

Deterministické zásobníkové automaty

Formálne jazyky a automaty (2)

Peter Kostolányi

25. apríla 2017

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

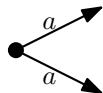
Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

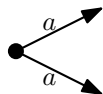
Zdroje nedeterminizmu v zásobníkových automatech

Zdroje nedeterminizmu v zásobníkových automatoch

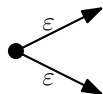


- ▶ $|\delta(p, a, Z)| \geq 2$
pre nejaké $q \in K$, $a \in \Sigma$ a $Z \in \Gamma$

Zdroje nedeterminizmu v zásobníkových automatoch

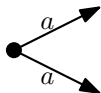


- ▶ $|\delta(p, a, Z)| \geq 2$
pre nejaké $q \in K$, $a \in \Sigma$ a $Z \in \Gamma$

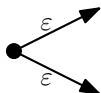


- ▶ $|\delta(p, \varepsilon, Z)| \geq 2$
pre nejaké $q \in K$ a $Z \in \Gamma$

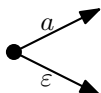
Zdroje nedeterminizmu v zásobníkových automatoch



- ▶ $|\delta(p, a, Z)| \geq 2$
pre nejaké $q \in K$, $a \in \Sigma$ a $Z \in \Gamma$

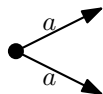


- ▶ $|\delta(p, \varepsilon, Z)| \geq 2$
pre nejaké $q \in K$ a $Z \in \Gamma$

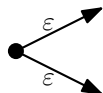


- ▶ $|\delta(p, a, Z)| \geq 1$ a $|\delta(p, \varepsilon, Z)| \geq 1$
pre nejaké $q \in K$, $a \in \Sigma$ a $Z \in \Gamma$

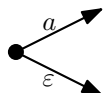
Zdroje nedeterminizmu v zásobníkových automatoch



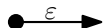
- ▶ $|\delta(p, a, Z)| \geq 2$
pre nejaké $q \in K$, $a \in \Sigma$ a $Z \in \Gamma$



- ▶ $|\delta(p, \varepsilon, Z)| \geq 2$
pre nejaké $q \in K$ a $Z \in \Gamma$



- ▶ $|\delta(p, a, Z)| \geq 1$ a $|\delta(p, \varepsilon, Z)| \geq 1$
pre nejaké $q \in K$, $a \in \Sigma$ a $Z \in \Gamma$



- ▶ Ak $|\delta(p, \varepsilon, Z)| = 1$ a $|\delta(p, c, Z)| = 0 \ \forall c \in \Sigma$:
prechod na ε možno vykonať deterministicky

Deterministické zásobníkové automaty (DPDA)

Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) *Pre všetky $q \in K$, $z \in \Sigma \cup \{\epsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*

Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) Pre všetky $q \in K$, $z \in \Sigma \cup \{\epsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*
- (ii) Ak $\delta(q, c, Z) \neq \emptyset$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$, tak $\delta(q, \epsilon, Z) = \emptyset$.*

Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) Pre všetky $q \in K$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*
- (ii) Ak $\delta(q, c, Z) \neq \emptyset$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$, tak $\delta(q, \varepsilon, Z) = \emptyset$.*

► Konfigurácia, krok výpočtu, $L(A)$, $N(A)$ – „zdedené“ od PDA

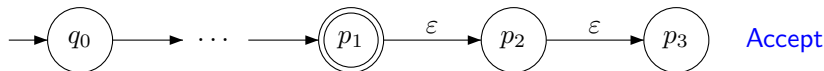
Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) Pre všetky $q \in K$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*
- (ii) Ak $\delta(q, c, Z) \neq \emptyset$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$, tak $\delta(q, \varepsilon, Z) = \emptyset$.*

► Konfigurácia, krok výpočtu, $L(A)$, $N(A)$ – „zdedené“ od PDA



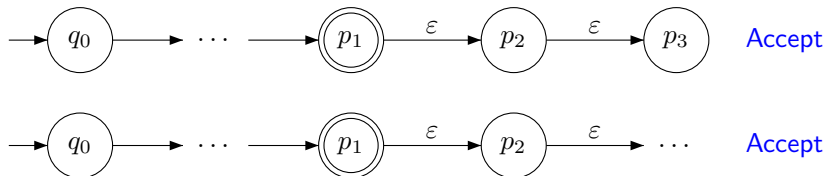
Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) Pre všetky $q \in K$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*
- (ii) Ak $\delta(q, c, Z) \neq \emptyset$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$, tak $\delta(q, \varepsilon, Z) = \emptyset$.*

► Konfigurácia, krok výpočtu, $L(A)$, $N(A)$ – „zdedené“ od PDA



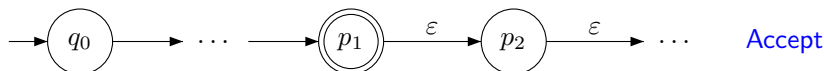
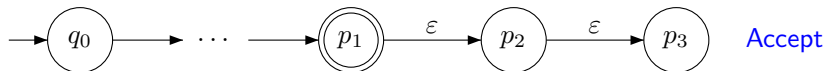
Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) Pre všetky $q \in K$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*
- (ii) Ak $\delta(q, c, Z) \neq \emptyset$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$, tak $\delta(q, \varepsilon, Z) = \emptyset$.*

► Konfigurácia, krok výpočtu, $L(A)$, $N(A)$ – „zdedené“ od PDA



► Alternatívna definícia pomocou čiastočnej prechodovej funkcie

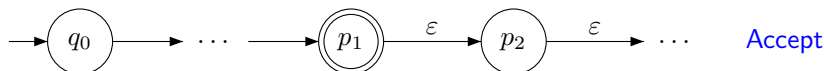
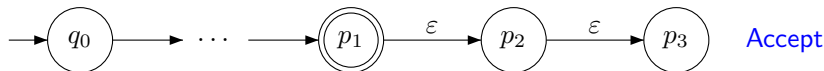
Deterministické zásobníkové automaty (DPDA)

Definícia

Deterministický zásobníkový automat je zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že:

- (i) *Pre všetky $q \in K$, $z \in \Sigma \cup \{\epsilon\}$ a $Z \in \Gamma$ je $|\delta(q, z, Z)| \leq 1$.*
- (ii) *Ak $\delta(q, c, Z) \neq \emptyset$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$, tak $\delta(q, \epsilon, Z) = \emptyset$.*

► Konfigurácia, krok výpočtu, $L(A)$, $N(A)$ – „zdedené“ od PDA



- Alternatívna definícia pomocou čiastočnej prechodovej funkcie
- Namiesto $\delta(p, c, Z) = \{(q, \gamma)\}$ píšeme aj $\delta(p, c, Z) = (q, \gamma)$

Deterministické bezkontextové jazyky

Deterministické bezkontextové jazyky

Definícia

Jazyk L je deterministický bezkontextový, ak existuje DPDA A taký, že $L(A) = L$ (teda A akceptuje jazyk L stavom).

Deterministické bezkontextové jazyky

Definícia

Jazyk L je deterministický bezkontextový, ak existuje DPDA A taký, že $L(A) = L$ (teda A akceptuje jazyk L stavom).

Označenie

Triedu všetkých deterministických bezkontextových jazykov označujeme ako $\mathcal{L}_{detCF}$.

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Motivácia

Motivácia

- ▶ Evidentne $\mathcal{L}_{detCF} \subseteq \mathcal{L}_{CF}$

Motivácia

- ▶ Evidentne $\mathcal{L}_{detCF} \subseteq \mathcal{L}_{CF}$
- ▶ Trieda $\mathcal{L}_{CF}$ nie je uzavretá na komplement

Motivácia

- ▶ Evidentne $\mathcal{L}_{detCF} \subseteq \mathcal{L}_{CF}$
- ▶ Trieda $\mathcal{L}_{CF}$ nie je uzavretá na komplement
- ▶ Dokážeme, že $\mathcal{L}_{detCF}$ je uzavretá na komplement

Motivácia

- ▶ Evidentne $\mathcal{L}_{detCF} \subseteq \mathcal{L}_{CF}$
- ▶ Trieda $\mathcal{L}_{CF}$ nie je uzavretá na komplement
- ▶ Dokážeme, že $\mathcal{L}_{detCF}$ je uzavretá na komplement
- ▶ Dôsledok: $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$

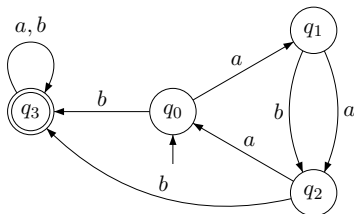
Uzavretosť triedy $\mathcal{R}$ na komplement (pripomenutie)

Uzavretosť triedy $\mathcal{R}$ na komplement (pripomenutie)

- ▶ Výmena akceptačných a neakceptačných stavov DKA

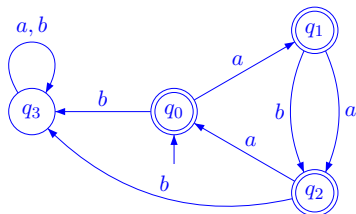
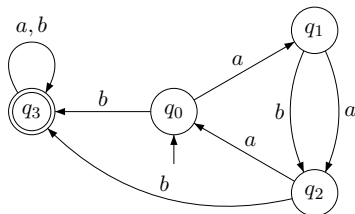
Uzavretosť triedy $\mathcal{R}$ na komplement (pripomenutie)

- Výmena akceptačných a neakceptačných stavov DKA



Uzavretosť triedy $\mathcal{R}$ na komplement (pripomenutie)

- Výmena akceptačných a neakceptačných stavov DKA



Konstrukcia DPDA akceptujúceho komplement

Konstrukcia DPDA akceptujúceho komplement

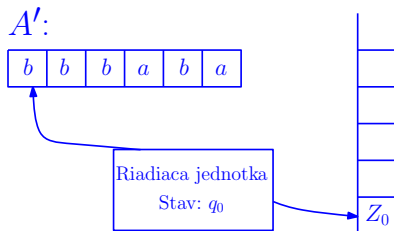
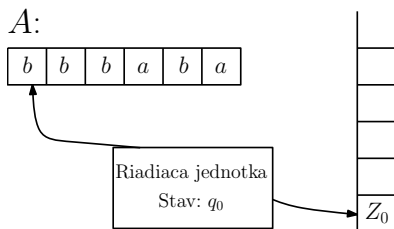
- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA

Konstrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“

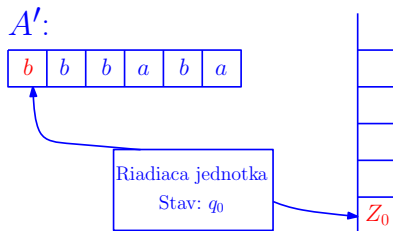
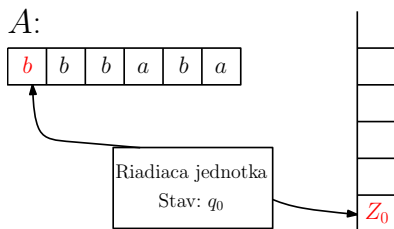
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



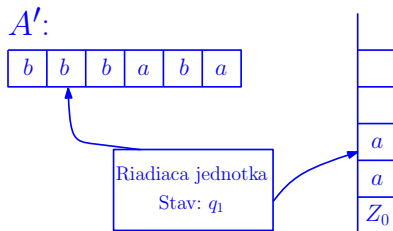
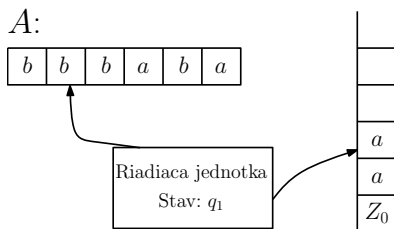
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



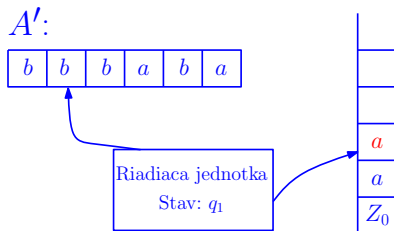
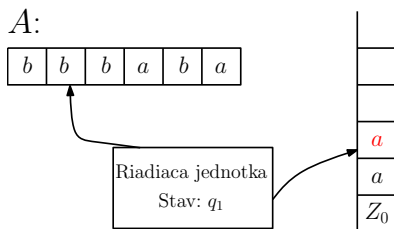
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



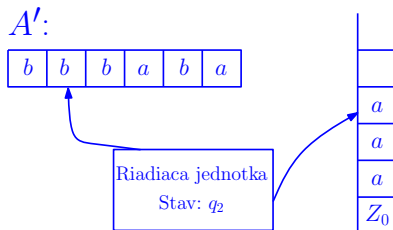
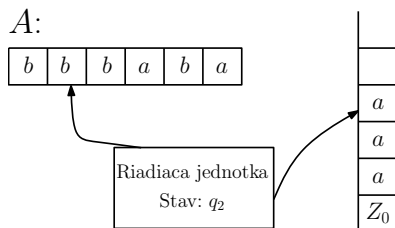
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



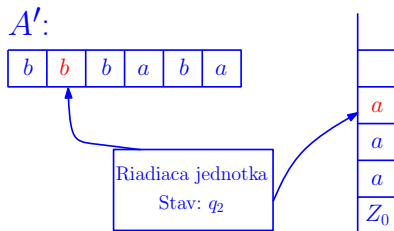
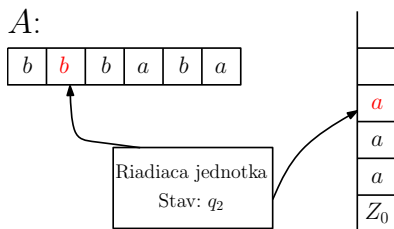
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



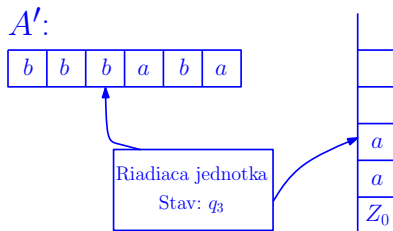
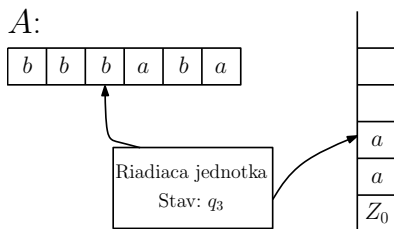
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



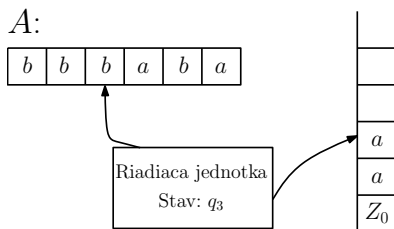
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“

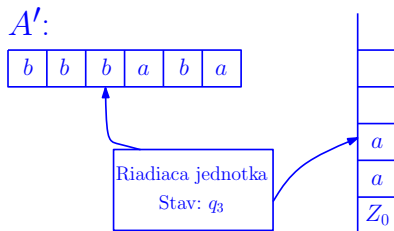


Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“

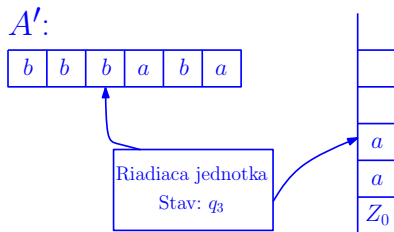
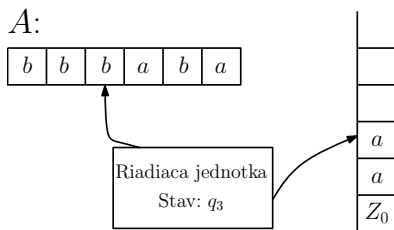


Nedefinovaný prechod!



Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“

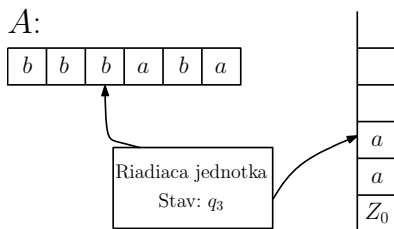


Nedefinovaný prechod!

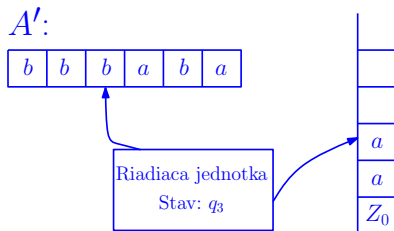
$$bbbaba \notin L(A)$$

Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 1: výmena akceptačných a neakceptačných stavov DPDA
- ▶ Problém 1: výpočet DPDA sa môže „zaseknúť“



Nedefinovaný prechod!
 $bbbaba \notin L(A)$



Nedefinovaný prechod!
 $bbbaba \notin L(A')$

Riešenie problému 1

Riešenie problému 1

Definícia

DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je „nezasekávajúci sa“ ak:

Riešenie problému 1

Definícia

DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je „nezasekávajúci sa“ ak:

- (i) Pre všetky $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$ je buď $\delta(q, c, Z) \neq \emptyset$, alebo $\delta(q, \varepsilon, Z) \neq \emptyset$.*

Riešenie problému 1

Definícia

DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je „nezasekávajúci sa“ ak:

- (i) Pre všetky $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$ je buď $\delta(q, c, Z) \neq \emptyset$, alebo $\delta(q, \varepsilon, Z) \neq \emptyset$.*
- (ii) A v žiadnom výpočte nevyprázdni zásobník.*

Riešenie problému 1

Definícia

DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je „nezasekávajúci sa“ ak:

- (i) Pre všetky $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$ je buď $\delta(q, c, Z) \neq \emptyset$, alebo $\delta(q, \varepsilon, Z) \neq \emptyset$.
- (ii) A v žiadnom výpočte nevyprázdni zásobník.

