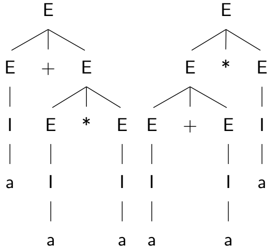
Definition 12.8 (Jednoznačnost a víceznačnost CFG)

- Bezkontextová gramatika $G = (V, T, P, S)$ je **víceznačná** pokud existuje aspoň jeden řetězec $w \in T^*$ pro který můžeme najít dva různé derivační stromy, oba s kořenem S dávající slovo w.
- V opačném případě nazýváme gramatiku **jednoznačnou**.
- Bezkontextový jazyk L je **jednoznačný**, jestliže existuje jednoznačná CFG G tak, že $L = L(G)$.
- **Bezkontextový jazyk L je (podstatně) nejednoznačný**, jestliže každá CFG G taková, že $L = L(G)$, je nejednoznačná. Takovému jazyku říkáme i **víceznačný**.

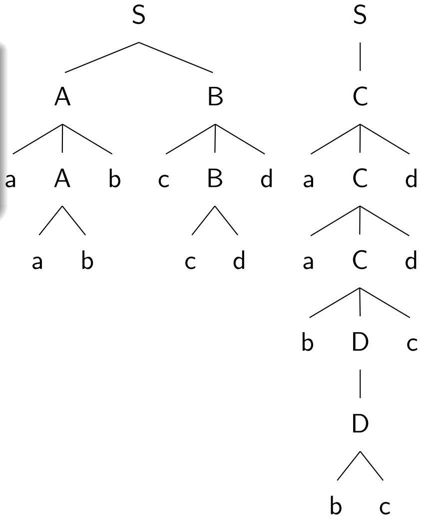
Example 12.6 (nejednoznačnost CFG)

Dva derivační stromy dávající $a + a * a$ ukazující víceznačnost gramatiky.



Příklad víceznačného jazyka

Dva derivační stromy pro *aabbccdd*.



Example 12.7 (Víceznačný jazyk)

Příklad víceznačného jazyka:

$$L = \{a^n b^n c^m d^m | n \geq 1, m \geq 1\} \cup$$
$$\cup \{a^n b^m c^m d^n | n \geq 1, m \geq 1\}.$$

1. $S \rightarrow AB|C$
2. $A \rightarrow aAb|ab$
3. $B \rightarrow cBd|cd$
4. $C \rightarrow aCd|aDd$
5. $D \rightarrow bDc|bc$.

Jakákoli gramatika pro daný jazyk bude generovat pro některá slova typu $a^n b^n c^n d^n$ dva různé derivační stromy.

Odstanění víceznačnosti gramatiky

- Neexistuje algoritmus, který nám řekne, zda je daná gramatika víceznačná.
- Existují bezkontextové jazyky, pro které neexistuje jednoznačná bezkontextová gramatika, pouze víceznačné CFG.
- Existují určitá doporučení pro odstranění víceznačnosti.

Víceznačnost má různé příčiny:

- Není respektovaná priorita operátorů.
- Posloupnost identických operátorů lze shlukovat zleva i zprava.
- $S \rightarrow$ if then S else $S|$ if then $S|\lambda$

  slovo 'if then if then else' má dva významy

  'if then (if then else)' nebo 'if then (if then) else'

  Řešení:

  - syntaktická chyba (Algol 60)
  - else patří k bližšímu if (preference pořadí pavidel)
  - závorky begin-end,odsazení v Python (asi nejčistší řešení).

Vynucení priority

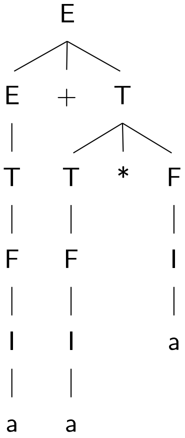
Řešením je zavést více různých proměnných, každou pro jednu úroveň 'priority'.
Konkrétně:

- **Faktor** je výraz který nesmí rozdělit žádný operátor.
  - identifikátory
  - výraz v závorkách
- **Term** je výraz, který nemůže rozdělit operátor $+$.
- **Výraz** může být rozdělen $*$ i $+$.

Jednoznačná gramatika pro výrazy:

1. $I \rightarrow a|b|Ia|Ib|I0|I1$
2. $F \rightarrow I|(E)$
3. $T \rightarrow F|T * F$
4. $E \rightarrow T|E + T$.

Jediný derivační strom pro $a + a * a$.



Nerozhodnutelné problémy o automatech a gramatikách

Definition 13.1 (Rozhodnutelný problém)

Problémem P myslíme matematicky/informaticky definovanou množinu otázek kódovatelnou řetězci nad abecedou Σ s odpověďmi $\in \{ano, ne\}$.
Problém je (algoritmicky) rozhodnutelný, pokud existuje Turingův stroj TM takový, že pro každý vstup $w \in P$ zastaví a navíc přijme právě když $P(w) = ano$ (tj. pro $P(w) = ne$ zastaví v ne–přijímacím stavu).
Problém který není algoritmicky rozhodnutelný nazýváme **nerozhodnutelný problém**.

'Rozhodnutelný' mluví o problémech, 'rekurzivní' o jazycích, jinak jde o 'totéž'.

Example 13.1 ('Problémy')

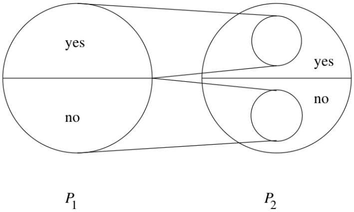
- Obsahuje vstupní slovo pět nul?
- Je vstupní slovo korektně definovaným kódem Turingova stroje v kódování výše?
- Zastaví TM kódu M nad slovem w?
- Zastaví TM kódu w nad slovem w?

Redukce

Definition 13.2 (Redukce)

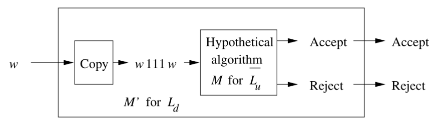
Redukcí problému P_1 na P_2, nazýváme algoritmus R, který pro každou instanci $w \in P_1$ zastaví a vydá $R(w) \in P_2$ tak, že

- $P_1(w) = ano$ právě když $P_2(R(w)) = ano$
- tj. i $P_1(w) = ne$ právě když $P_2(R(w)) = ne$.



Example 13.2

Redukce TM pro L_d na TM pro $\overline{L_u}$:



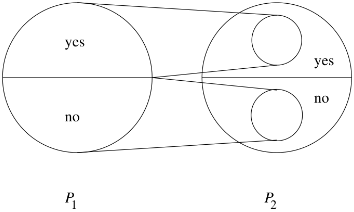
- P_1 = Nepřijímá TM reprezentovaný w vstupní slovo w?
- P_2 = Nepřijímá TM reprezentovaný M vstupní slovo w?

Věta o (ne)rozhodnutelnosti díky redukci

Theorem 13.1 (Redukce)

Pokud existuje redukce problému P_1 na P_2, pak:

- *Pokud P_1 je nerozhodnutelný, pak je nerozhodnutelný i P_2.*
- *Pokud P_1 není rekurzivně spočetný, pak není RE ani P_2.*



Proof.

