vytiskni.

TODO OBRAZEK
Všimněte si, že $B$ nepotřebuje znát kód $A$, nedochází k žádné definici kruhem.

# Self-reference ve Scheme

Stroj $P_w$ = kvótování a uzavření do procedury

```
(lambda(x) 'w))
```

Modul $B$ (zkombinovaní = obalení `let*`em)

```
(lambda (M)
  `(let* ((B ,M)
          (A (lambda(x) ',M)))
  (B (A ())))))
```

Modul $A$

```
(lambda (w)
   (lambda(x) 'kod_B))
```

# Self-reference ve Scheme (celý kód)

```
(let* ((B
        (lambda (M)
          `(let* ((B ,M) (A (lambda (x) ',M)))
             (B (A ())))))
       (A
        (lambda (x)
          '(lambda (M)
             `(let* ((B ,M) (A (lambda (x) ',M)))
                (B (A ()))))))))
  (B (A ())))
```

# Věta o rekurzi

## Věta

*Nechť $T$ je TS, který vyčisluje $t : \Sigma^* \times \Sigma^* \to \Sigma^*$. Existuje TS $R$, který počítá funkci $r : \Sigma^* \to \Sigma^*$, kde pro každé $w$ platí*

$$r(w) = t(\langle R \rangle, w)$$

**WTF?**

Tvrzení věty říká, že abychom vytvořili TS $R$, který pracuje s vlastním kódem, stačí umět udělat TS $T$, který pracuje stejně ale dostává kód $[R]$ na vstup.

## Důkaz.

Podobná konstrukce jako $SELF$ (NA TABULI) $\square$

# Věta o rekurzi ve Scheme

```
(let* (
       (B (lambda (M)
            `(let* ((B ,M)
                    (A (lambda(x) ',M))
                    (T display))
               (T (B (A ()))))))

       (A (lambda(x)
            (quote
             (lambda (M)
               `(let* ((B ,M)
                       (A (lambda(x) ',M))
                       (T display))
                  (T (B (A ()))))))))

       (T display))
  (display (B (A ()))))
```

# Aplikace věty o rekurzi

## Definice

Pokud $M$ je TS, pak *délka* kódu $M$ je $|[M]|$. Říkáme, že $M$ je *minimální*, pokud neexistuje ekvivalentní TS s kratším kódem. Nechť

$$MIN_{\mathsf{TM}} = \{[M] \mid \ M \text{ je minimální TS}\}.$$

## Věta

$MIN_{\mathsf{TM}}$ *není rekurzivní.*

## Důkaz.

NA TABULI □

# Aplikace věty o rekurzi: Věta o pevném bodě

**Věta**

*Nechť* $\Sigma^* \to \Sigma^*$ *je vyčíslitelná funkce. Pak existuje TS* $F$*, tž.* $t([F])$ *je kód ekvivalentního TS.*

**Důkaz.**

NA TABULI □