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- ▶ Nový „odpadový stav“ q_{odp} a „falošné dno“ D

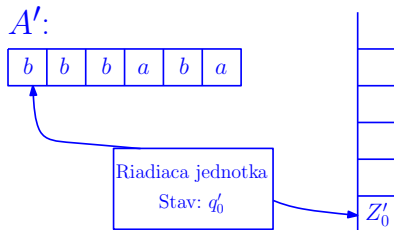
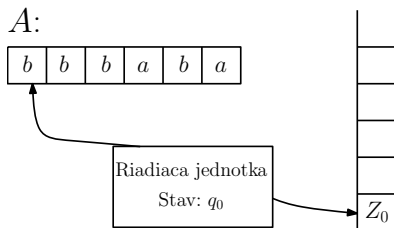
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



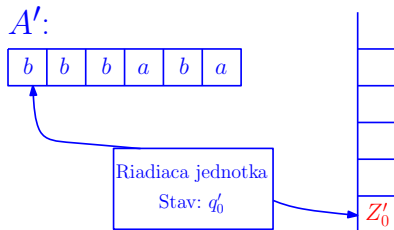
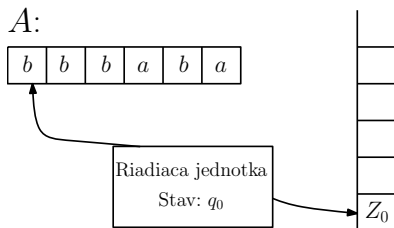
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



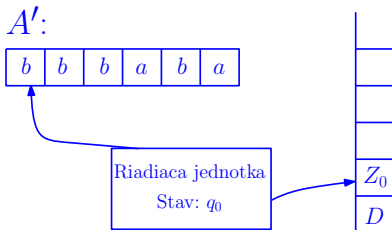
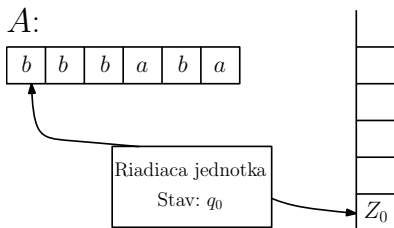
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



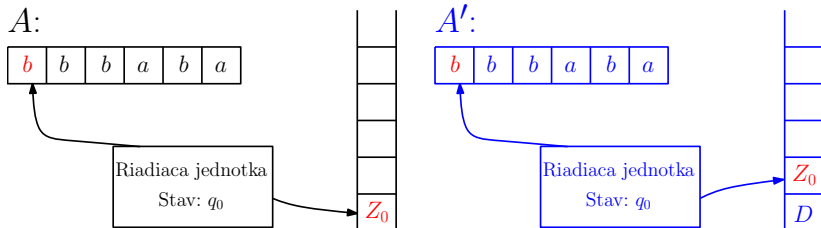
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



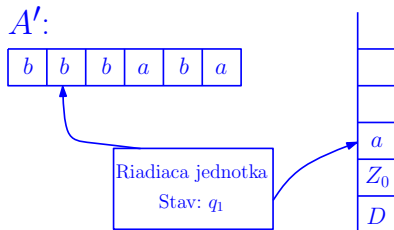
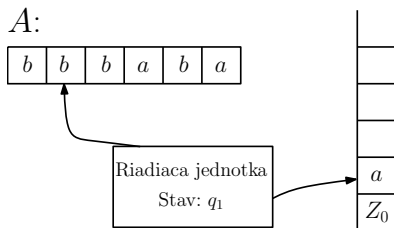
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



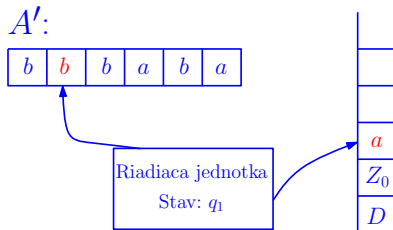
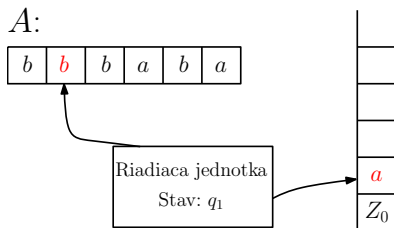
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



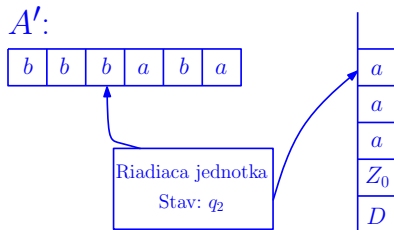
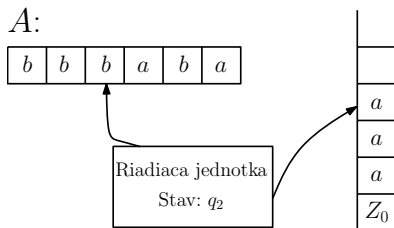
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



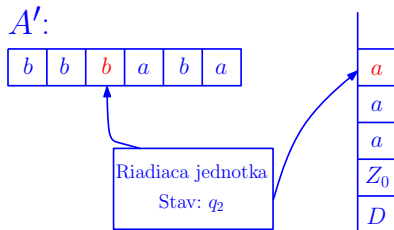
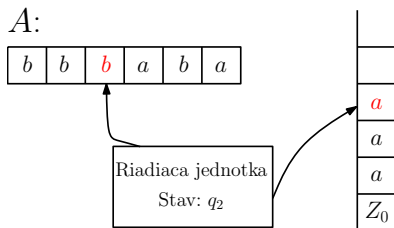
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



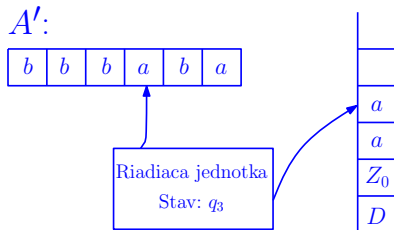
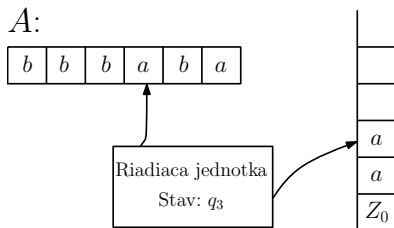
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



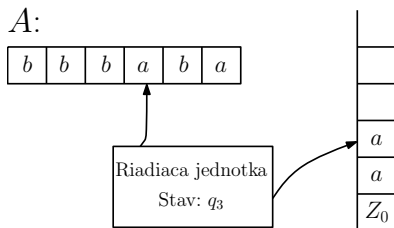
Riešenie problému 1

Lema

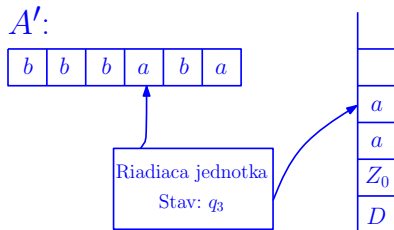
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



Nedefinovaný prechod!



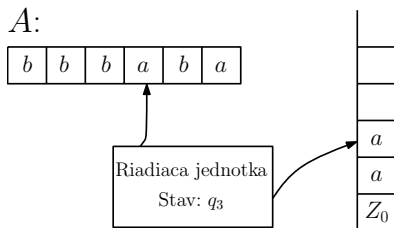
Riešenie problému 1

Lema

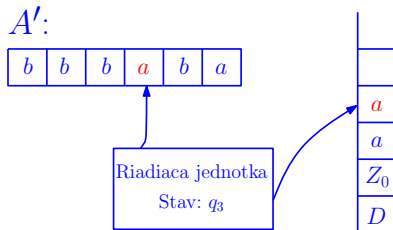
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



Nedefinovaný prechod!



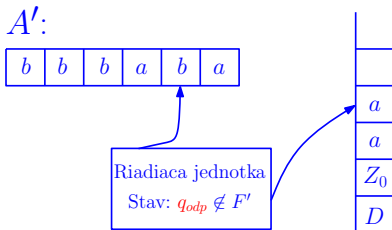
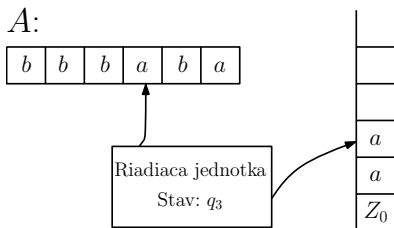
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



Nedefinovaný prechod!

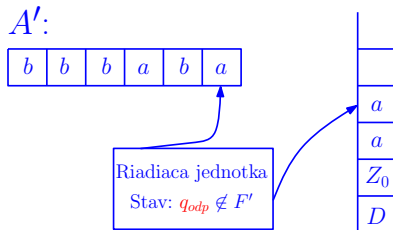
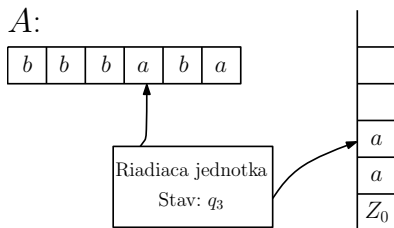
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



Nedefinovaný prechod!

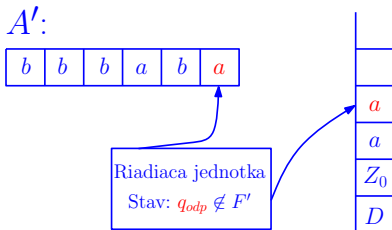
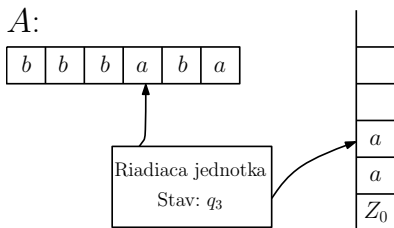
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



Nedefinovaný prechod!

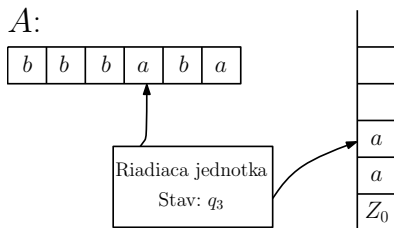
Riešenie problému 1

Lema

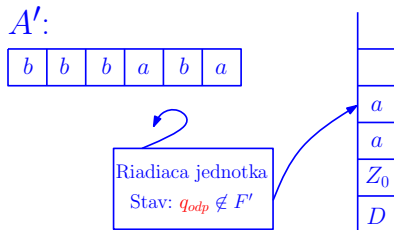
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 1:



Nedefinovaný prechod!



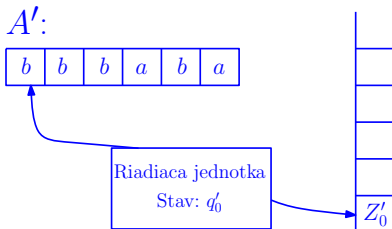
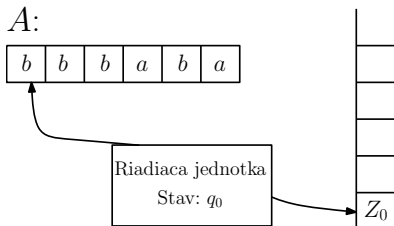
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



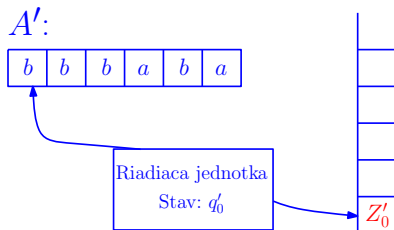
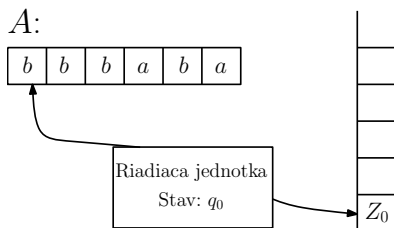
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



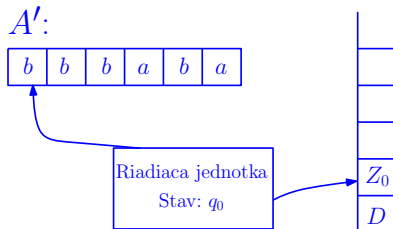
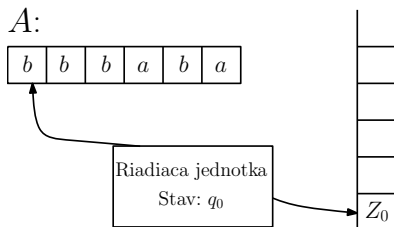
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

► Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



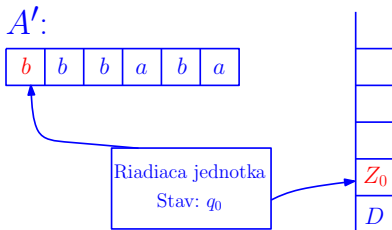
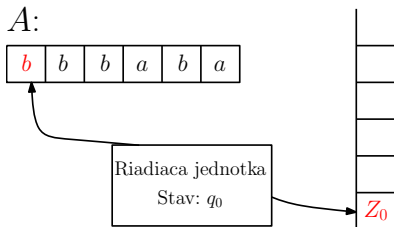
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



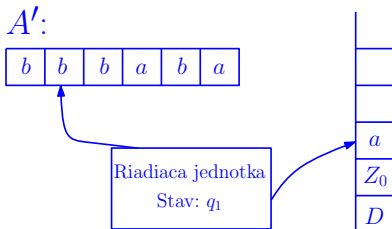
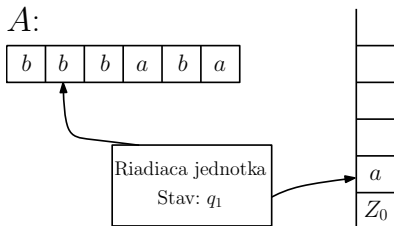
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



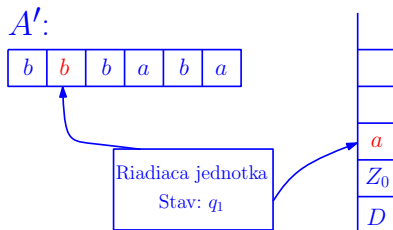
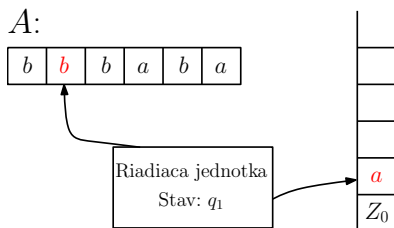
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



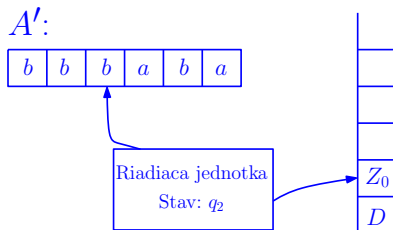
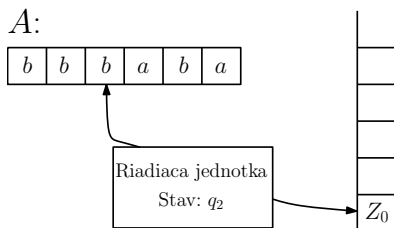
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



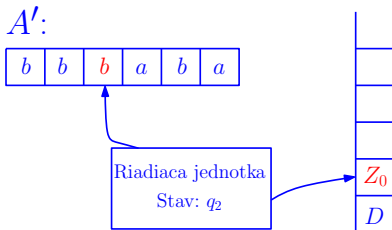
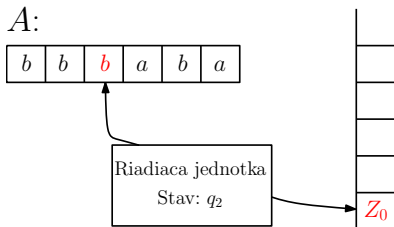
Riešenie problému 1

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



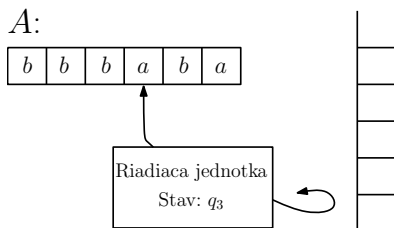
Riešenie problému 1

Lema

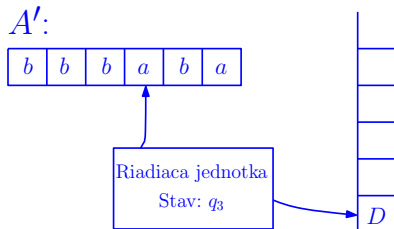
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



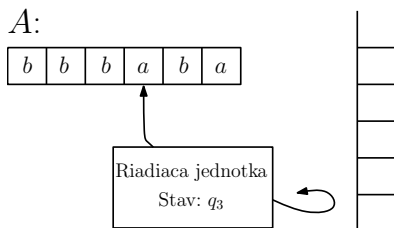
Riešenie problému 1

Lema

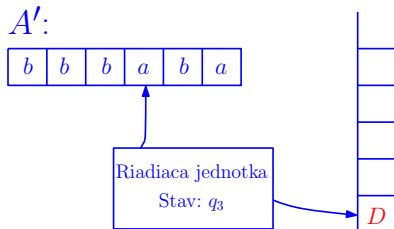
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



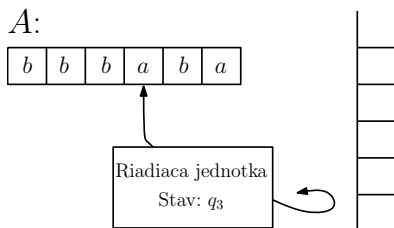
Riešenie problému 1

Lema

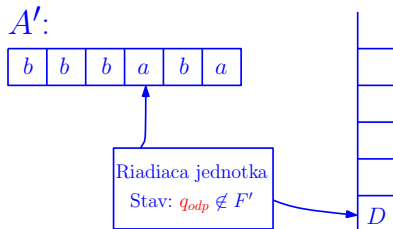
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