- Předpokládejme P_1 je nerozhodnutelný. Je–li možné rozhodnout P_2, pak můžeme zkombinovat redukci P_1 na P_2 s algoritmem rozhodujícím P_2 pro konstrukci algoritmu rozhodujícího P_1. Proto je P_2 nerozhodnutelný.
- Předpokládejme P_1 ne–RE, ale P_2 je RE. Podobně jako výše zkombinujeme redukci a výsledek P_2 k důkazu P_1 je RE; SPOR.

□

Problém zastavení

Víme: L_u je rekurzivně spočetný, ale není rekurzivní.

Definition 13.3 (Problém zastavení)

Instancí problému zastavení je dvojice řetězců $M, w \in \{0,1\}^*$.
Problém zastavení je najít alogritmus $Halt(M, w)$, který vydá 1 právě když stroj M zastaví na vstupu w, jinak vydá 0.

Theorem (Problém zastavení)

Problém zastavení není rozhodnutelný.

Proof.

- Redukujeme L_d na $Halt$.
- Předpokládejme, že máme algoritmus (Turingův stroj) pro $Halt()$.
- Modifikujeme ho na stroj $Halt_{no}(w)$; $w \in \{0,1\}^*$:
  - Pokud $Halt(w, w)$, spustíme nekonečný cyklus
  - jinak zastavíme.
- Otázka $Halt(Halt_{no}, Halt_{no})$ není řešitelná, proto algoritmus $Halt()$ nemůže existovat.

Postův korespondenční problém

Definition 13.4 (Postův korespondenční problém)

Instance **Postova korespondenčního problému (PCP)** jsou dva seznamy slov nad abecedou Σ značené $A = w_1, w_2, \ldots, w_k$ a $B = x_1, x_2, \ldots, x_k$ stejné délky k.
Pro každé i, dvojice (w_i, x_i) se nazývá **odpovídající** dvojice.
Instance PCP **má řešení**, pokud existuje posloupnost jednoho či více přirozených čísel $i_1, i_2, \ldots, i_m$ tak že $w_{i_1} w_{i_2} \ldots w_{i_m} = x_{i_1} x_{i_2} \ldots x_{i_m}$ tj. dostaneme stejné slovo. V tom případě říkáme, že posloupnost $i_1, i_2, \ldots, i_m$ **je řešení**.
Postův korespondenční problém je: Pro danou instanci PCP, rozhodněte, zda má řešení.

Example 13.3

	Seznam A	Seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

- $\Sigma = \{0, 1\}$, seznamy A,B v tabulce.
- Řešení $2, 1, 1, 3$ vytvoří slovo 101111110.
- Jiné řešení: 2,1,1,3,2,1,1,3.

Postův korespondenční problém

Definition 13.4 (Postův korespondenční problém)

Instance **Postova korespondenčního problému (PCP)** jsou dva seznamy slov nad abecedou Σ značené $A = w_1, w_2, \ldots, w_k$ a $B = x_1, x_2, \ldots, x_k$ stejné délky k. Pro každé i, dvojice (w_i, x_i) se nazývá **odpovídající** dvojice.

Instance PCP **má řešení**, pokud existuje posloupnost jednoho či více přirozených čísel $i_1, i_2, \ldots, i_m$ tak že $w_{i_1} w_{i_2} \ldots w_{i_m} = x_{i_1} x_{i_2} \ldots x_{i_m}$ tj. dostaneme stejné slovo. V tom případě říkáme, že posloupnost $i_1, i_2, \ldots, i_m$ **je řešení**.

Postův korespondenční problém je: Pro danou instanci PCP, rozhodněte, zda má řešení.

Example 13.3

	Seznam A	Seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

- $\Sigma = \{0, 1\}$, seznamy A,B v tabulce.
- Řešení $2, 1, 1, 3$ vytvoří slovo 101111110.
- Jiné řešení: 2,1,1,3,2,1,1,3.

Postův korespondenční problém

Definition 13.4 (Postův korespondenční problém)

Instance **Postova korespondenčního problému (PCP)** jsou dva seznamy slov nad abecedou Σ značené $A = w_1, w_2, \ldots, w_k$ a $B = x_1, x_2, \ldots, x_k$ stejné délky k. Pro každé i, dvojice (w_i, x_i) se nazývá **odpovídající** dvojice.
Instance PCP **má řešení**, pokud existuje posloupnost jednoho či více přirozených čísel $i_1, i_2, \ldots, i_m$ tak že $w_{i_1} w_{i_2} \ldots w_{i_m} = x_{i_1} x_{i_2} \ldots x_{i_m}$ tj. dostaneme stejné slovo. V tom případě říkáme, že posloupnost $i_1, i_2, \ldots, i_m$ **je řešení**.
Postův korespondenční problém je: Pro danou instanci PCP, rozhodněte, zda má řešení.

Example 13.3

	Seznam A	Seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

- $\Sigma = \{0, 1\}$, seznamy A,B v tabulce.
- Řešení $2, 1, 1, 3$ vytvoří slovo 101111110.
- Jiné řešení: 2,1,1,3,2,1,1,3.

Postův korespondenční problém

Definition 13.4 (Postův korespondenční problém)

Instance **Postova korespondenčního problému (PCP)** jsou dva seznamy slov nad abecedou Σ značené $A = w_1, w_2, \dots, w_k$ a $B = x_1, x_2, \dots, x_k$ stejné délky k. Pro každé i, dvojice (w_i, x_i) se nazývá **odpovídající** dvojice.
Instance PCP **má řešení**, pokud existuje posloupnost jednoho či více přirozených čísel $i_1, i_2, \dots, i_m$ tak že $w_{i_1} w_{i_2} \dots w_{i_m} = x_{i_1} x_{i_2} \dots x_{i_m}$ tj. dostaneme stejné slovo. V tom případě říkáme, že posloupnost $i_1, i_2, \dots, i_m$ **je řešení**.
Postův korespondenční problém je: Pro danou instanci PCP, rozhodněte, zda má řešení.

Example 13.3

	Seznam A	Seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

- $\Sigma = \{0, 1\}$, seznamy A,B v tabulce.
- Řešení $2, 1, 1, 3$ vytvoří slovo 101111110.
- Jiné řešení: 2,1,1,3,2,1,1,3.

Postův korespondenční problém

Definition 13.4 (Postův korespondenční problém)

Instance **Postova korespondenčního problému (PCP)** jsou dva seznamy slov nad abecedou Σ značené $A = w_1, w_2, \ldots, w_k$ a $B = x_1, x_2, \ldots, x_k$ stejné délky k. Pro každé i, dvojice (w_i, x_i) se nazývá **odpovídající** dvojice.
Instance PCP **má řešení**, pokud existuje posloupnost jednoho či více přirozených čísel $i_1, i_2, \ldots, i_m$ tak že $w_{i_1} w_{i_2} \ldots w_{i_m} = x_{i_1} x_{i_2} \ldots x_{i_m}$ tj. dostaneme stejné slovo. V tom případě říkáme, že posloupnost $i_1, i_2, \ldots, i_m$ **je řešení**.
Postův korespondenční problém je: Pro danou instanci PCP, rozhodněte, zda má řešení.

Example 13.3

	Seznam A	Seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

- $\Sigma = \{0, 1\}$, seznamy A,B v tabulce.
- Řešení $2, 1, 1, 3$ vytvoří slovo 101111110.
- Jiné řešení: 2,1,1,3,2,1,1,3.