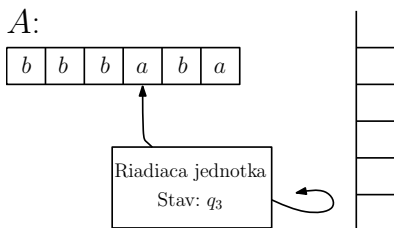
Riešenie problému 1

Lema

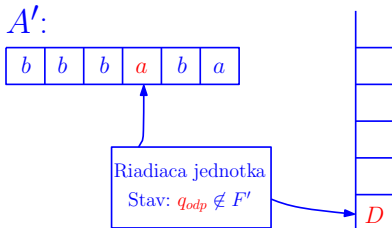
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



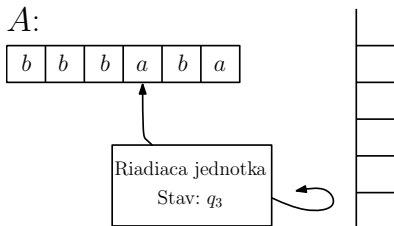
Riešenie problému 1

Lema

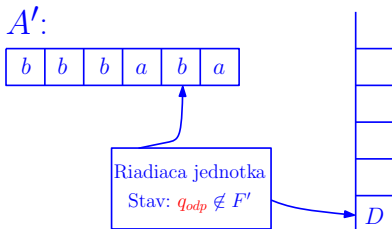
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



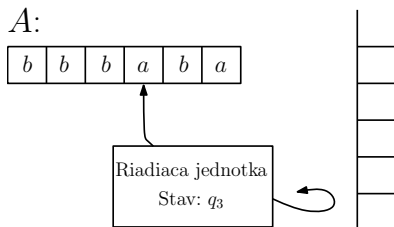
Riešenie problému 1

Lema

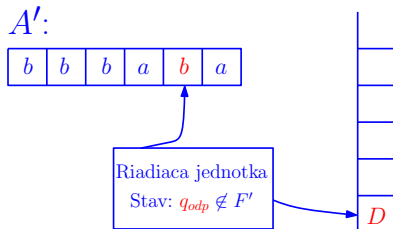
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



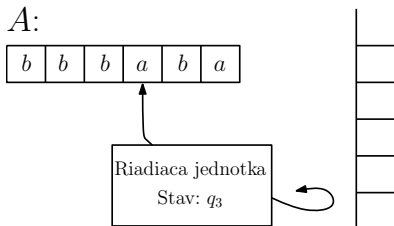
Riešenie problému 1

Lema

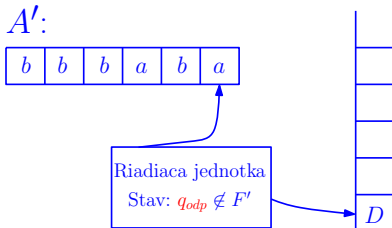
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



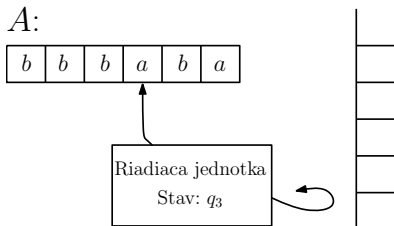
Riešenie problému 1

Lema

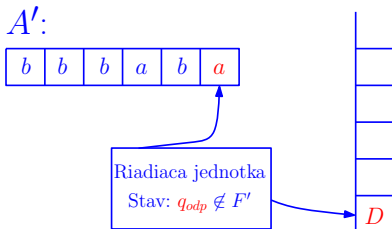
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



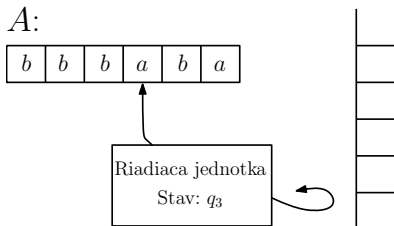
Riešenie problému 1

Lema

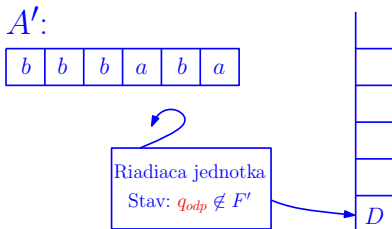
Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- Nový „odpadový stav“ q_{odp} a „falošné dno“ D

Situácia 2:



Náraz na dno!



Riešenie problému 1 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

Riešenie problému 1 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- ▶ $K' = K \cup \{q'_0, q_{odp}\}$, kde q'_0, q_{odp} sú nové stavy

Riešenie problému 1 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- ▶ $K' = K \cup \{q'_0, q_{odp}\}$, kde q'_0, q_{odp} sú nové stavy
- ▶ $\Gamma' = \Gamma \cup \{Z'_0, D\}$, kde Z'_0, D sú nové symboly

Riešenie problému 1 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- ▶ $K' = K \cup \{q'_0, q_{odp}\}$, kde q'_0, q_{odp} sú nové stavy
- ▶ $\Gamma' = \Gamma \cup \{Z'_0, D\}$, kde Z'_0, D sú nové symboly
- ▶ $F' = F$

Riešenie problému 1 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- ▶ $K' = K \cup \{q'_0, q_{odp}\}$, kde q'_0, q_{odp} sú nové stavy
- ▶ $\Gamma' = \Gamma \cup \{Z'_0, D\}$, kde Z'_0, D sú nové symboly
- ▶ $F' = F$

Pôvodný prechod	Nový prechod
	$\delta'(q'_0, \varepsilon, Z'_0) = (q_0, DZ_0)$
	$\delta'(q'_0, \varepsilon, Z) = (q_{odp}, Z) \quad \forall Z \in \Gamma \cup \{D\}$
	$\delta'(p, \varepsilon, Z'_0) = (q_{odp}, Z'_0) \quad \forall p \in K \cup \{q_{odp}\}$
$\delta(p, z, Z) = (q, \gamma)$	$\delta'(p, z, Z) = (q, \gamma)$
	$\delta'(p, a, Z) = (q_{odp}, Z)$
$\delta(p, a, Z) = \delta(p, \varepsilon, Z) = \emptyset$	$\delta'(p, \varepsilon, D) = (q_{odp}, D) \quad \forall p \in K$
	$\delta'(q_{odp}, c, Z) = (q_{odp}, Z) \quad \forall c \in \Sigma, Z \in \Gamma \cup \{D\}$

Riešenie problému 1 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje „nezasekávajúci sa“ DPDA A' taký, že $L(A') = L(A)$.

- ▶ $K' = K \cup \{q'_0, q_{odp}\}$, kde q'_0, q_{odp} sú nové stavy
- ▶ $\Gamma' = \Gamma \cup \{Z'_0, D\}$, kde Z'_0, D sú nové symboly
- ▶ $F' = F$

Pôvodný prechod	Nový prechod
	$\delta'(q'_0, \varepsilon, Z'_0) = (q_0, DZ_0)$
	$\delta'(q'_0, \varepsilon, Z) = (q_{odp}, Z) \quad \forall Z \in \Gamma \cup \{D\}$
	$\delta'(p, \varepsilon, Z'_0) = (q_{odp}, Z'_0) \quad \forall p \in K \cup \{q_{odp}\}$
$\delta(p, z, Z) = (q, \gamma)$	$\delta'(p, z, Z) = (q, \gamma)$
	$\delta'(p, a, Z) = (q_{odp}, Z)$
$\delta(p, a, Z) = \delta(p, \varepsilon, Z) = \emptyset$	$\delta'(p, \varepsilon, D) = (q_{odp}, D) \quad \forall p \in K$
	$\delta'(q_{odp}, c, Z) = (q_{odp}, Z) \quad \forall c \in \Sigma, Z \in \Gamma \cup \{D\}$

- ▶ Do q_{odp} možno predĺžiť aj niektoré akceptačné výpočty

Konstrukcia DPDA akceptujúceho komplement

Konštrukcia DPDA akceptujúceho komplement

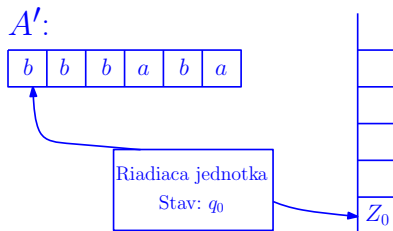
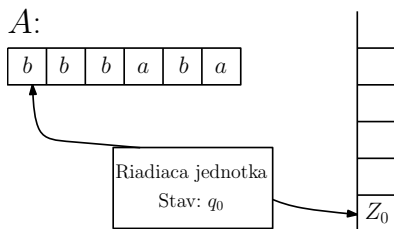
- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA

Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „*nezasekávajúceho sa*“ DPDA
- ▶ Problém 2: výpočet DPDA môže skončiť v „epsilonovom cykle“

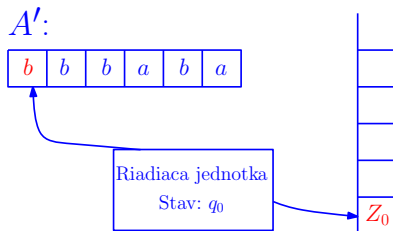
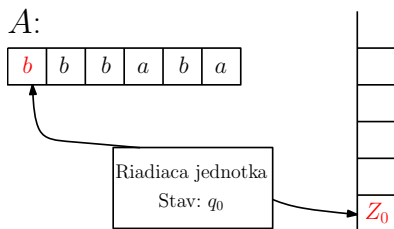
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ Problém 2: výpočet DPDA môže skončiť v „epsilonovom cykle“



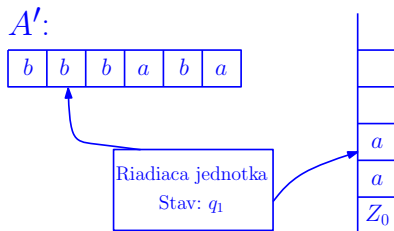
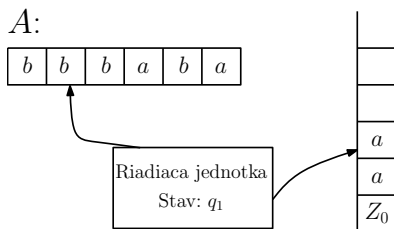
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ Problém 2: výpočet DPDA môže skončiť v „epsilonovom cykle“



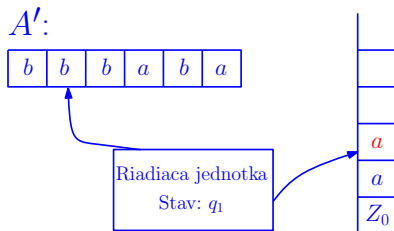
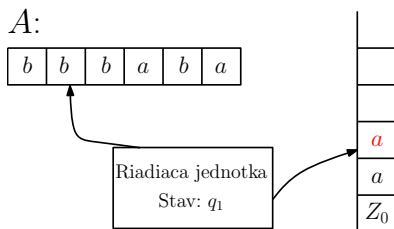
Konštrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 2:** výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ **Problém 2:** výpočet DPDA môže skončiť v „epsilonovom cykle“



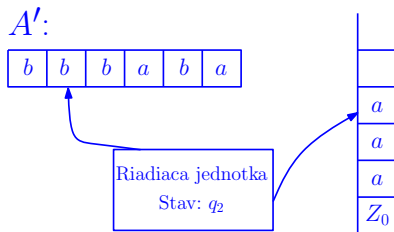
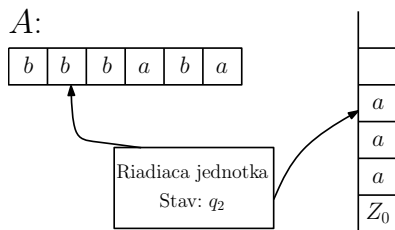
Konštrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 2:** výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ **Problém 2:** výpočet DPDA môže skončiť v „epsilonovom cykle“



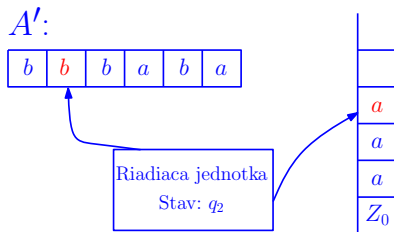
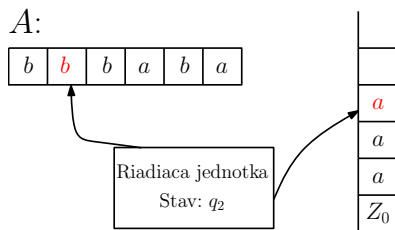
Konštrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 2:** výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ **Problém 2:** výpočet DPDA môže skončiť v „epsilonovom cykle“



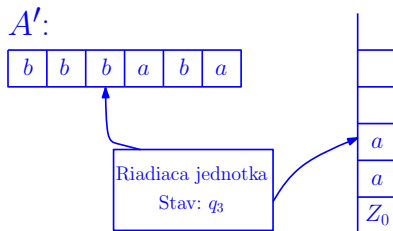
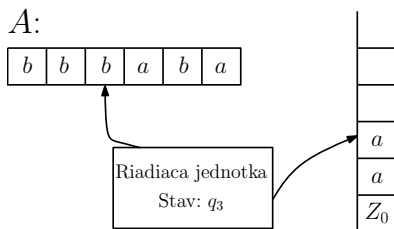
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ Problém 2: výpočet DPDA môže skončiť v „epsilonovom cykle“



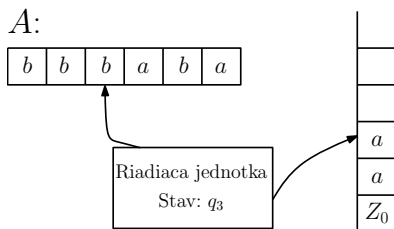
Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ Problém 2: výpočet DPDA môže skončiť v „epsilonovom cykle“

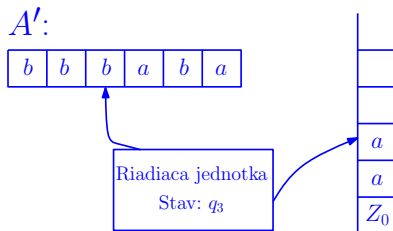


Konštrukcia DPDA akceptujúceho komplement

- ▶ Naivný prístup 2: výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ Problém 2: výpočet DPDA môže skončiť v „epsilonovom cykle“

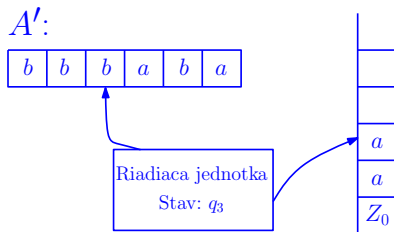
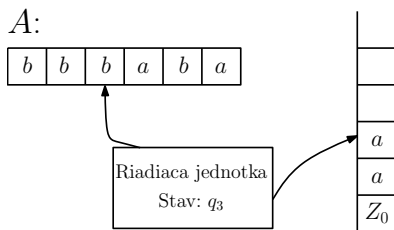


Už len prechody na ε



Konštrukcia DPDA akceptujúceho komplement

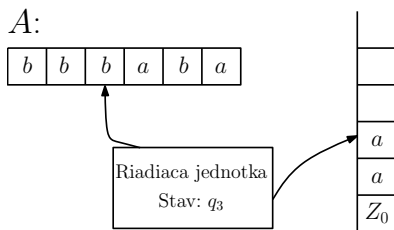
- ▶ **Naivný prístup 2:** výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ **Problém 2:** výpočet DPDA môže skončiť v „epsilonovom cykle“



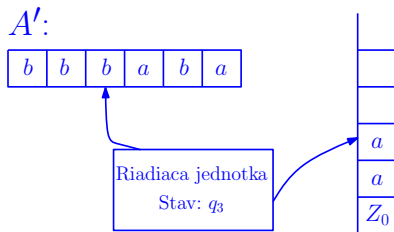
Už len prechody na ε
 $bbbaba \notin L(A)$

Konštrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 2:** výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ DPDA
- ▶ **Problém 2:** výpočet DPDA môže skončiť v „epsilonovom cykle“



Už len prechody na ε
 $bbbaba \notin L(A)$



Už len prechody na ε
 $bbbaba \notin L(A')$

Riešenie problému 2

Riešenie problému 2

- ▶ Ukážeme, že „epsilonové cykly“ možno detegovať

Riešenie problému 2

- ▶ Ukážeme, že „epsilonové cykly“ možno detegovať
- ▶ Zostrojíme automat, ktorý po detekcii „epsilonového cyklu“ prejde do „odpadového stavu“, v ktorom dočíta vstup

Riešenie problému 2

- ▶ Ukážeme, že „epsilonové cykly“ možno detegovať
- ▶ Zostrojíme automat, ktorý po detekcii „epsilonového cyklu“ prejde do „odpadového stavu“, v ktorom dočíta vstup
- ▶ „Epsilonové cykly“ sú dvoch druhov:

Riešenie problému 2

- ▶ Ukážeme, že „epsilonové cykly“ možno detegovať
- ▶ Zostrojíme automat, ktorý po detekcii „epsilonového cyklu“ prejde do „odpadového stavu“, v ktorom dočíta vstup
- ▶ „Epsilonové cykly“ sú dvoch druhov:
 1. Buď zásobník rastie nad všetky medze

Riešenie problému 2

- ▶ Ukážeme, že „epsilonové cykly“ možno detegovať
- ▶ Zostrojíme automat, ktorý po detekcii „epsilonového cyklu“ prejde do „odpadového stavu“, v ktorom dočíta vstup
- ▶ „Epsilonové cykly“ sú dvoch druhov:
 1. Buď zásobník rastie nad všetky medze
 2. Alebo sa obsah zásobníka začne opakovať

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Označenie

Ak je počas celého výpočtu $(p, uv, \gamma X) \vdash^ (q, v, \gamma \beta Y)$ viac ako $|\gamma|$ symbolov na zásobníku, píšeme $(p, uv, \gamma X) \vdash_{>|\gamma|}^* (q, v, \gamma \beta Y)$.*

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Označenie

Ak je počas celého výpočtu $(p, uv, \gamma X) \vdash^* (q, v, \gamma \beta Y)$ viac ako $|\gamma|$ symbolov na zásobníku, píšeme $(p, uv, \gamma X) \vdash_{>|\gamma|}^* (q, v, \gamma \beta Y)$.

- Podvýpočet $(q, \varepsilon w, \gamma Z) \vdash_{>|\gamma|}^+ (q, w, \gamma \beta Z) \rightsquigarrow$ „ ε -cyklus“

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Označenie

Ak je počas celého výpočtu $(p, uv, \gamma X) \vdash^* (q, v, \gamma \beta Y)$ viac ako $|\gamma|$ symbolov na zásobníku, píšeme $(p, uv, \gamma X) \vdash_{>|\gamma|}^* (q, v, \gamma \beta Y)$.

- ▶ Podvýpočet $(q, \varepsilon w, \gamma Z) \vdash_{>|\gamma|}^+ (q, w, \gamma \beta Z) \rightsquigarrow$ „ ε -cyklus“
- ▶ $(q, w, \gamma Z) \vdash^+ (q, w, \gamma \beta Z) \vdash^+ (q, w, \gamma \beta \beta Z) \vdash^+ \dots$

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Označenie

Ak je počas celého výpočtu $(p, uv, \gamma X) \vdash^* (q, v, \gamma \beta Y)$ viac ako $|\gamma|$ symbolov na zásobníku, píšeme $(p, uv, \gamma X) \vdash_{>|\gamma|}^* (q, v, \gamma \beta Y)$.

- ▶ Podvýpočet $(q, \varepsilon w, \gamma Z) \vdash_{>|\gamma|}^+ (q, w, \gamma \beta Z) \rightsquigarrow$ „ ε -cyklus“
- ▶ $(q, w, \gamma Z) \vdash^+ (q, w, \gamma \beta Z) \vdash^+ (q, w, \gamma \beta \beta Z) \vdash^+ \dots$
- ▶ **Ukážeme:** ak vo výpočte na ε zásobník dostatočne „narastie“, musí v ňom existovať podvýpočet takéhoto typu

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Označenie

Ak je počas celého výpočtu $(p, uv, \gamma X) \vdash^* (q, v, \gamma \beta Y)$ viac ako $|\gamma|$ symbolov na zásobníku, píšeme $(p, uv, \gamma X) \vdash_{>|\gamma|}^* (q, v, \gamma \beta Y)$.

- ▶ Podvýpočet $(q, \varepsilon w, \gamma Z) \vdash_{>|\gamma|}^+ (q, w, \gamma \beta Z) \rightsquigarrow$ „ ε -cyklus“
- ▶ $(q, w, \gamma Z) \vdash^+ (q, w, \gamma \beta Z) \vdash^+ (q, w, \gamma \beta \beta Z) \vdash^+ \dots$
- ▶ **Ukážeme:** ak vo výpočte na ε zásobník dostatočne „narastie“, musí v ňom existovať podvýpočet takéhoto typu
- ▶ Na detekciu týchto cyklov stačí sledovať „nárast“ zásobníka

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$

Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

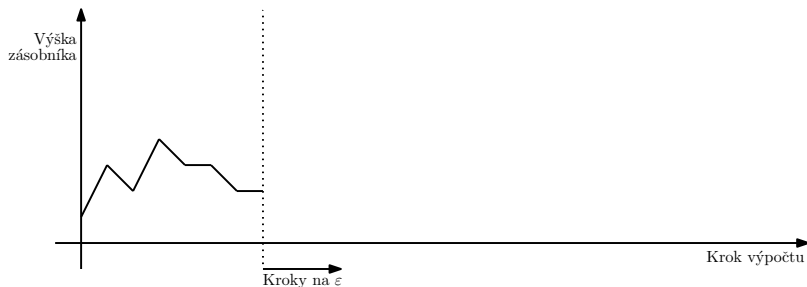
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

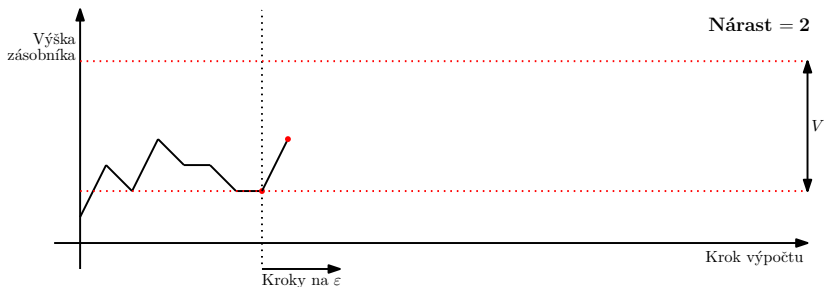
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

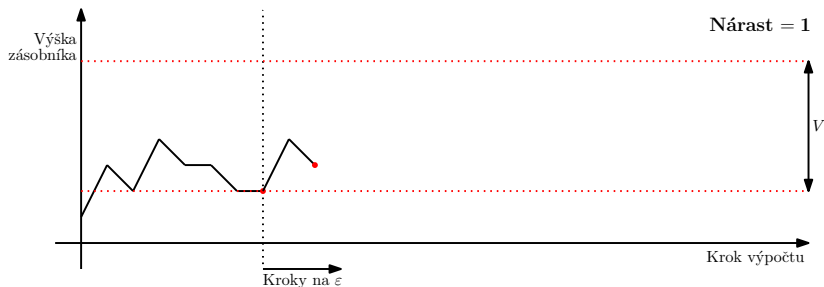
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

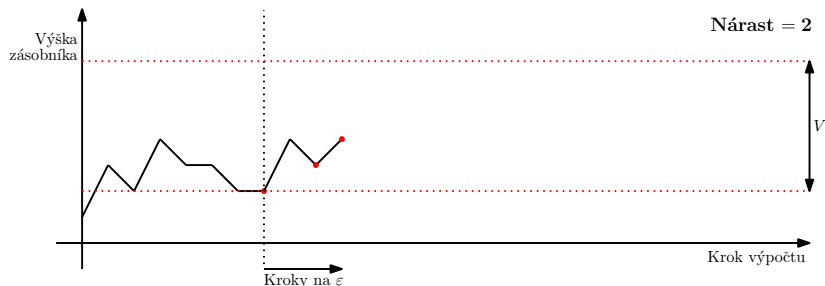
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

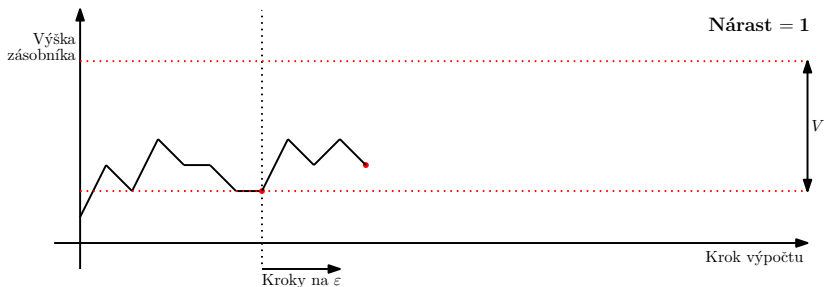
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

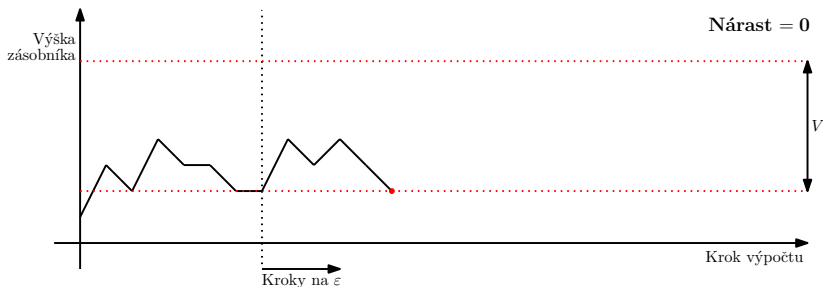
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

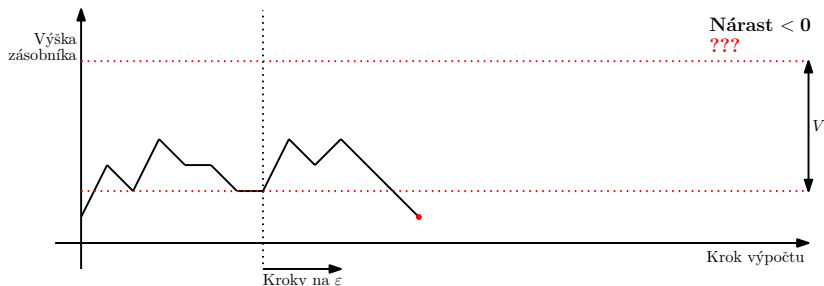
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

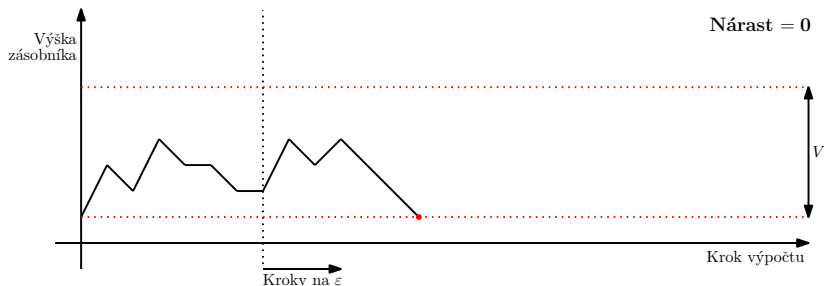
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

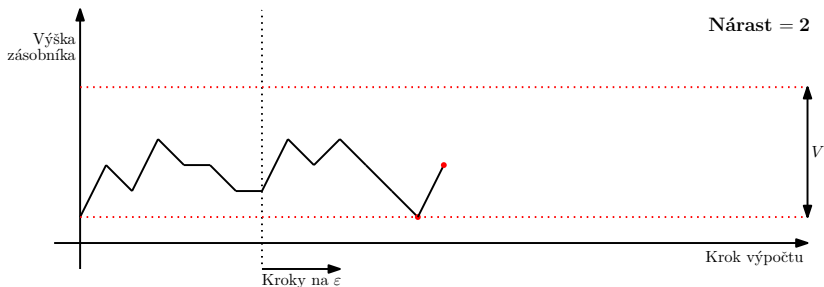
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

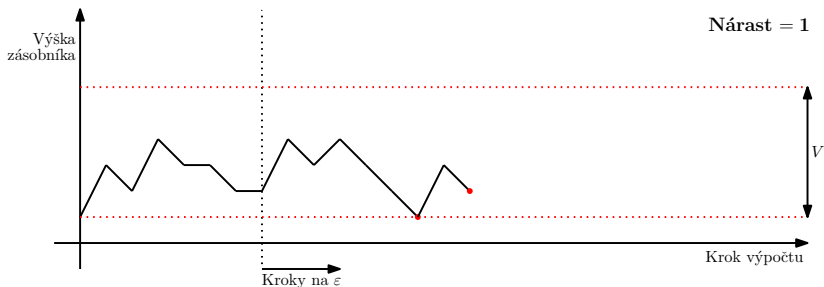
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

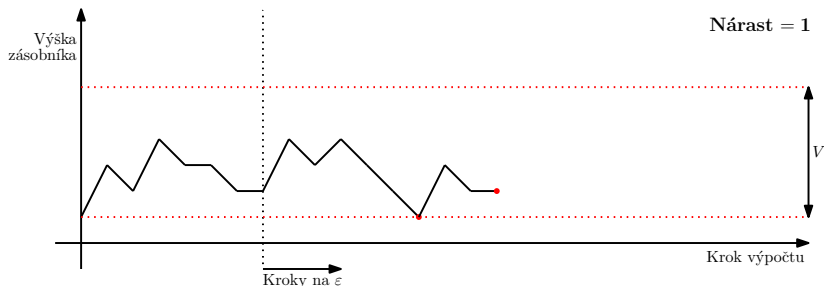
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

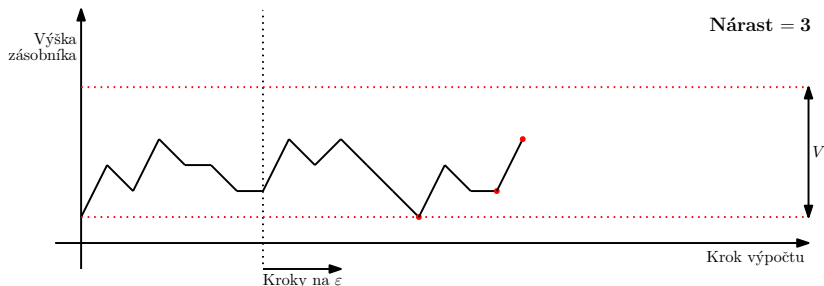
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

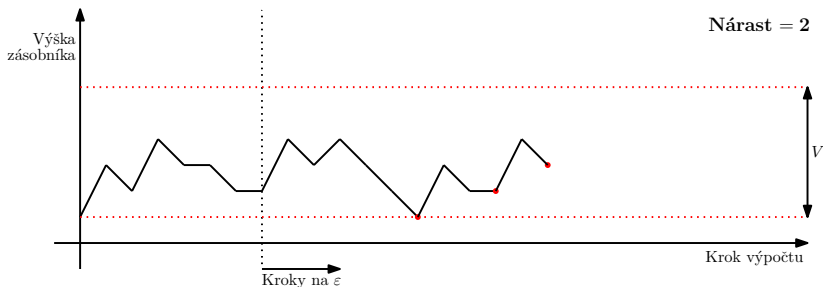
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

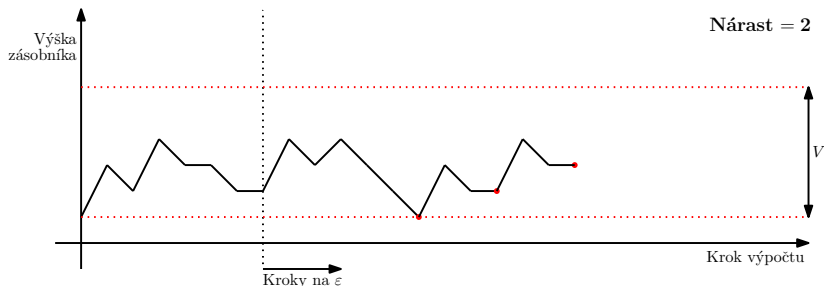
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

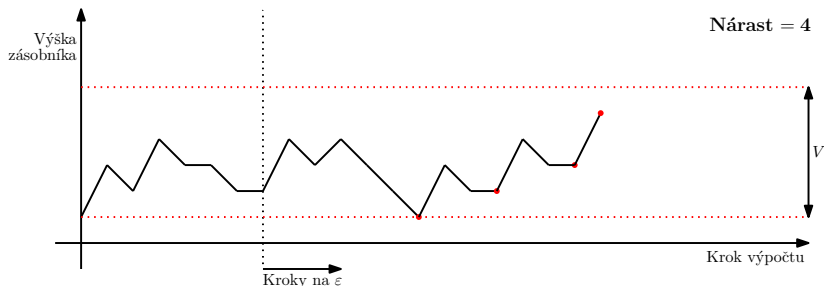
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

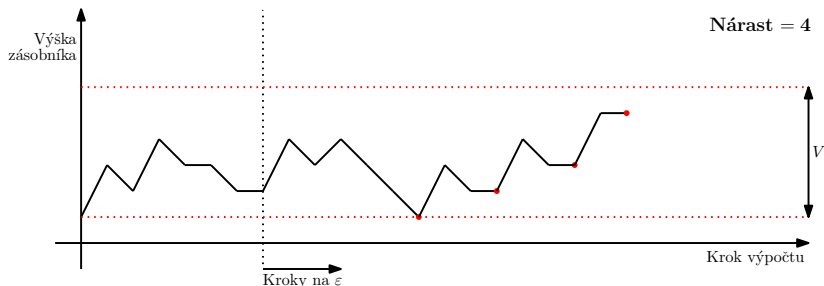
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

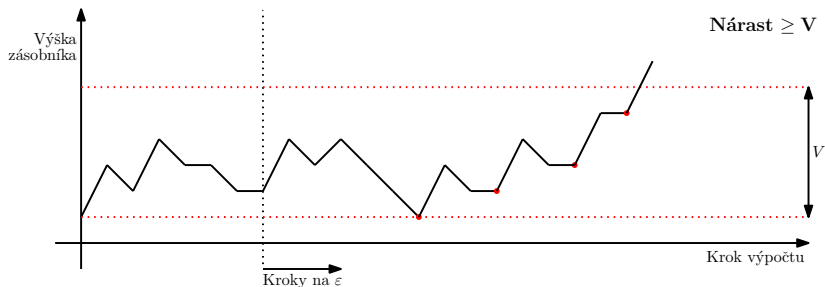
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

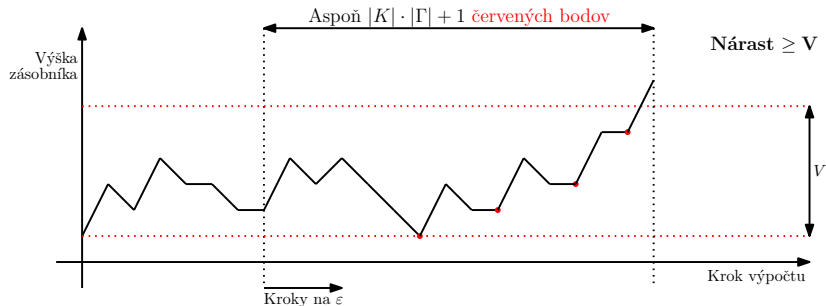
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