Postův korespondenční problém

Definition 13.4 (Postův korespondenční problém)

Instance **Postova korespondenčního problému (PCP)** jsou dva seznamy slov nad abecedou Σ značené $A = w_1, w_2, \ldots, w_k$ a $B = x_1, x_2, \ldots, x_k$ stejné délky k. Pro každé i, dvojice (w_i, x_i) se nazývá **odpovídající** dvojice.
Instance PCP **má řešení**, pokud existuje posloupnost jednoho či více přirozených čísel $i_1, i_2, \ldots, i_m$ tak že $w_{i_1} w_{i_2} \ldots w_{i_m} = x_{i_1} x_{i_2} \ldots x_{i_m}$ tj. dostaneme stejné slovo. V tom případě říkáme, že posloupnost $i_1, i_2, \ldots, i_m$ **je řešení**.
Postův korespondenční problém je: Pro danou instanci PCP, rozhodněte, zda má řešení.

Example 13.3

	Seznam A	Seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

- $\Sigma = \{0, 1\}$, seznamy A,B v tabulce.
- Řešení $2, 1, 1, 3$ vytvoří slovo 101111110.
- Jiné řešení: 2,1,1,3,2,1,1,3.

Částečná řešení

Example 13.4

$\Sigma = \{0, 1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

A: $10101\cdots$
B: $101011\cdots$

Částečná řešení

Example 13.4

$\Sigma = \{0,1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

A: $10101\cdots$
B: $101011\cdots$

Částečná řešení

Example 13.4

$\Sigma = \{0, 1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

A: $10101\cdots$
B: $101011\cdots$

Částečná řešení

Example 13.4

$\Sigma = \{0,1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

- $i_2 = 1$, řetězce 1010 101101 nesouhlasí na 4.pozici.

A: $10101\cdots$
B: $101011\cdots$

Částečná řešení

Example 13.4

$\Sigma = \{0, 1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Částečná řešení

Example 13.4

$\Sigma = \{0, 1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \dots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \dots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \dots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10 \cdots$
  B: $101 \cdots$
- $i_2 = 1$, řetězce $\begin{matrix} 1010 \\ 101101 \end{matrix}$ nesouhlasí na 4.pozici.
- $i_2 = 2$, $\begin{matrix} 10011 \\ 10111 \end{matrix}$ nesouhlasí na 3.pozici.
- Je možné jen $i_2 = 3$.

A: $10101 \cdots$
B: $101011 \cdots$

- Jsme ve stejné pozici jako po volbě $i_1 = 1$.
- Nelze dostat oba řetězce na stejnou délku.

Částečná řešení

Example 13.4

$\Sigma = \{0, 1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

- $i_2 = 1$, řetězce $\begin{array}{l}1010\\101101\end{array}$ nesouhlasí na 4.pozici.
- $i_2 = 2$, $\begin{array}{l}10011\\10111\end{array}$ nesouhlasí na 3.pozici.
- Je možné jen $i_2 = 3$.

A: $10101\cdots$
B: $101011\cdots$

- Jsme ve stejné pozici jako po volbě $i_1 = 1$.
- Nelze dostat oba řetězce na stejnou délku.

Částečná řešení

Example 13.4

$\Sigma = \{0,1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$
- $i_2 = 1$, řetězce $\begin{array}{l}1010\\101101\end{array}$ nesouhlasí na 4.pozici.
- $i_2 = 2$, $\begin{array}{l}10011\\10111\end{array}$ nesouhlasí na 3.pozici.
- Je možné jen $i_2 = 3$.

A: $10101\cdots$
B: $101011\cdots$

- Jsme ve stejné pozici jako po volbě $i_1 = 1$.
- Nelze dostat oba řetězce na stejnou délku.

Částečná řešení

Example 13.4

$\Sigma = \{0,1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

- $i_2 = 1$, řetězce $\begin{matrix}1010\\101101\end{matrix}$ nesouhlasí na 4.pozici.
- $i_2 = 2$, $\begin{matrix}10011\\10111\end{matrix}$ nesouhlasí na 3.pozici.
- Je možné jen $i_2 = 3$.

A: $10101\cdots$
B: $101011\cdots$

- Jsme ve stejné pozici jako po volbě $i_1 = 1$.
- Nelze dostat oba řetězce na stejnou délku.

Částečná řešení

Example 13.4

$\Sigma = \{0,1\}$. Neexistuje řešení pro seznamy:

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011.

Definition 13.5 (Částečné řešení)

Částečným řešením nazýváme posloupnost indexů $i_1, i_2, \ldots, i_r$ taková že jeden z řetězců $w_{i_1}, w_{i_2}, \ldots, w_{i_r}$ a $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ je prefix druhého (i v případě, že řetězce nejsou totožné).

Lemma

Je–li posloupnost čísel řešením, pak je každý prefix částečným řešením.

Zdůvodnění:

- $i_1 = 1$, jinak by první symbol neodpovídal.
- Máme částečné řešení:
  A: $10\cdots$
  B: $101\cdots$

- $i_2 = 1$, řetězce $\begin{matrix}1010\\101101\end{matrix}$ nesouhlasí na 4.pozici.
- $i_2 = 2$, $\begin{matrix}10011\\10111\end{matrix}$ nesouhlasí na 3.pozici.
- Je možné jen $i_2 = 3$.

A: $10101\cdots$
B: $101011\cdots$

- Jsme ve stejné pozici jako po volbě $i_1 = 1$.
- Nelze dostat oba řetězce na stejnou délku.

Modifikovaný Postův korespondenční probém MPCP

Definition 13.6 (Modifikovaný Postův korespondenční probém MPCP)

Mějme PCP, tj. seznamy $A = w_1, w_2, \ldots, w_k$ a $B = x_1, x_2, \ldots, x_k$. Hledáme seznam 0 nebo více přirozených čísel $i_1, i_2, \ldots, i_m$ tak že $\mathbf{w_1}, w_{i_1}, w_{i_2}, \ldots, w_{i_m} = \mathbf{x_1}, x_{i_1}, x_{i_2}, \ldots, x_{i_m}$. V tom případě říkáme, že PCP **má iniciální řešení**.

Modifikovaný Postův korespondenční problém: má PCP iniciální řešení?

Example 13.5

Tento PCP nemá iniciální řešení.

	seznam A	seznam B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Proof:

- Částečné instance $\begin{matrix} 1 \\ 111 \end{matrix}$, $\begin{matrix} 11 \\ 111111 \end{matrix}$ se nikdy nesrovnají na stejnou délku.
- Jiné volby vedou k různým písmenům abecedy.