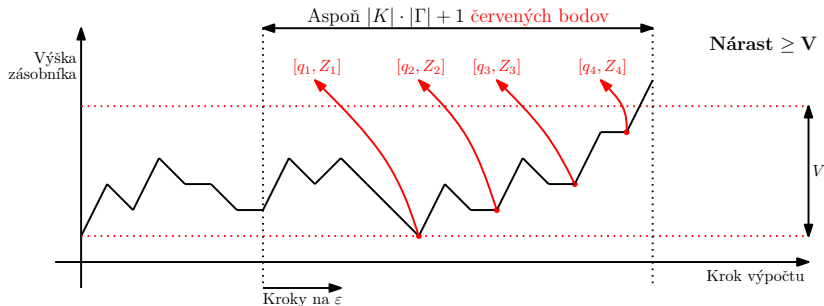
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

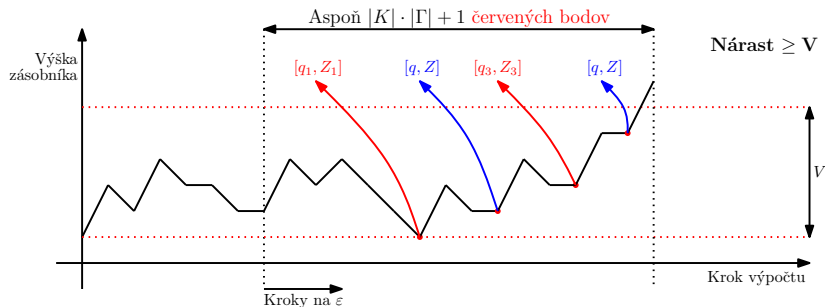
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

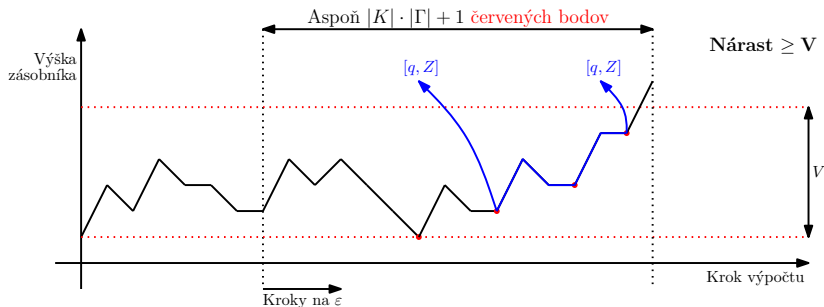
$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s neobmedzene rastúcim zásobníkom

$$k := \max\{|\beta| \mid \beta \in \Gamma^*; \exists p, q \in K \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \beta)\}$$

$$V := (k - 1) \cdot (|K| \cdot |\Gamma| + 1)$$



Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa
 $s \leq h < s + V$ pre nejaké s

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

- ▶ Pamätať si súčasne nárast a počet krokov

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

- ▶ Pamätať si súčasne nárast a počet krokov
- ▶ Ak nárast = 0 a zníži sa zásobník: vynulovať počítadlo krokov

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

- ▶ Pamätať si súčasne nárast a počet krokov
- ▶ Ak nárast = 0 a zníži sa zásobník: vynulovať počítadlo krokov
- ▶ Pri kroku na písmeno vynulovať nárast aj počítadlo krokov

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

- ▶ Pamätať si súčasne nárast a počet krokov
- ▶ Ak nárast = 0 a zníži sa zásobník: vynulovať počítadlo krokov
- ▶ Pri kroku na písmeno vynulovať nárast aj počítadlo krokov
- ▶ Nárast $\geq V$: „ ε -cyklus“ s neobmedzene rastúcim zásobníkom

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

- ▶ Pamätať si súčasne nárast a počet krokov
- ▶ Ak nárast = 0 a zníži sa zásobník: vynulovať počítadlo krokov
- ▶ Pri kroku na písmeno vynulovať nárast aj počítadlo krokov
- ▶ Nárast $\geq V$: „ ε -cyklus“ s neobmedzene rastúcim zásobníkom
- ▶ $\geq C$ krokov: „ ε -cyklus“ s opakujúcim sa obsahom zásobníka

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka

Výpočet je v takomto „ ε -cykle“ kedykoľvek:

- ▶ Od nejakej konfigurácie výška zásobníka h spĺňa $s \leq h < s + V$ pre nejaké s
- ▶ Táto situácia trvá aspoň $C := |K| \cdot (|\Gamma| + 1)^V + 1$ krokov na ε

Detekcia „ ε -cyklov“ s opakujúcim sa obsahom zásobníka:

- ▶ Pamätať si súčasne nárast a počet krokov
- ▶ Ak nárast = 0 a zníži sa zásobník: vynulovať počítadlo krokov
- ▶ Pri kroku na písmeno vynulovať nárast aj počítadlo krokov
- ▶ Nárast $\geq V$: „ ε -cyklus“ s neobmedzene rastúcim zásobníkom
- ▶ $\geq C$ krokov: „ ε -cyklus“ s opakujúcim sa obsahom zásobníka