□

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.
- $\forall i = 1, \ldots, k$ def. z_i rozšířením x_i s $*$ **před** každým písmenem x_i.
- $y_0 = *y_1$, $z_0 = z_1$.
- $y_{k+1} = \$$, $z_{k+1} = *\$$.
- $i_1, i_2, \ldots, i_m$ je iniciální řešení, iff $0, i_1, i_2, \ldots, i_m, (k+1)$ je řešení PCP. □

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.
- $\forall i = 1, \ldots, k$ def. z_i rozšířením x_i s $*$ **před** každým písmenem x_i.
- $y_0 = *y_1$, $z_0 = z_1$.
- $y_{k+1} = \$$, $z_{k+1} = *\$$.
- $i_1, i_2, \ldots, i_m$ je iniciální řešení, iff $0, i_1, i_2, \ldots, i_m, (k+1)$ je řešení PCP. □

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.
- $\forall i = 1, \ldots, k$ def. z_i rozšířením x_i s $*$ **před** každým písmenem x_i.
- $y_0 = *y_1$, $z_0 = z_1$.
- $y_{k+1} = \$$, $z_{k+1} = *\$$.
- $i_1, i_2, \ldots, i_m$ je iniciální řešení, iff $0, i_1, i_2, \ldots, i_m, (k+1)$ je řešení PCP. □

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.
- $\forall i = 1, \ldots, k$ def. z_i rozšířením x_i s $*$ **před** každým písmenem x_i.
- $y_0 = *y_1$, $z_0 = z_1$.
- $y_{k+1} = \$$, $z_{k+1} = *\$$.
- $i_1, i_2, \ldots, i_m$ je iniciální řešení, iff $0, i_1, i_2, \ldots, i_m, (k+1)$ je řešení PCP.

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.
- $\forall i = 1, \ldots, k$ def. z_i rozšířením x_i s $*$ **před** každým písmenem x_i.
- $y_0 = *y_1$, $z_0 = z_1$.
- $y_{k+1} = \$$, $z_{k+1} = *\$$.
- $i_1, i_2, \ldots, i_m$ je iniciální řešení, iff $0, i_1, i_2, \ldots, i_m, (k+1)$ je řešení PCP.

MPCP redukce na PCP

Lemma 13.1 (Red. MPCP na PCP)

$w \in MPCP$ má iniciální řešení, právě když má $R(w)$ řešení.

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Example 13.6 (MPCP redukce na PCP.)

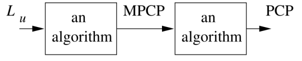
	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Proof:

- Vezměme nové symboly $*, \$ \notin \Sigma$.
- $\forall i = 1, \ldots, k$ definujeme y_i rozšířením w_i s $*$ za každým písmenem w_i.
- $\forall i = 1, \ldots, k$ def. z_i rozšířením x_i s $*$ **před** každým písmenem x_i.
- $y_0 = *y_1$, $z_0 = z_1$.
- $y_{k+1} = \$$, $z_{k+1} = *\$$.
- $i_1, i_2, \ldots, i_m$ je iniciální řešení, iff $0, i_1, i_2, \ldots, i_m, (k+1)$ je řešení PCP.

Nerozhodnutelnost PCP

- Chceme dokázat, že PCP je algoritmicky nerozhodnutelný.
- Redukovali jsme MPCP na PCP (minulý slajd)
- a redukujeme L_u na MPCP.



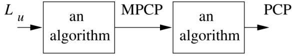
Algorithm: Redukce L_u na MPCP

Konstruujeme MPCP pro TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, který nikdy nepíše B a nejde hlavou doleva od počáteční pozice. Nechť $w \in \Sigma^*$ je vstupní slovo.

seznam A	seznam B	
$\#$	$\#q_0w\#$	
X	X	$\forall X \in \Gamma$
$\#$	$\#$	
qX	Yp	pro $\delta(q, X) = (p, Y, R)$
ZqX	pZY	pro $\delta(q, X) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
$q\#$	$Yp\#$	pro $\delta(q, B) = (p, Y, R)$
$Zq\#$	$pZY\#$	pro $\delta(q, B) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
XqY	q	$q \in F$, přijímající stav
Xq	q	$q \in F$
qY	q	$q \in F$
$q\#\#$	$q\#$	$q \in F$.

Nerozhodnutelnost PCP

- Chceme dokázat, že PCP je algoritmicky nerozhodnutelný.
- Redukovali jsme MPCP na PCP (minulý slajd)
- a redukujeme L_u na MPCP.



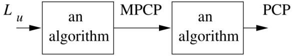
Algorithm: Redukce L_u na MPCP

Konstruujeme MPCP pro TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, který nikdy nepíše B a nejde hlavou doleva od počáteční pozice. Nechť $w \in \Sigma^*$ je vstupní slovo.

seznam A	seznam B	
$\#$	$\#q_0w\#$	
X	X	$\forall X \in \Gamma$
$\#$	$\#$	
qX	Yp	pro $\delta(q, X) = (p, Y, R)$
ZqX	pZY	pro $\delta(q, X) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
$q\#$	$Yp\#$	pro $\delta(q, B) = (p, Y, R)$
$Zq\#$	$pZY\#$	pro $\delta(q, B) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
XqY	q	$q \in F$, přijímající stav
Xq	q	$q \in F$
qY	q	$q \in F$
$q\#\#$	$q\#$	$q \in F$.

Nerozhodnutelnost PCP

- Chceme dokázat, že PCP je algoritmicky nerozhodnutelný.
- Redukovali jsme MPCP na PCP (minulý slajd)
- a redukujeme L_u na MPCP.



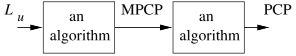
Algorithm: Redukce L_u na MPCP

Konstruujeme MPCP pro TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, který nikdy nepíše B a nejde hlavou doleva od počáteční pozice. Nechť $w \in \Sigma^*$ je vstupní slovo.

seznam A	seznam B	
$\#$	$\#q_0w\#$	
X	X	$\forall X \in \Gamma$
$\#$	$\#$	
qX	Yp	pro $\delta(q, X) = (p, Y, R)$
ZqX	pZY	pro $\delta(q, X) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
$q\#$	$Yp\#$	pro $\delta(q, B) = (p, Y, R)$
$Zq\#$	$pZY\#$	pro $\delta(q, B) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
XqY	q	$q \in F$, přijímající stav
Xq	q	$q \in F$
qY	q	$q \in F$
$q\#\#$	$q\#$	$q \in F$.

Nerozhodnutelnost PCP

- Chceme dokázat, že PCP je algoritmicky nerozhodnutelný.
- Redukovali jsme MPCP na PCP (minulý slajd)
- a redukujeme L_u na MPCP.



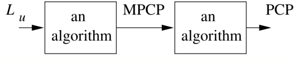
Algorithm: Redukce L_u na MPCP

Konstruujeme MPCP pro TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, který nikdy nepíše B a nejde hlavou doleva od počáteční pozice. Nechť $w \in \Sigma^*$ je vstupní slovo.

seznam A	seznam B	
$\#$	$\#q_0w\#$	
X	X	$\forall X \in \Gamma$
$\#$	$\#$	
qX	Yp	pro $\delta(q, X) = (p, Y, R)$
ZqX	pZY	pro $\delta(q, X) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
$q\#$	$Yp\#$	pro $\delta(q, B) = (p, Y, R)$
$Zq\#$	$pZY\#$	pro $\delta(q, B) = (p, Y, L)$, $Z \in \Gamma$ symbol pásky
XqY	q	$q \in F$, přijímající stav
Xq	q	$q \in F$
qY	q	$q \in F$
$q\#\#$	$q\#$	$q \in F$.

Nerozhodnutelnost PCP

- Chceme dokázat, že PCP je algoritmicky nerozhodnutelný.
- Redukovali jsme MPCP na PCP (minulý slajd)
- a redukujeme L_u na MPCP.



Algorithm: Redukce L_u na MPCP

Konstruujeme MPCP pro TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, který nikdy nepíše B a nejde hlavou doleva od počáteční pozice. Nechť $w \in \Sigma^*$ je vstupní slovo.

seznam A	seznam B	
$\#$	$\#q_0w\#$	
X	X	$\forall\ X \in \Gamma$
$\#$	$\#$	
qX	Yp	pro $\delta(q, X) = (p, Y, R)$
ZqX	pZY	pro $\delta(q, X) = (p, Y, L), Z \in \Gamma$ symbol pásky
$q\#$	$Yp\#$	pro $\delta(q, B) = (p, Y, R)$
$Zq\#$	$pZY\#$	pro $\delta(q, B) = (p, Y, L), Z \in \Gamma$ symbol pásky
XqY	q	$q \in F$, přijímající stav
Xq	q	$q \in F$
qY	q	$q \in F$
$q\#\#$	$q\#$	$q \in F$.

Example 13.7

Konvertujme TM $M = (\{q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_1, B, \{q_3\})$

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$	$\delta(q_i, B)$
q_1	$(q_2, 1, R)$	$(q_2, 0, L)$	$(q_2, 1, L)$
q_2	$(q_3, 0, L)$	$(q_1, 0, R)$	$(q_2, 0, R)$
q_3	–	–	–

a vstupní slovo $w = 01$ na instanci MPCP.

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

Seznam dvojic bez B symbolu (ve dvou tabulkách)

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \# \#$	$\#$

Example 13.7

Konvertujme TM $M = (\{q_1, q_2, q_3\}, \{0,1\}, \{0,1,X,Y,B\}, \delta, q_1, B, \{q_3\})$

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$	$\delta(q_i, B)$
q_1	$(q_2, 1, R)$	$(q_2, 0, L)$	$(q_2, 1, L)$
q_2	$(q_3, 0, L)$	$(q_1, 0, R)$	$(q_2, 0, R)$
q_3	–	–	–

a vstupní slovo $w = 01$ na instanci MPCP.

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

Seznam dvojic bez B symbolu (ve dvou tabulkách)

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

Example 13.7

Konvertujme TM $M = (\{q_1, q_2, q_3\}, \{0,1\}, \{0,1,X,Y,B\}, \delta, q_1, B, \{q_3\})$

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$	$\delta(q_i, B)$
q_1	$(q_2, 1, R)$	$(q_2, 0, L)$	$(q_2, 1, L)$
q_2	$(q_3, 0, L)$	$(q_1, 0, R)$	$(q_2, 0, R)$
q_3	–	–	–

a vstupní slovo $w = 01$ na instanci MPCP.

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

Seznam dvojic bez B symbolu (ve dvou tabulkách)

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

A : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

B : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

$A: \quad \# q_1 0 1 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

$B: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

A : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

B : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

$A:\quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

$B:\quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností
  $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

$A: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

$B: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

A : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

B : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11 \#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

A : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 0 1 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

B : $\# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 10 1 \# q_3 01 \# q_3 1 \# q_3 \#\#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

$A: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

$B: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \# \#$

MPCP simulace TM

seznam A	seznam B	zdroj
$q_1 0$	$1 q_2$	z $\delta(q_1, 0) = (q_2, 1, R)$
$0 q_1 1$	$q_2 00$	z $\delta(q_1, 1) = (q_2, 0, L)$
$1 q_1 1$	$q_2 10$	z $\delta(q_1, 1) = (q_2, 0, L)$
$0 q_1 \#$	$q_2 01\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$1 q_1 \#$	$q_2 11\#$	z $\delta(q_1, B) = (q_2, 1, L)$
$0 q_2 0$	$q_3 00$	z $\delta(q_2, 0) = (q_3, 0, L)$
$1 q_2 0$	$q_3 10$	z $\delta(q_2, 0) = (q_3, 0, L)$
$q_2 1$	$0 q_1$	z $\delta(q_2, 1) = (q_1, 0, R)$
$q_2 \#$	$0 q_2 \#$	z $\delta(q_2, B) = (q_2, 0, R)$

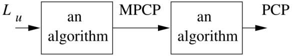
seznam A	seznam B
$\#$	$\# q_1 01 \#$
0	0
1	1
$\#$	$\#$
$0 q_3 0$	q_3
$0 q_3 1$	q_3
$1 q_3 0$	q_3
$1 q_3 1$	q_3
$0 q_3$	q_3
$1 q_3$	q_3
$q_3 0$	q_3
$q_3 1$	q_3
$q_3 \#\#$	$\#$

- M přijímá posloupností $q_1 01 \vdash 1 q_2 1 \vdash 10 q_1 \vdash 1 q_2 01 \vdash q_3 101$.

$A: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

$B: \quad \# q_1 01 \# 1 q_2 1 \# 10 q_1 \# 1 q_2 01 \# q_3 101 \# q_3 01 \# q_3 1 \# q_3 \#\#$

PCP je algoritmicky nerozhodnutelný



Theorem 13.2 (PCP je algoritmicky nerozhodnutelný)

Postův korespondenční problém PCP je algoritmicky nerozhodnutelný.

Proof.

Předchozí algoritmus redukuje L_u na MPCP. Chceme dokázat:

- M přijímá w právě když zkonstruovaný PCP má iniciální řešení.

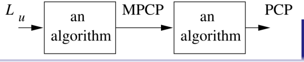
$\Rightarrow$ Pokud $w \in L(M)$, začneme iniciálním párem a simulujeme výpočet M na w.

$\Leftarrow$ Máme–li iniciální řešení PCP, odpovídá přijímajícímu výpočtu M nad w.

- MPCP musí začít první dvojicí.
- Dokud $q \notin F$, mazací pravidla se nepoužijí.
- Pokud $q \notin F$, částečné řešení je tvaru: $\begin{array}{l}\text{A:}x\\ \text{B:}xy\end{array}$, t.j. B je delší než A
- tedy musel skončit v přijímajícím stavu.

$\square$

PCP je algoritmicky nerozhodnutelný



Theorem 13.2 (PCP je algoritmicky nerozhodnutelný)

Postův korespondenční problém PCP je algoritmicky nerozhodnutelný.

Proof.

Předchozí algoritmus redukuje L_u na MPCP. Chceme dokázat:

- M přijímá w právě když zkonstruovaný PCP má iniciální řešení.

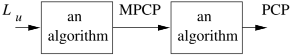
$\Rightarrow$ Pokud $w \in L(M)$, začneme iniciálním párem a simulujeme výpočet M na w.

$\Leftarrow$ Máme–li iniciální řešení PCP, odpovídá přijímajícímu výpočtu M nad w.

- MPCP musí začít první dvojicí.
- Dokud $q \notin F$, mazací pravidla se nepoužijí.
- Pokud $q \notin F$, částečné řešení je tvaru: $\begin{matrix} \text{A:}x \\ \text{B:}xy \end{matrix}$, t.j. B je delší než A
- tedy musel skončit v přijímajícím stavu.

$\square$

PCP je algoritmicky nerozhodnutelný



Theorem 13.2 (PCP je algoritmicky nerozhodnutelný)

Postův korespondenční problém PCP je algoritmicky nerozhodnutelný.