Takto možno detegovať oba druhy „ ε -cyklov“ súčasne

Riešenie problému 2

Riešenie problému 2



Riešenie problému 2



Riešenie problému 2



Riešenie problému 2



Riešenie problému 2



Riešenie problému 2



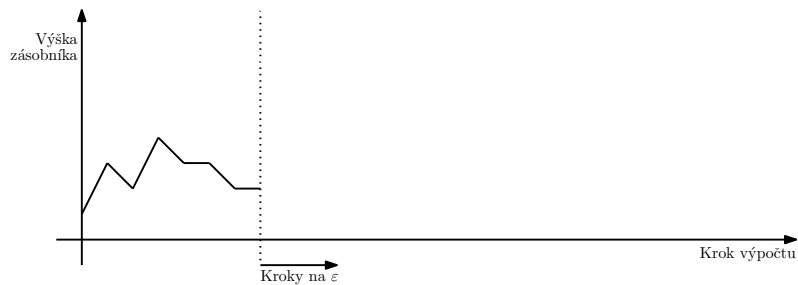
Riešenie problému 2



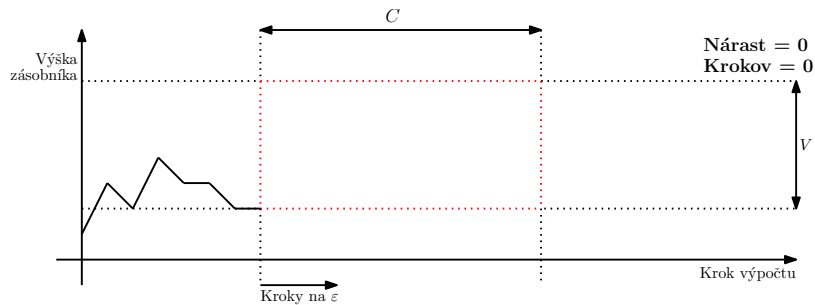
Riešenie problému 2



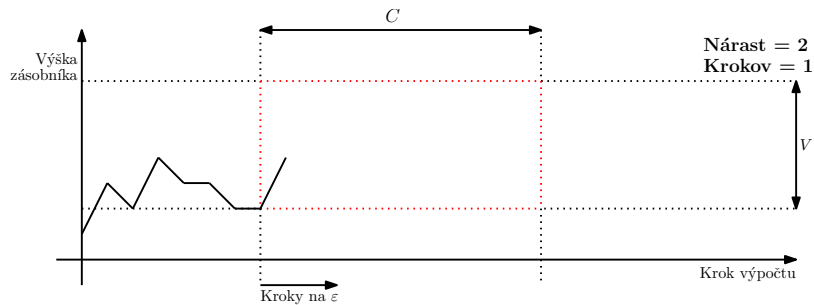
Riešenie problému 2



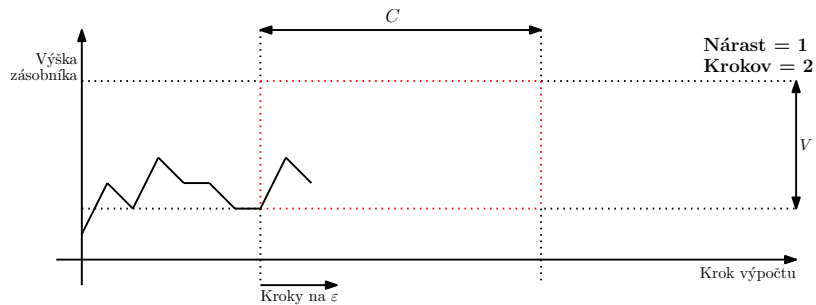
Riešenie problému 2



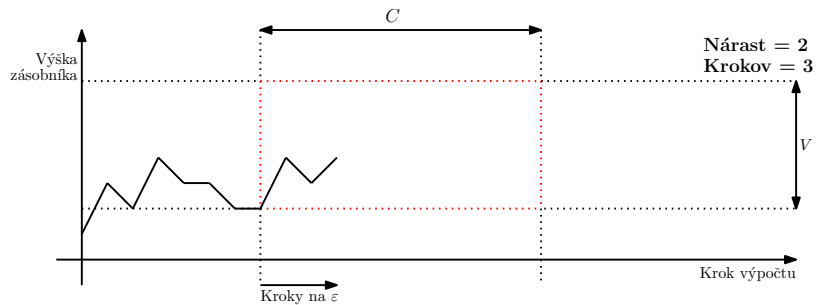
Riešenie problému 2



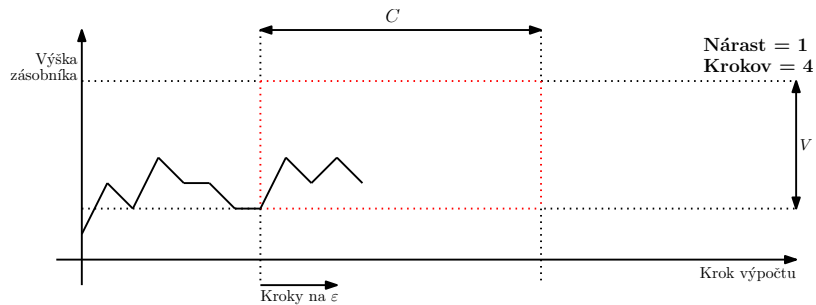
Riešenie problému 2



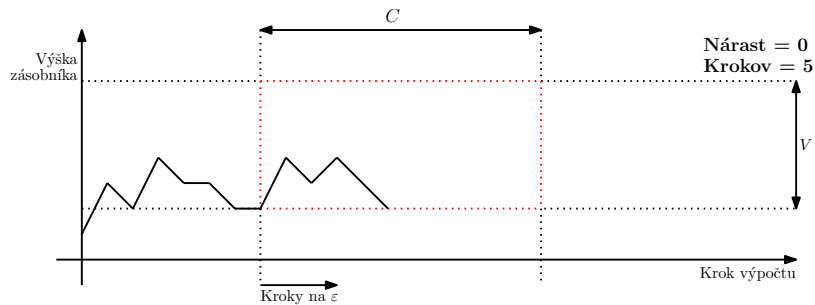
Riešenie problému 2



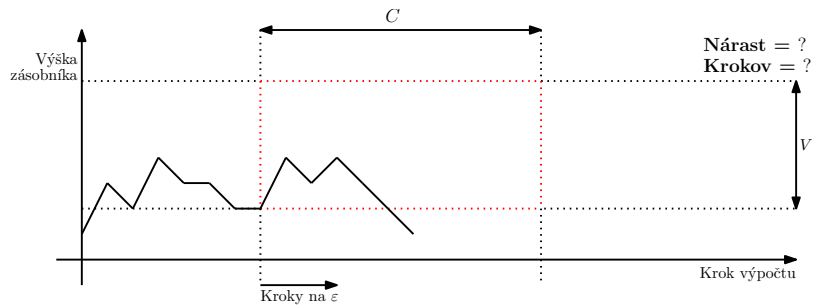
Riešenie problému 2



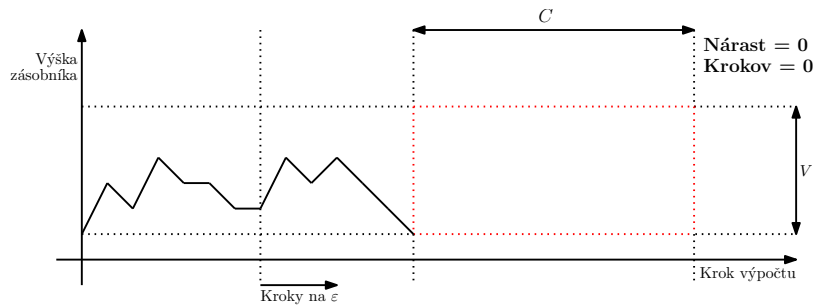
Riešenie problému 2



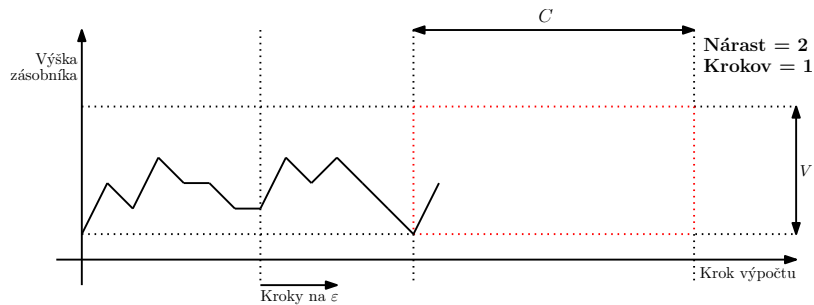
Riešenie problému 2



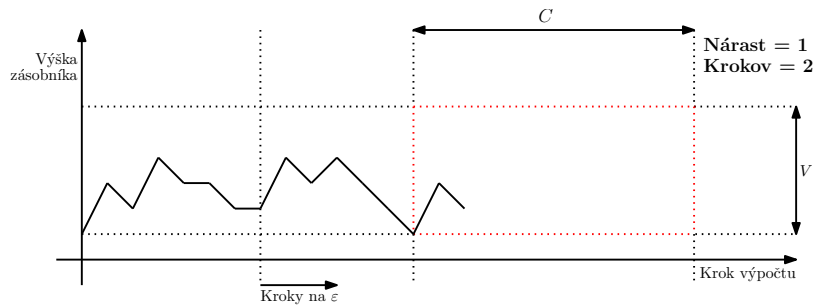
Riešenie problému 2



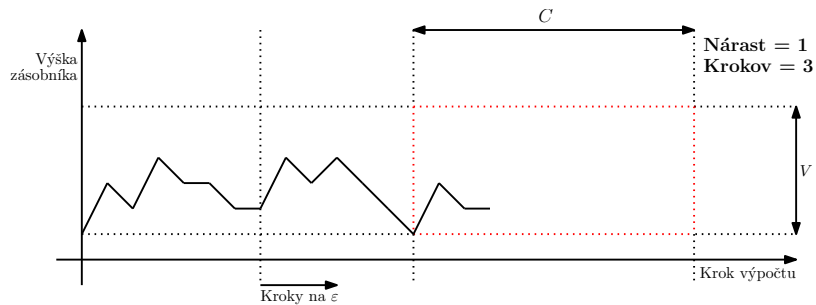
Riešenie problému 2



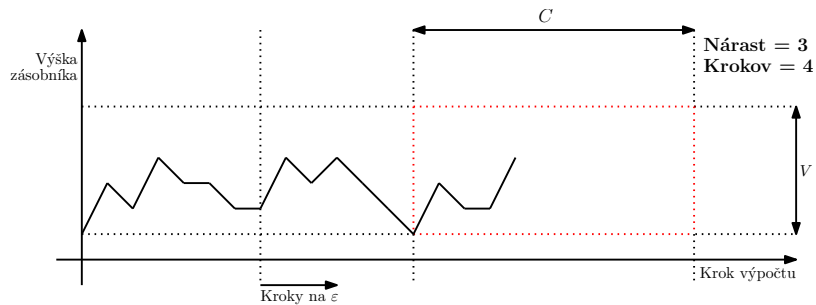
Riešenie problému 2



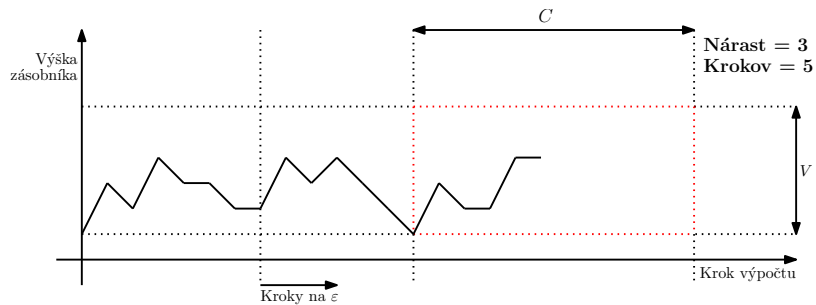
Riešenie problému 2



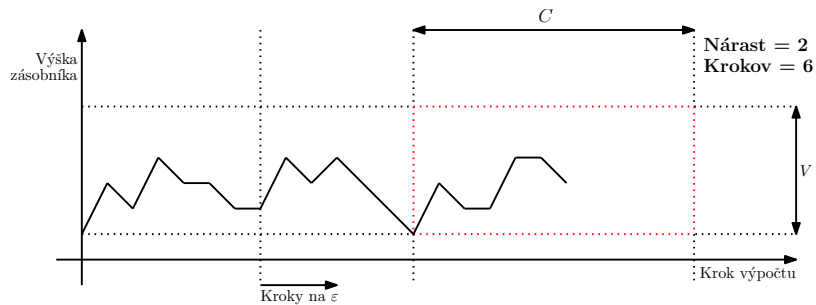
Riešenie problému 2



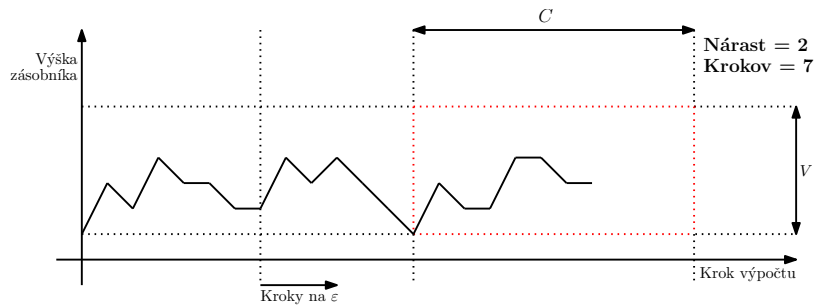
Riešenie problému 2



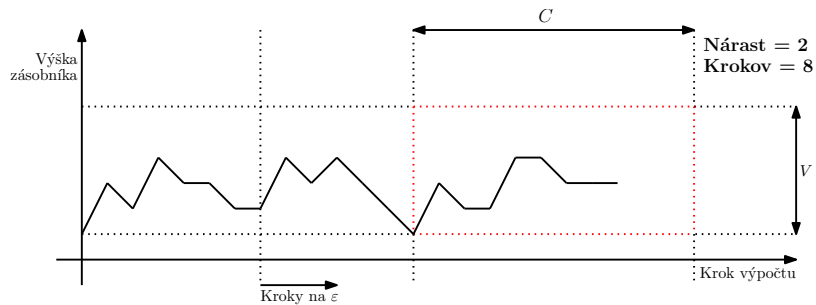
Riešenie problému 2



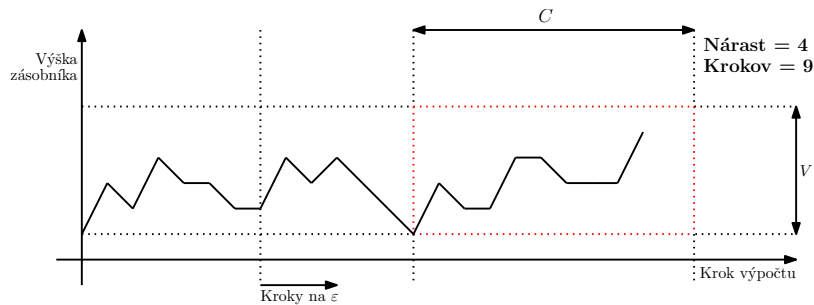
Riešenie problému 2



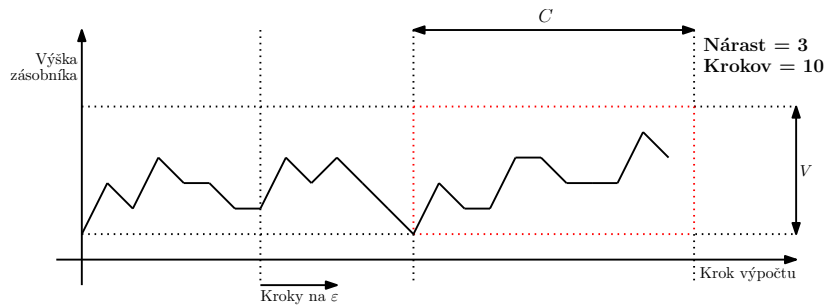
Riešenie problému 2



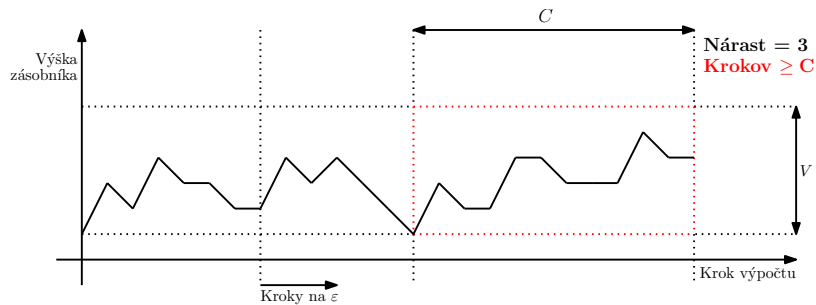
Riešenie problému 2



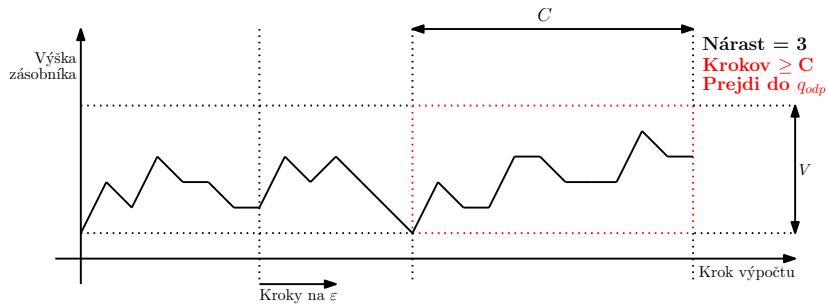
Riešenie problému 2



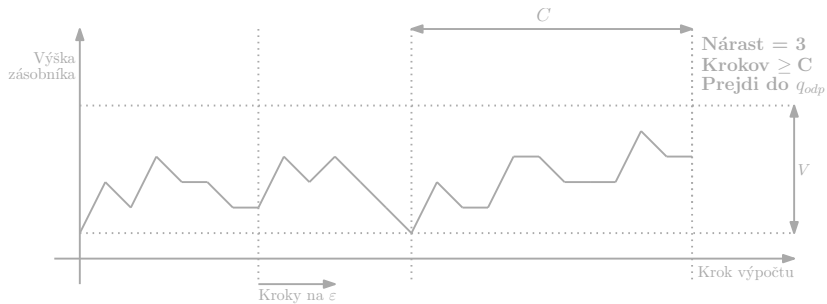
Riešenie problému 2



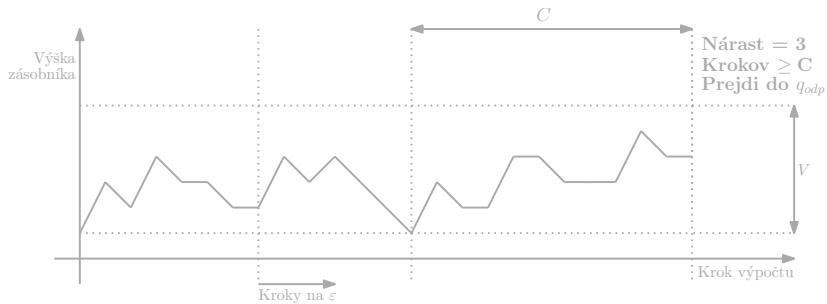
Riešenie problému 2



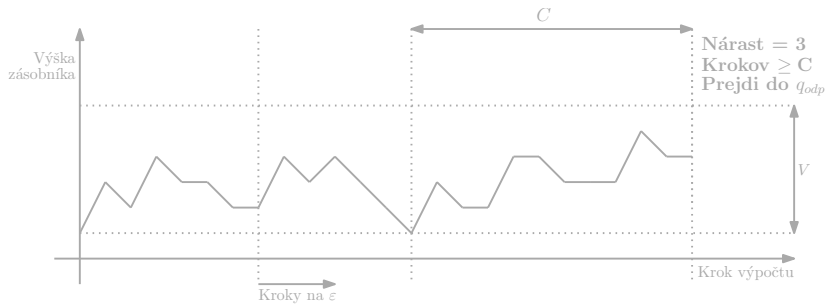
Riešenie problému 2



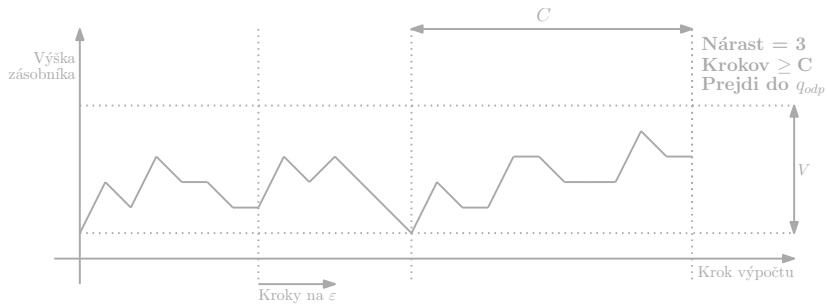
Riešenie problému 2



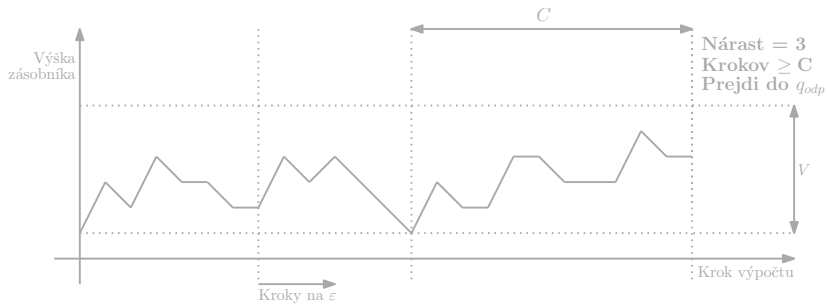
Riešenie problému 2



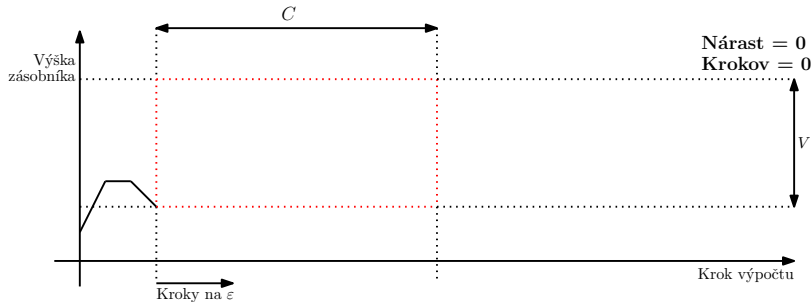
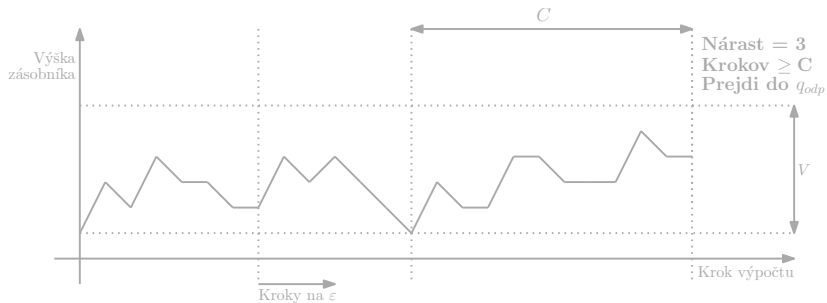
Riešenie problému 2



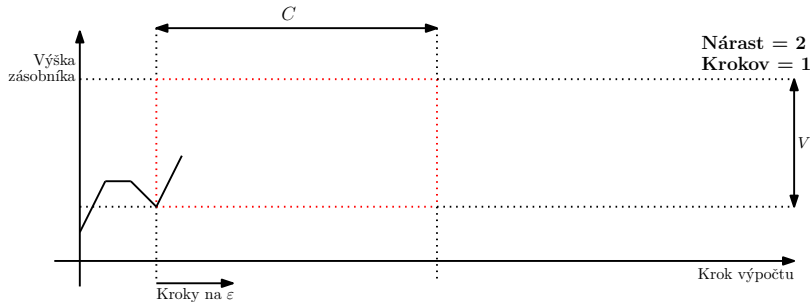
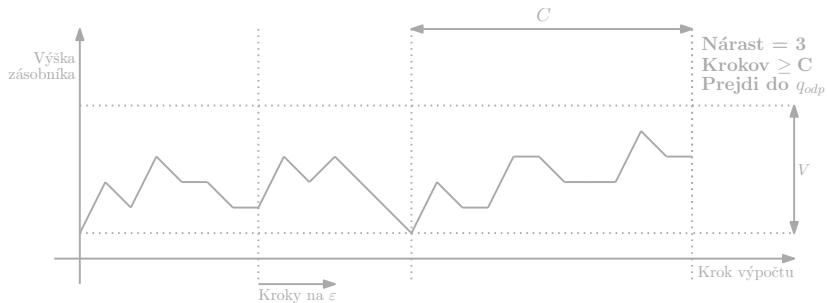
Riešenie problému 2



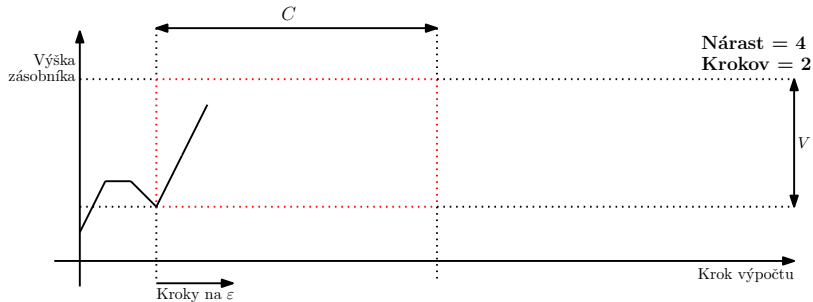
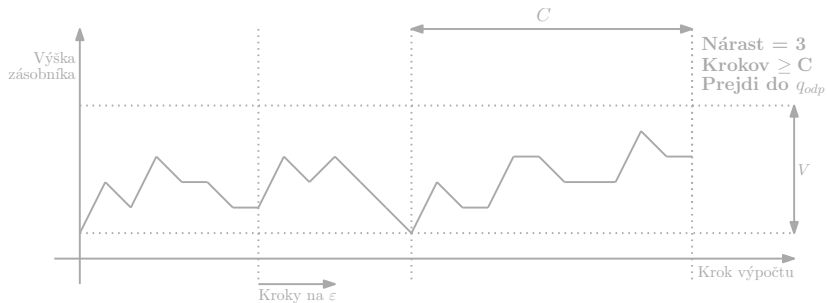
Riešenie problému 2



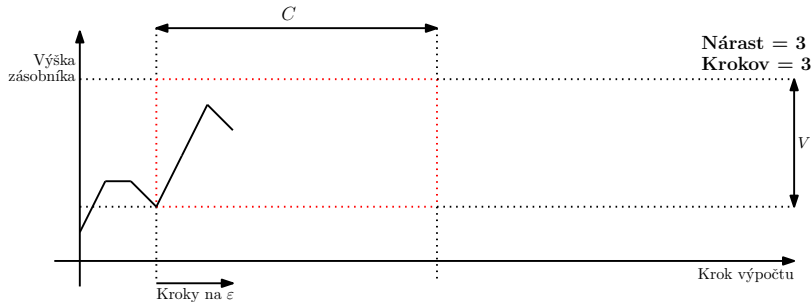
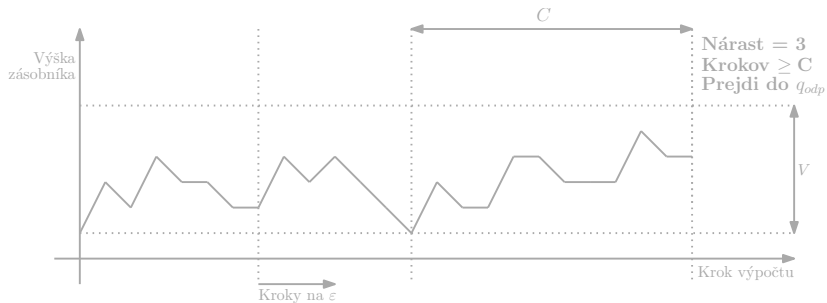
Riešenie problému 2



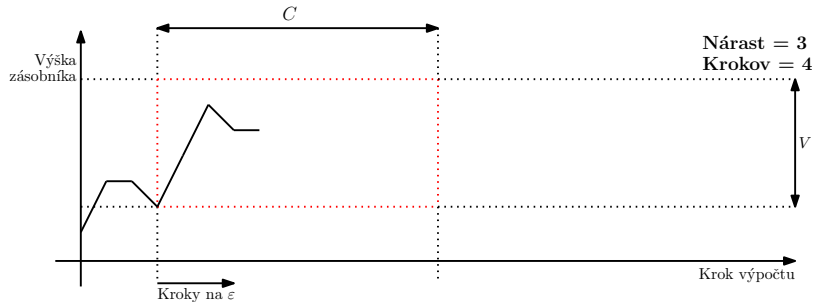
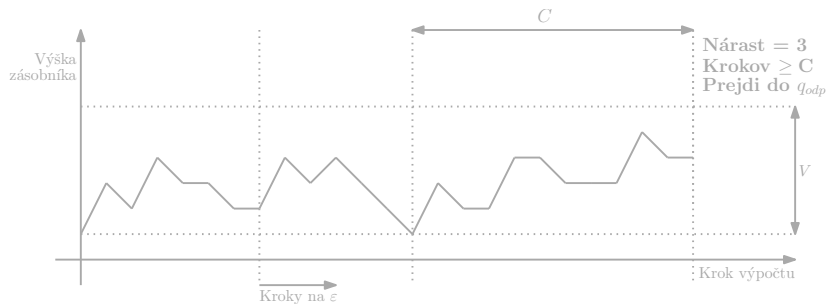
Riešenie problému 2