Proof.

Předchozí algoritmus redukuje L_u na MPCP. Chceme dokázat:

- M přijímá w právě když zkonstruovaný PCP má iniciální řešení.

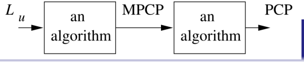
$\Rightarrow$ Pokud $w \in L(M)$, začneme iniciálním párem a simulujeme výpočet M na w.

$\Leftarrow$ Máme–li iniciální řešení PCP, odpovídá přijímajícímu výpočtu M nad w.

- MPCP musí začít první dvojicí.
- Dokud $q \notin F$, mazací pravidla se nepoužijí.
- Pokud $q \notin F$, částečné řešení je tvaru: $\begin{array}{l} A{:}x \\ B{:}xy \end{array}$, t.j. B je delší než A
- tedy musel skončit v přijímajícím stavu.

$\square$

PCP je algoritmicky nerozhodnutelný



Theorem 13.2 (PCP je algoritmicky nerozhodnutelný)

Postův korespondenční problém PCP je algoritmicky nerozhodnutelný.

Proof.

Předchozí algoritmus redukuje L_u na MPCP. Chceme dokázat:

- M přijímá w právě když zkonstruovaný PCP má iniciální řešení.

$\Rightarrow$ Pokud $w \in L(M)$, začneme iniciálním párem a simulujeme výpočet M na w.

$\Leftarrow$ Máme–li iniciální řešení PCP, odpovídá přijímajícímu výpočtu M nad w.

- MPCP musí začít první dvojicí.
- Dokud $q \notin F$, mazací pravidla se nepoužijí.
- Pokud $q \notin F$, částečné řešení je tvaru: $\begin{array}{l}\text{A}:x\\ \text{B}:xy\end{array}$, t.j. B je delší než A
- tedy musel skončit v přijímajícím stavu.

$\square$

Algoritmická rozhodnutelnost u CFL

Pro bezkontextové jazyky je algoritmicky rozhodnutelné

- zda dané slovo patří či nepatří do jazyka
  - prázdné slovo zvlášť
  - pak algoritmus CYK
  - nebo otestovat všechny derivace s $2|w| - 1$ pravidly,
- zda je jazyk prázdný
  - algoritmus redukce gramatiky (ne–nenerujících a nedosažitelných), zjistíme, zda lze z S generovat terminální slovo

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).

Redukujeme PKP na náš problém.

Mějme instanci PCP ($A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k$), množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$
\begin{array}{lll}
G_A & A \rightarrow & w_1Aa_1|w_2Aa_2|\ldots|w_kAa_k| \\
 & & w_1a_1|w_2a_2|\ldots|w_ka_k \\
G_B & B \rightarrow & x_1Ba_1|x_2Ba_2|\ldots|x_kBa_k| \\
 & & x_1a_1|x_2a_2|\ldots|x_ka_k \\
G_{AB} & \{S \rightarrow & A|B\} \cup G_A \cup G_B.
\end{array}
$$

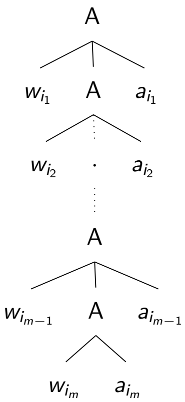
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).



Redukujeme PKP na náš problém.

Mějme instanci PCP ($A = w_1, w_2, \dots, w_k, B = x_1, x_2, \dots, x_k$), množinu indexů $a_1, a_2, \dots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$
\begin{array}{lll}
G_A & A \rightarrow & w_1Aa_1|w_2Aa_2|\dots|w_kAa_k| \\
 & & w_1a_1|w_2a_2|\dots|w_ka_k \\
G_B & B \rightarrow & x_1Ba_1|x_2Ba_2|\dots|x_kBa_k| \\
 & & x_1a_1|x_2a_2|\dots|x_ka_k \\
G_{AB} & \{S \rightarrow & A|B\} \cup G_A \cup G_B.
\end{array}
$$

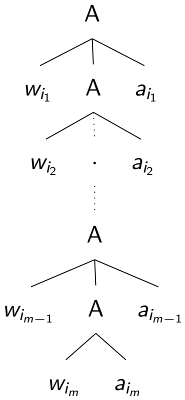
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).



Redukujeme PKP na náš problém.

Mějme instanci PCP $(A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k)$, množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$
\begin{array}{lll}
G_A & A \rightarrow & w_1Aa_1|w_2Aa_2|\ldots|w_kAa_k| \\
 & & w_1a_1|w_2a_2|\ldots|w_ka_k \\
G_B & B \rightarrow & x_1Ba_1|x_2Ba_2|\ldots|x_kBa_k| \\
 & & x_1a_1|x_2a_2|\ldots|x_ka_k \\
G_{AB} & \{S \rightarrow & A|B\} \cup G_A \cup G_B.
\end{array}
$$

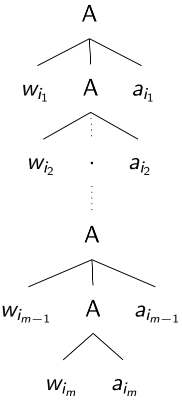
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).



Redukujeme PKP na náš problém.

Mějme instanci PCP $(A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k)$, množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

G_A	$A \rightarrow$	$w_1 A a_1 \vert w_2 A a_2 \vert \ldots \vert w_k A a_k \vert$
		$w_1 a_1 \vert w_2 a_2 \vert \ldots \vert w_k a_k$
G_B	$B \rightarrow$	$x_1 B a_1 \vert x_2 B a_2 \vert \ldots \vert x_k B a_k \vert$
		$x_1 a_1 \vert x_2 a_2 \vert \ldots \vert x_k a_k$
G_{AB}	$\{S \rightarrow$	$A \vert B\} \cup G_A \cup G_B$.

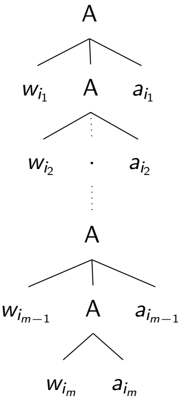
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).



Redukujeme PKP na náš problém.

Mějme instanci PCP $(A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k)$, množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$G_A \quad A \rightarrow \quad w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k |$$
$$w_1 a_1 | w_2 a_2 | \ldots | w_k a_k$$
$$G_B \quad B \rightarrow \quad x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k |$$
$$x_1 a_1 | x_2 a_2 | \ldots | x_k a_k$$
$$G_{AB} \quad \{S \rightarrow \quad A | B\} \cup G_A \cup G_B.$$

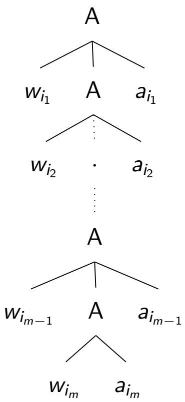
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).



Redukujeme PKP na náš problém.

Mějme instanci PCP ($A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k$), množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$\begin{array}{lll} G_A & A \rightarrow & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\ G_B & B \rightarrow & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \\ G_{AB} & \{S \rightarrow & A|B\} \cup G_A \cup G_B. \end{array}$$

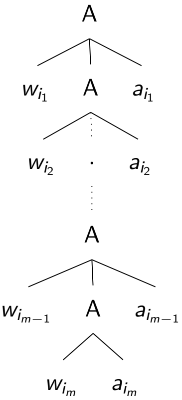
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stromy).