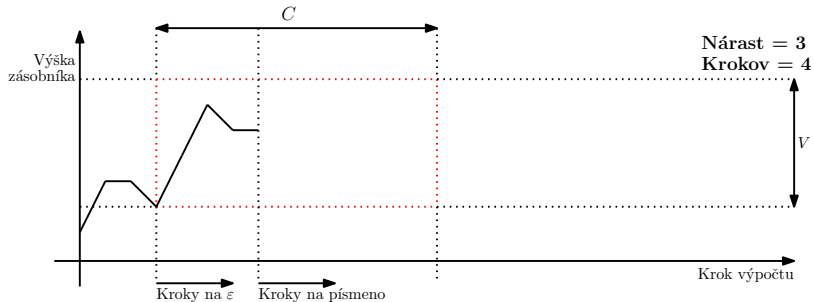
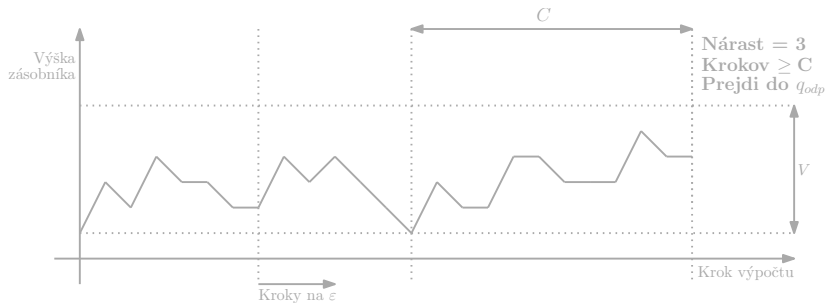
Riešenie problému 2



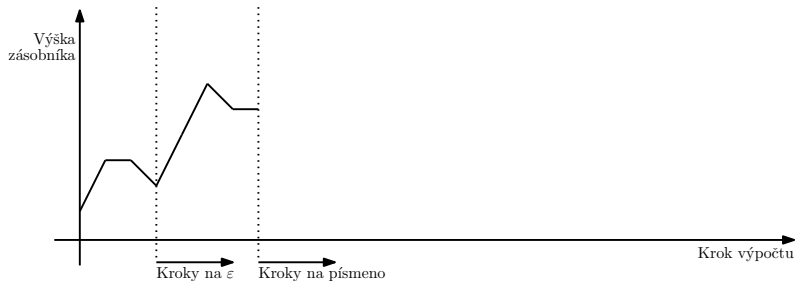
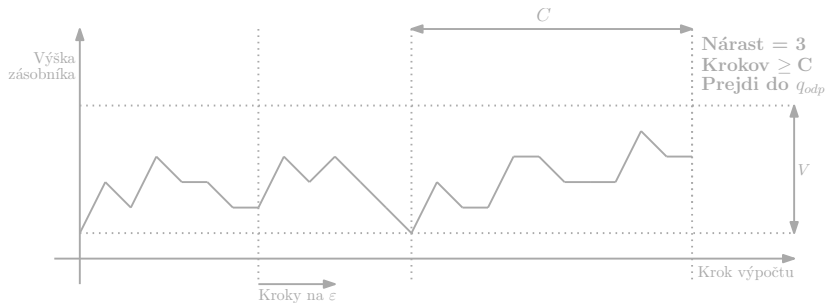
Riešenie problému 2



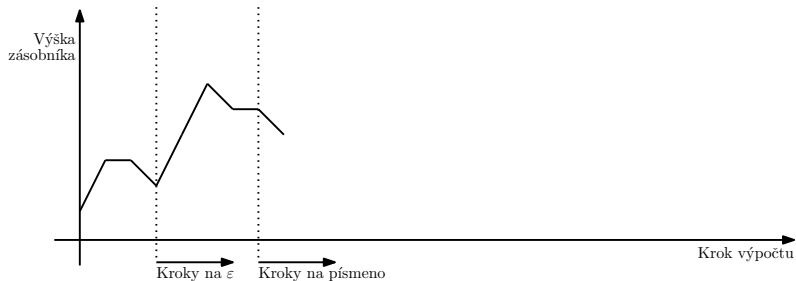
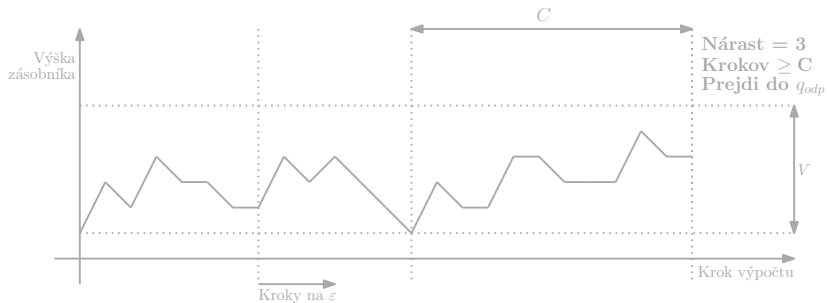
Riešenie problému 2



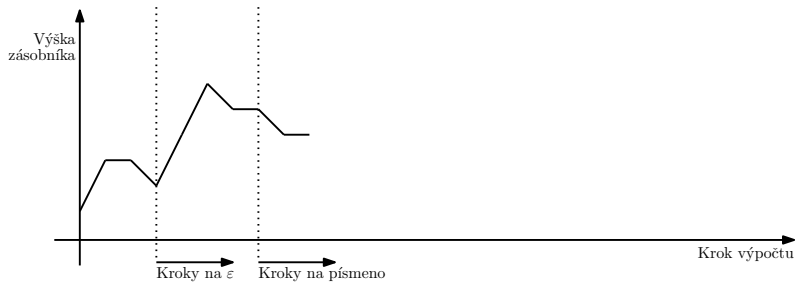
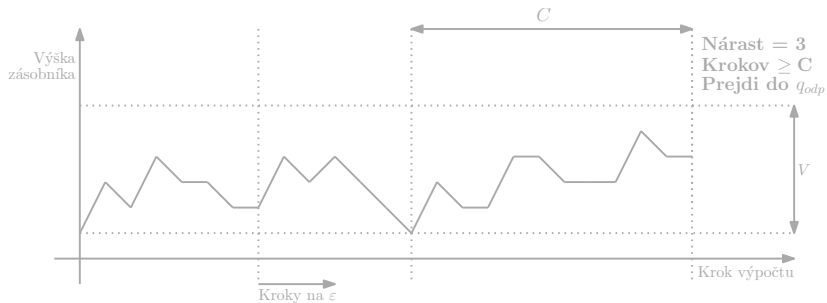
Riešenie problému 2



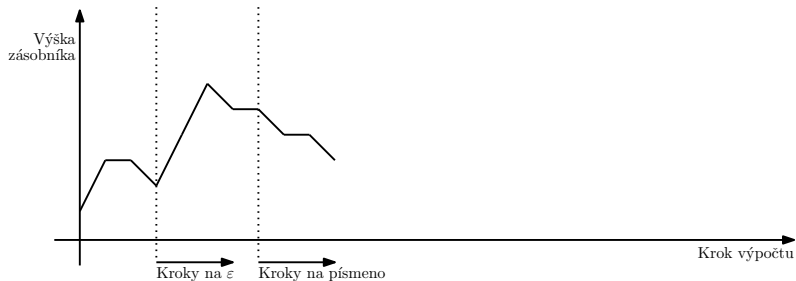
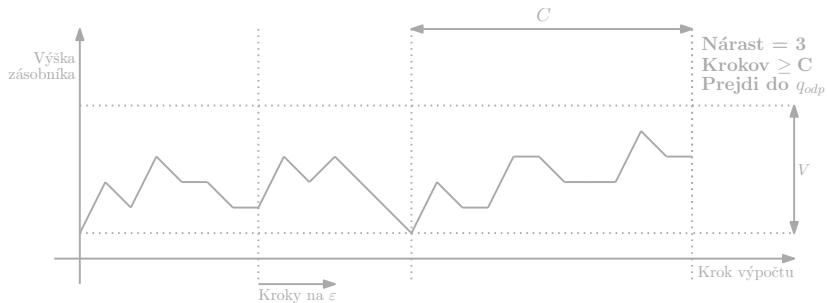
Riešenie problému 2



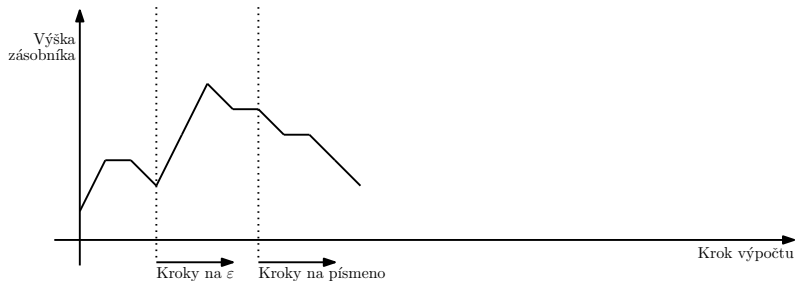
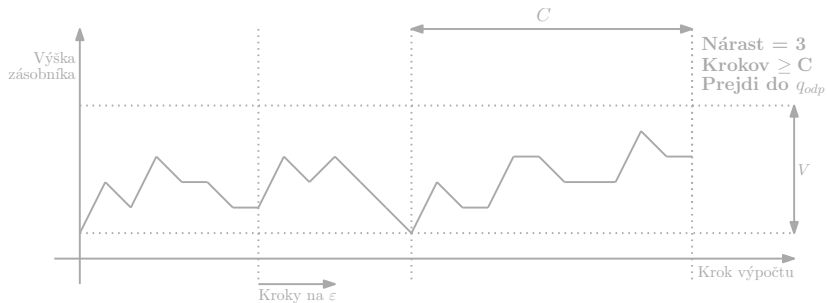
Riešenie problému 2



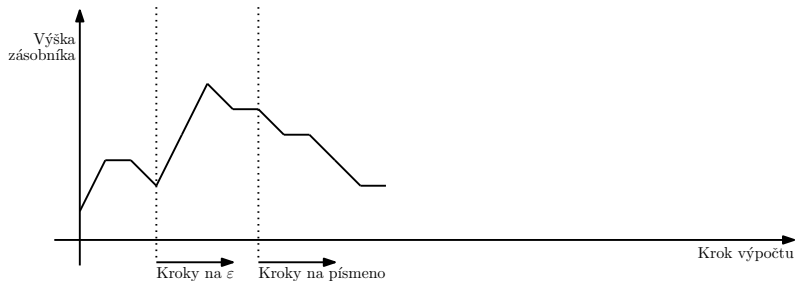
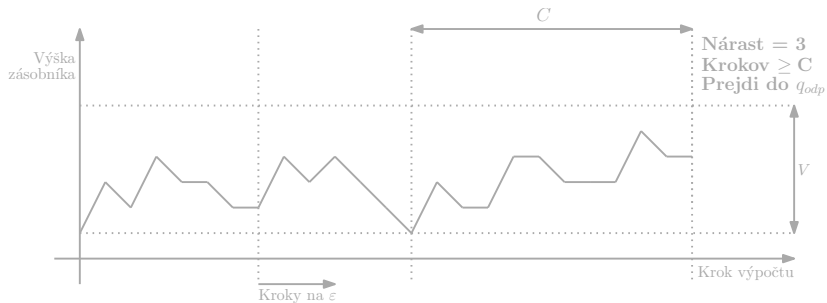
Riešenie problému 2



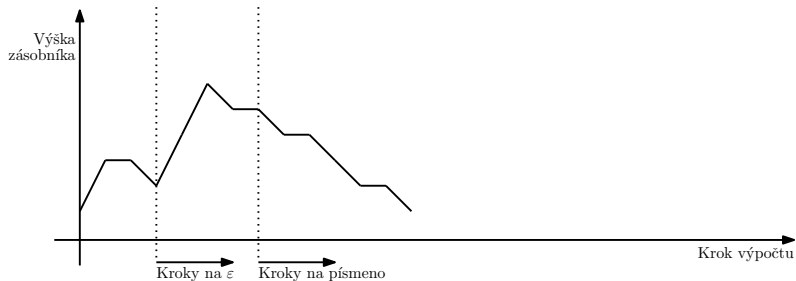
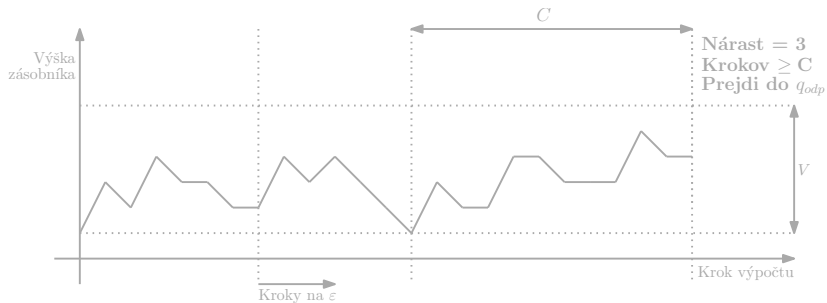
Riešenie problému 2



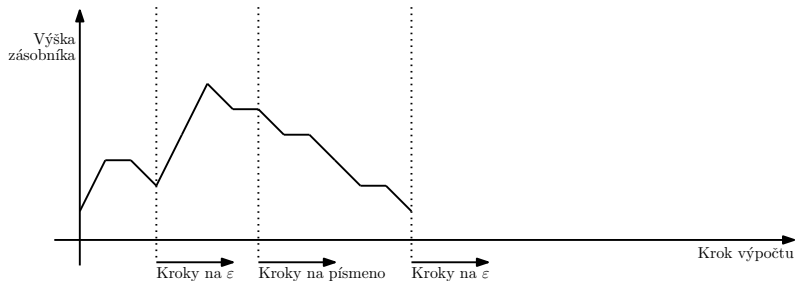
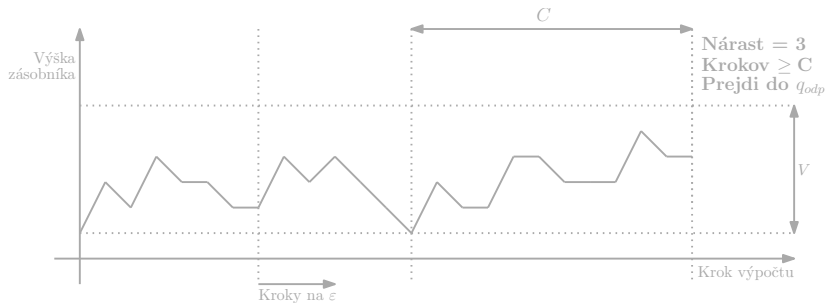
Riešenie problému 2



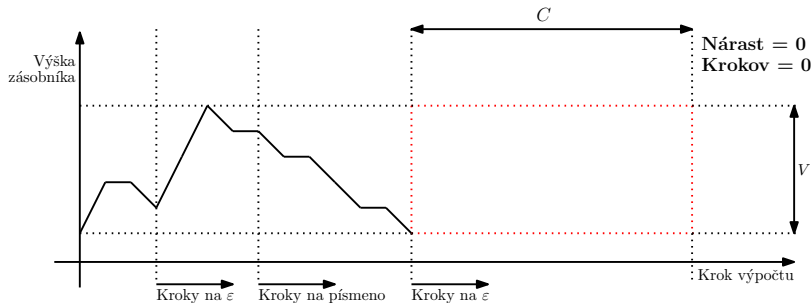
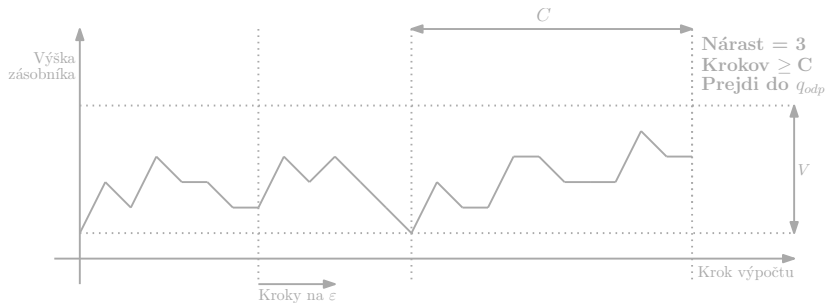
Riešenie problému 2



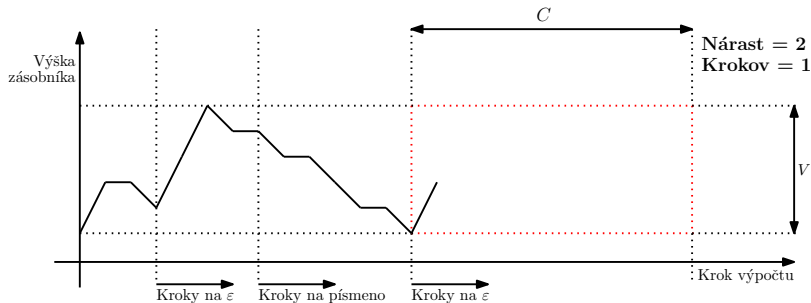
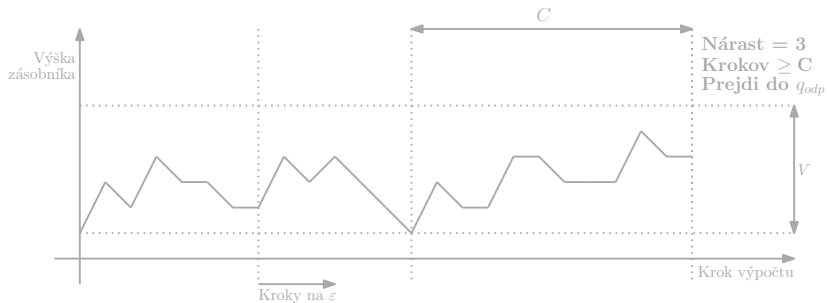
Riešenie problému 2