Redukujeme PKP na náš problém.

Mějme instanci PCP ($A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k$), množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$\begin{array}{lll} G_A & A \rightarrow & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\ G_B & B \rightarrow & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \\ G_{AB} & \{S \rightarrow & A | B\} \cup G_A \cup G_B. \end{array}$$

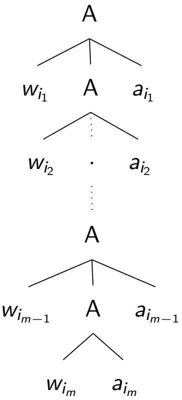
Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelnost víceznačnosti CFG

Theorem 13.3

Je algoritmicky nerozhodnutelné, zda je bezkontextová gramatika víceznačná (tj. existuje slovo jazyka gramatiky, které má dva různé derivační stomy).



Redukujeme PKP na náš problém.
Mějme instanci PCP ($A = w_1, w_2, \ldots, w_k, B = x_1, x_2, \ldots, x_k$), množinu indexů $a_1, a_2, \ldots, a_k \in N$ a tři gramatiky G_A, G_B, G_{AB}:

$$\begin{array}{lll}
G_A & A \rightarrow & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\
 & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\
G_B & B \rightarrow & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\
 & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \\
G_{AB} & \{S \rightarrow & A | B\} \cup G_A \cup G_B.
\end{array}$$

Gramatika G_{AB} je víceznačná právě když instance (A, B) PCP má řešení.

- Každé slovo v G_A má jednoznačnou derivaci (danou a_i vpravo). Podobně pro B.

Nerozhodnutelné problémy pro bezkontextové jazyky CFG

Theorem 13.4

Mějme G_1, G_2 bezkontextové gramatiky, R regulární výraz. Následující problémy jsou algoritmicky nerozhodnutelné:

1. *Je $L(G_1) \cap L(G_2) = \emptyset$?*
2. *Je $L(G_1) = T^*$ pro nějakou abecedu T?*
3. *Je $L(G_1) = L(G_2)$?*
4. *Je $L(G_1) = L(R)$?*
5. *Je $L(G_1) \subseteq L(G_2)$?*
6. *Je $L(R) \subseteq L(G_1)$?*

Průnik $L(G_1) \cap L(G_2) = \emptyset$

Proof: 1 $L(G_1) \cap L(G_2) = \emptyset$

Převedeme PKP na (1)

- zvolíme nové terminály $\{a_1, a_2, \ldots, a_m\}$ pro kódy indexů
$$\begin{array}{lll} G_1 & A \to & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\ G_2 & B \to & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \end{array}$$
- PKP má řešení právě když $L(G_1) \cap L(G_2) \neq \emptyset$
- první část se musí rovnat, druhá (a_i) zajišťuje stejné pořadí. $\square$

Vše $L(G) = T^*$

Proof: 2 $L(G) = T^*$

Převedeme PKP na (2):

- zvolíme nové terminály $\{a_1, a_2, \dots, a_m\}$ pro kódy indexů
  $$\begin{array}{lll} G_1 & A \to & w_1 A a_1 | w_2 A a_2 | \dots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \dots | w_k a_k \\ G_2 & B \to & x_1 B a_1 | x_2 B a_2 | \dots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \dots | x_k a_k \end{array}$$
- jazyky $L(G_1), L(G_2)$ jsou deterministické,
- tedy $\overline{L(G_1)}, \overline{L(G_2)}$ jsou deterministické CFL a $\overline{L(G_1)} \cup \overline{L(G_2)}$ je CFL
- máme CFG G gramatiku s $L(G) = \overline{L(G_1)} \cup \overline{L(G_2)}$
- PKP má řešení $\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset \Leftrightarrow L(G) = \overline{L(G_1)} \cup \overline{L(G_2)} \neq \Sigma^*$ □

- Poznámka: $L(G) = \emptyset$ je algoritmicky rozhodnutelné.
- CFL nejsou uzavřené na doplněk, pouze deterministické CFL ano.

Vše $L(G) = T^*$

Proof: 2 $L(G) = T^*$

Převedeme PKP na (2):

- zvolíme nové terminály $\{a_1, a_2, \ldots, a_m\}$ pro kódy indexů

$$\begin{array}{lll} G_1 & A \rightarrow & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\ G_2 & B \rightarrow & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \end{array}$$

- jazyky $L(G_1), L(G_2)$ jsou deterministické,
- tedy $\overline{L(G_1)}, \overline{L(G_2)}$ jsou deterministické CFL a $\overline{L(G_1)} \cup \overline{L(G_2)}$ je CFL
- máme CFG G gramatiku s $L(G) = \overline{L(G_1)} \cup \overline{L(G_2)}$
- PKP má řešení $\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset \Leftrightarrow L(G) = \overline{L(G_1)} \cup \overline{L(G_2)} \neq \Sigma^*$ □

- Poznámka: $L(G) = \emptyset$ je algoritmicky rozhodnutelné.
- CFL nejsou uzavřené na doplněk, pouze deterministické CFL ano.

Vše $L(G) = T^*$

Proof: 2 $L(G) = T^*$

Převedeme PKP na (2):

- zvolíme nové terminály $\{a_1, a_2, \ldots, a_m\}$ pro kódy indexů

$$\begin{array}{lll} G_1 & A \to & w_1Aa_1|w_2Aa_2|\ldots|w_kAa_k| \\ & & w_1a_1|w_2a_2|\ldots|w_ka_k \\ G_2 & B \to & x_1Ba_1|x_2Ba_2|\ldots|x_kBa_k| \\ & & x_1a_1|x_2a_2|\ldots|x_ka_k \end{array}$$

- jazyky $L(G_1), L(G_2)$ jsou deterministické,
- tedy $\overline{L(G_1)}, \overline{L(G_2)}$ jsou deterministické CFL a $\overline{L(G_1)} \cup \overline{L(G_2)}$ je CFL
- máme CFG G gramatiku s $L(G) = \overline{L(G_1)} \cup \overline{L(G_2)}$
- PKP má řešení $\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset \Leftrightarrow L(G) = \overline{L(G_1)} \cup \overline{L(G_2)} \neq \Sigma^*$ □

- Poznámka: $L(G) = \emptyset$ je algoritmicky rozhodnutelné.
- CFL nejsou uzavřené na doplněk, pouze deterministické CFL ano.

Vše $L(G) = T^*$

Proof: 2 $L(G) = T^*$

Převedeme PKP na (2):

- zvolíme nové terminály $\{a_1, a_2, \ldots, a_m\}$ pro kódy indexů

$$\begin{array}{lll} G_1 & A \rightarrow & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\ G_2 & B \rightarrow & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \end{array}$$

- jazyky $L(G_1), L(G_2)$ jsou deterministické,
- tedy $\overline{L(G_1)}, \overline{L(G_2)}$ jsou deterministické CFL a $\overline{L(G_1)} \cup \overline{L(G_2)}$ je CFL
- máme CFG G gramatiku s $L(G) = \overline{L(G_1)} \cup \overline{L(G_2)}$
- PKP má řešení $\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset \Leftrightarrow L(G) = \overline{L(G_1)} \cup \overline{L(G_2)} \neq \Sigma^*$ $\square$

- Poznámka: $L(G) = \emptyset$ je algoritmicky rozhodnutelné.
- CFL nejsou uzavřené na doplněk, pouze deterministické CFL ano.

Vše $L(G) = T^*$

Proof: 2 $L(G) = T^*$

Převedeme PKP na (2):

- zvolíme nové terminály $\{a_1, a_2, \ldots, a_m\}$ pro kódy indexů

$$\begin{array}{lll} G_1 & A \rightarrow & w_1 A a_1 | w_2 A a_2 | \ldots | w_k A a_k | \\ & & w_1 a_1 | w_2 a_2 | \ldots | w_k a_k \\ G_2 & B \rightarrow & x_1 B a_1 | x_2 B a_2 | \ldots | x_k B a_k | \\ & & x_1 a_1 | x_2 a_2 | \ldots | x_k a_k \end{array}$$

- jazyky $L(G_1), L(G_2)$ jsou deterministické,
- tedy $\overline{L(G_1)}, \overline{L(G_2)}$ jsou deterministické CFL a $\overline{L(G_1)} \cup \overline{L(G_2)}$ je CFL
- máme CFG G gramatiku s $L(G) = \overline{L(G_1)} \cup \overline{L(G_2)}$
- PKP má řešení $\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset \Leftrightarrow L(G) = \overline{L(G_1)} \cup \overline{L(G_2)} \neq \Sigma^*$.

- Poznámka: $L(G) = \emptyset$ je algoritmicky rozhodnutelné.
- CFL nejsou uzavřené na doplněk, pouze deterministické CFL ano.

Proof: 3-6

Zbylé algoritmicky nerozhodnutelné problémy.

3. Je $L(G_1) = L(G_2)$?
   - Důkaz: ať G_1 generuje Σ^*.
4. Je $L(G_1) = L(R)$?
   - Důkaz: za R zvolíme Σ^*.
5. Je $L(G_1) \subseteq L(G_2)$?
   - Důkaz: ať G_1 generuje Σ^*.
6. Je $L(R) \subseteq L(G_1)$?
   - Důkaz: za R zvolíme Σ^*. □

- Poznámka: $L(G) \subseteq L(R)$ je algoritmicky rozhodnutelné $L(G) \subseteq L(R) \Leftrightarrow L(G) \cap \overline{L(R)} = \emptyset$ a zároveň $(L(G) \cap \overline{L(R)})$ je CFL (uzavřenost operací)

Proof: 3-6

Zbylé algoritmicky nerozhodnutelné problémy.

3 Je $L(G_1) = L(G_2)$?
   - Důkaz: ať G_1 generuje Σ^*.

4 Je $L(G_1) = L(R)$?
   - Důkaz: za R zvolíme Σ^*.

5 Je $L(G_1) \subseteq L(G_2)$?
   - Důkaz: ať G_1 generuje Σ^*.

6 Je $L(R) \subseteq L(G_1)$?
   - Důkaz: za R zvolíme Σ^*. $\square$

- Poznámka: $L(G) \subseteq L(R)$ je algoritmicky rozhodnutelné $L(G) \subseteq L(R) \Leftrightarrow L(G) \cap \overline{L(R)} = \emptyset$ a zároveň $(L(G) \cap \overline{L(R)})$ je CFL (uzavřenost operací)

Proof: 3-6

Zbylé algoritmicky nerozhodnutelné problémy.

3 Je $L(G_1) = L(G_2)$?
- Důkaz: ať G_1 generuje Σ^*.

4 Je $L(G_1) = L(R)$?
- Důkaz: za R zvolíme Σ^*.

5 Je $L(G_1) \subseteq L(G_2)$?
- Důkaz: ať G_1 generuje Σ^*.

6 Je $L(R) \subseteq L(G_1)$?
- Důkaz: za R zvolíme Σ^*. □

- Poznámka: $L(G) \subseteq L(R)$ je algoritmicky rozhodnutelné $L(G) \subseteq L(R) \Leftrightarrow L(G) \cap \overline{L(R)} = \emptyset$ a zároveň $(L(G) \cap \overline{L(R)})$ je CFL (uzavřenost operací)

Proof: 3-6

Zbylé algoritmicky nerozhodnutelné problémy.

3 Je $L(G_1) = L(G_2)$?
- Důkaz: ať G_1 generuje Σ^*.

4 Je $L(G_1) = L(R)$?
- Důkaz: za R zvolíme Σ^*.

5 Je $L(G_1) \subseteq L(G_2)$?
- Důkaz: ať G_1 generuje Σ^*.

6 Je $L(R) \subseteq L(G_1)$?
- Důkaz: za R zvolíme Σ^*. $\square$

- Poznámka: $L(G) \subseteq L(R)$ je algoritmicky rozhodnutelné
$L(G) \subseteq L(R) \Leftrightarrow L(G) \cap \overline{L(R)} = \emptyset$ a zároveň $(L(G) \cap \overline{L(R)})$ je CFL (uzavřenost operací)

Proof: 3-6

Zbylé algoritmicky nerozhodnutelné problémy.

3 Je $L(G_1) = L(G_2)$?
  - Důkaz: ať G_1 generuje Σ^*.

4 Je $L(G_1) = L(R)$?
  - Důkaz: za R zvolíme Σ^*.

5 Je $L(G_1) \subseteq L(G_2)$?
  - Důkaz: ať G_1 generuje Σ^*.

6 Je $L(R) \subseteq L(G_1)$?
  - Důkaz: za R zvolíme Σ^*. $\square$

- Poznámka: $L(G) \subseteq L(R)$ je algoritmicky rozhodnutelné
  $L(G) \subseteq L(R) \Leftrightarrow L(G) \cap \overline{L(R)} = \emptyset$ a zároveň $(L(G) \cap \overline{L(R)})$ je CFL (uzavřenost operací)

Shrnutí

Popis nekonečných objektů konečnými prostředky

- regulární jazyky
  - konečné automaty (NFA, 2FA)
  - Nerode (rozklad), Kleene (elementární operace), pumpování
- bezkontextové jazyky
  - zásobníkové automaty (DPDA$\neq$ PDA)
  - pumpování
- kontextové jazyky
  - lineárně omezené automaty
  - monotonie
- rekurzivně spočetné jazyky
  - Turingovy stroje
  - algoritmická nerozhodnutelnost

použití nejen pro práci s jazyky.

Přehled kapitol

1. Úvod, Iterační lemma pro reg. jazyky
2. Redukovaný DFA a ekvivalence automatů, stavů
3. Nedeterministické λ–NFA, Operace zachovávající regularitu
4. Regulární výrazy, Kleeneova věta, Substituce, Homomorfizmus
5. Dvousměrné FA, Mealy a Moore stroje
6. Gramatiky, Chomského hierarchie, víceznačnost
7. Chomského NF, Pumping Lemma pro CFL
8. CYK – náležení do CFL
9. Zásobníkové automaty, Deterministické PDA
10. Uzávěrové vlastnosti, Dykovy jazyky
11. Turingův stroj, rozšíření
12. Lineárně omezené automaty, Univerzální TM, Diagonální jazyk
13. Nerozhodnutelné problémy, Postův korespondenční p.