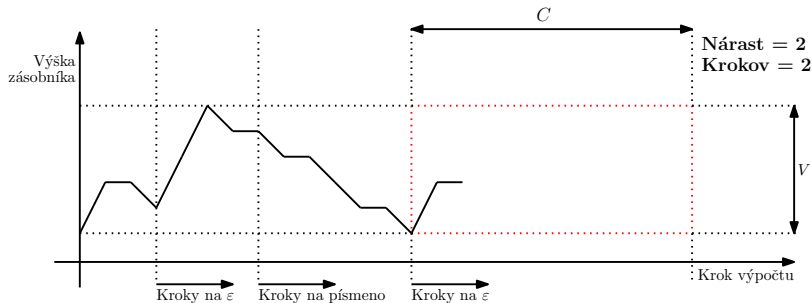
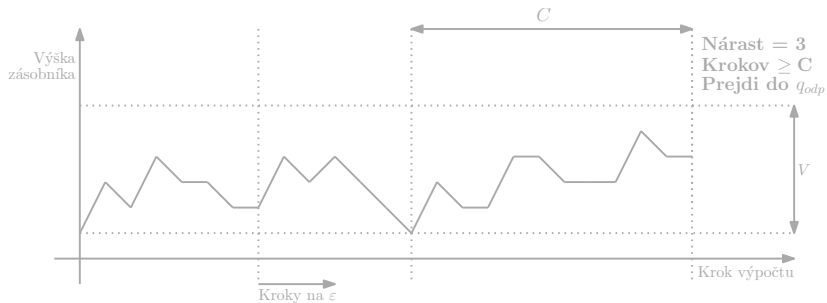
Riešenie problému 2



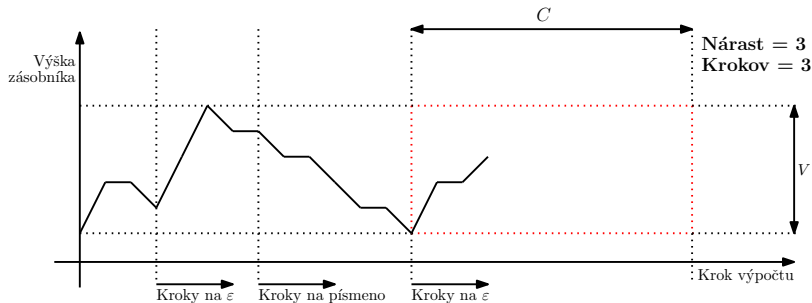
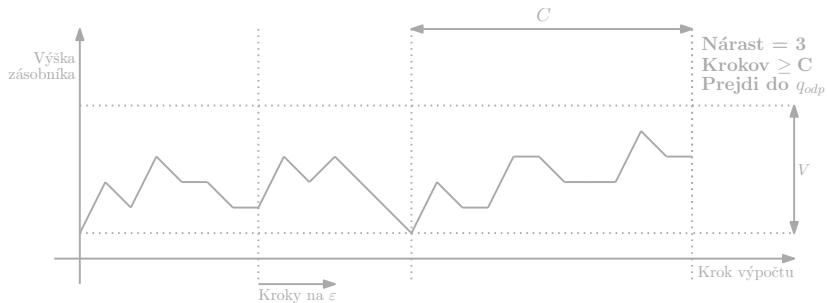
Riešenie problému 2



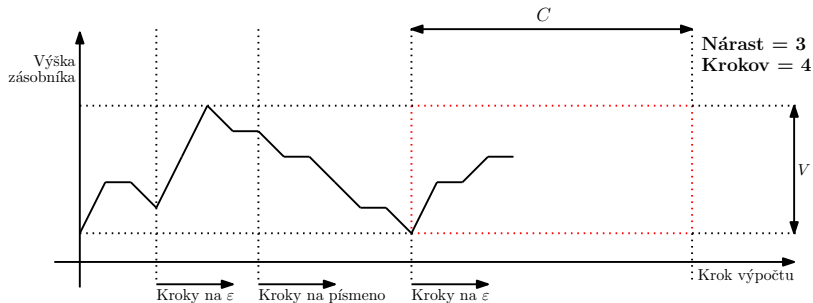
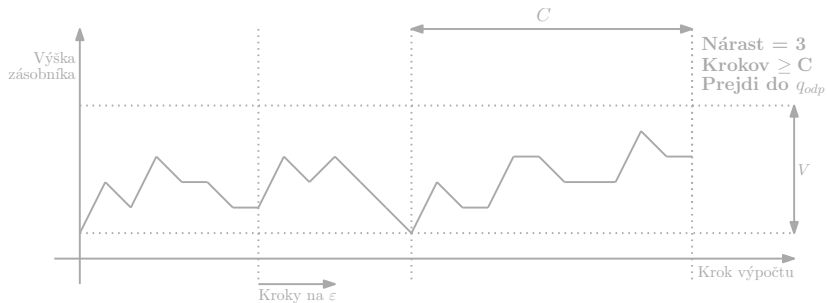
Riešenie problému 2



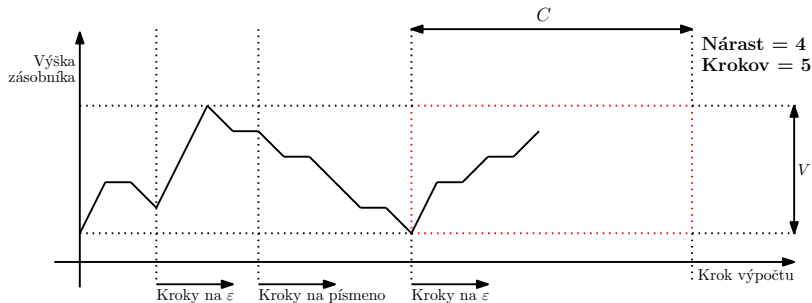
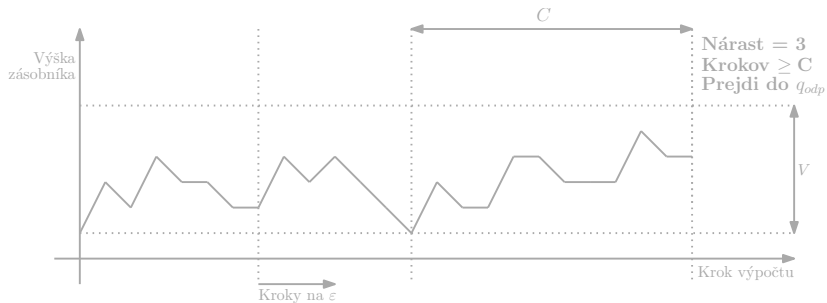
Riešenie problému 2



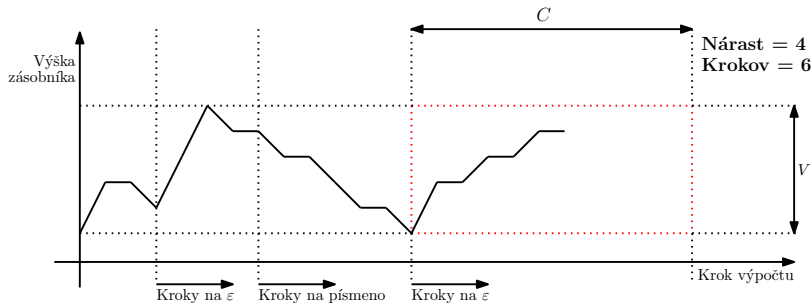
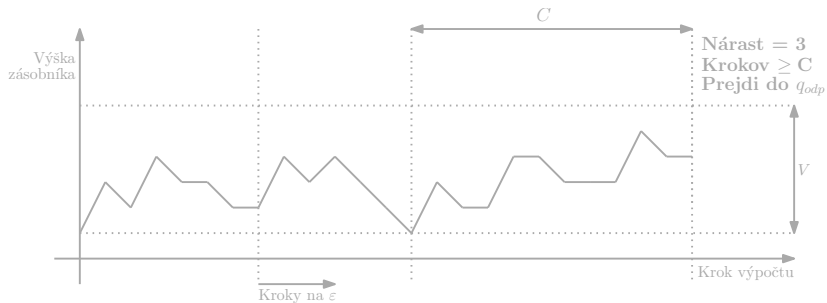
Riešenie problému 2



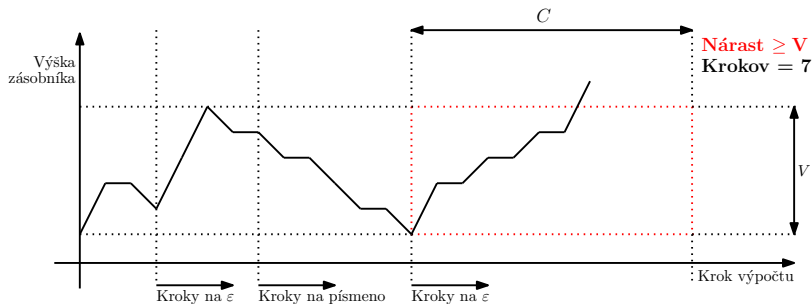
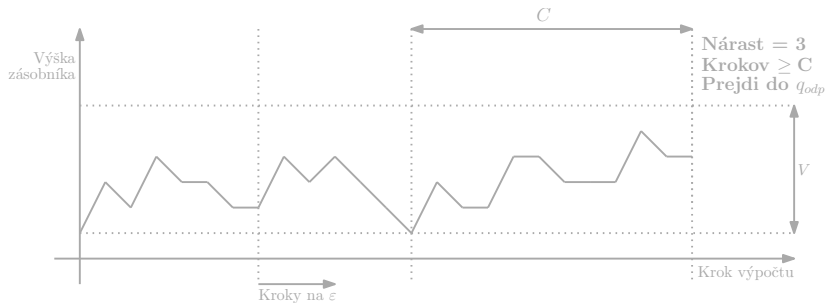
Riešenie problému 2



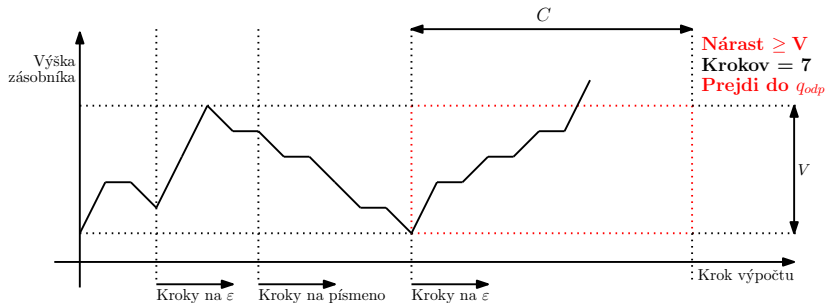
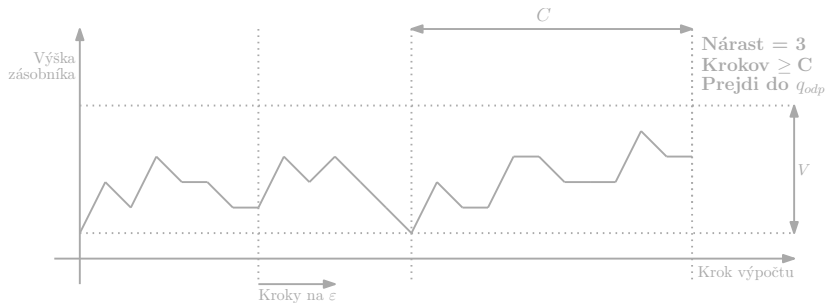
Riešenie problému 2



Riešenie problému 2



Riešenie problému 2



Riešenie problému 2 – formálna konštrukcia

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

- ▶ BUNV nech A je „nezasekávajúci sa“

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

- ▶ BUNV nech A je „nezasekávajúci sa“
- ▶ $K' = (K \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}) \cup \{q_{odp}\}$
(v stave je uložený nárast a počet krokov)

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

- ▶ BUNV nech A je „nezasekávajúci sa“
- ▶ $K' = (K \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}) \cup \{q_{odp}\}$
(v stave je uložený nárast a počet krokov)
- ▶ $q'_0 = [q_0, 0, 0]$

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

- ▶ BUNV nech A je „nezasekávajúci sa“
- ▶ $K' = (K \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}) \cup \{q_{odp}\}$
(v stave je uložený nárast a počet krokov)
- ▶ $q'_0 = [q_0, 0, 0]$
- ▶ $F' = F \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}$

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

- ▶ BUNV nech A je „nezasekávajúci sa“
- ▶ $K' = (K \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}) \cup \{q_{odp}\}$
(v stave je uložený nárast a počet krokov)
- ▶ $q'_0 = [q_0, 0, 0]$
- ▶ $F' = F \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}$
- ▶ Do q_{odp} bude možné predĺžiť aj niektoré akceptačné výpočty

Riešenie problému 2 – formálna konštrukcia

Lema

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = L(A)$, ktorý každý vstup dočíta do konca a zastane.

- ▶ BUNV nech A je „nezasekávajúci sa“
- ▶ $K' = (K \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}) \cup \{q_{odp}\}$
(v stave je uložený nárast a počet krokov)
- ▶ $q'_0 = [q_0, 0, 0]$
- ▶ $F' = F \times \{0, \dots, V - 1\} \times \{0, \dots, C - 1\}$
- ▶ Do q_{odp} bude možné predĺžiť aj niektoré akceptačné výpočty

Pôvodný prechod		Nový prechod	
$\delta(p, c, Z) = (q, \beta)$	$c \in \Sigma$	$\delta'([p, s, t], c, Z) = ([q, 0, 0], \beta)$	$\forall s, t$
$\delta(p, \varepsilon, Z) = (q, \varepsilon)$		$\delta'([p, 0, t], \varepsilon, Z) = ([q, 0, 0], \varepsilon)$	$\forall t$
		$\delta'([p, s, C - 1], \varepsilon, Z) = (q_{odp}, Z)$	$\forall s \geq 1$
		$\delta'([p, s, t], \varepsilon, Z) = ([q, s - 1, t + 1], \varepsilon)$	$\forall s \geq 1, t < C - 1$
$\delta(p, \varepsilon, Z) = (q, \beta)$	$ \beta \geq 1$	$\delta'([p, s, t], \varepsilon, Z) = (q_{odp}, Z)$	$\forall s \geq V - \beta + 1, t$
	$ \beta \geq 1$	$\delta'([p, s, C - 1], \varepsilon, Z) = (q_{odp}, Z)$	$\forall s$
	$ \beta \geq 1$	$\delta'([p, s, t], \varepsilon, Z) = ([q, s + \beta - 1, t + 1], \beta)$	$\forall s \leq V - \beta , t < C - 1$
		$\delta'(q_{odp}, c, Z) = (q_{odp}, Z)$	$\forall c \in \Sigma, Z \in \Gamma$

Konstrukcia DPDA akceptujúceho komplement

Konštrukcia DPDA akceptujúceho komplement

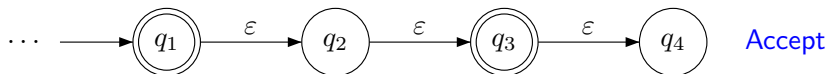
- ▶ Naivný prístup 3: výmena akceptačných a neakceptačných stavov DPDA, ktorý každý vstup dočíta do konca a zastane

Konštrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 3:** výmena akceptačných a neakceptačných stavov DPDA, ktorý každý vstup dočíta do konca a zastane
- ▶ **Problém 3:** výpočet sa môže skončiť niekoľkými krokmi na ε

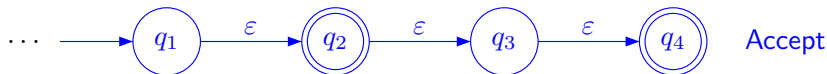
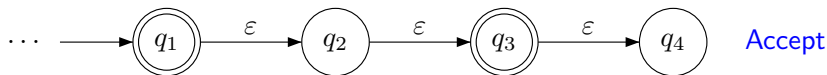
Konstrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 3:** výmena akceptačných a neakceptačných stavov DPDA, ktorý každý vstup dočíta do konca a zastane
- ▶ **Problém 3:** výpočet sa môže skončiť niekoľkými krokmi na ε



Konstrukcia DPDA akceptujúceho komplement

- ▶ **Naivný prístup 3:** výmena akceptačných a neakceptačných stavov DPDA, ktorý každý vstup dočíta do konca a zastane
- ▶ **Problém 3:** výpočet sa môže skončiť niekoľkými krokmi na ε



Riešenie problému 3

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Nie	Nie	$\dots$	Nie	Nie

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

- Pre každý stav $q \in K$ bude mať A' (??? dva ???) stavy:

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

- Pre každý stav $q \in K$ bude mať A' (??? dva ???) stavy:
 - $[q, 0]$: zatiaľ žiaden akceptačný stav v „ ε -chvoste“

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

- ▶ Pre každý stav $q \in K$ bude mať A' (??? dva ???) stavy:
 - ▶ $[q, 0]$: zatiaľ žiaden akceptačný stav v „ ε -chvoste“
 - ▶ $[q, 1]$: aspoň jeden akceptačný stav v „ ε -chvoste“

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

- Pre každý stav $q \in K$ bude mať A' (??? dva ???) stavy:
 - $[q, 0]$: zatiaľ žiaden akceptačný stav v „ ε -chvoste“
 - $[q, 1]$: aspoň jeden akceptačný stav v „ ε -chvoste“

$[q_1, 0]$	$[q_2, 1]$	$[q_3, 1]$	$\dots$	$[q_{m-1}, 1]$	$[q_m, 1]$
Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
Áno	Áno	Áno	$\dots$	Áno	Nie
Nie	Áno	Nie	$\dots$	Nie	Nie

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

- Pre každý stav $q \in K$ bude mať A' štyri stavy:
 - $[q, 0]$: zatiaľ žiaden akceptačný stav v „ ε -chvoste“
 - $[q, 1]$: aspoň jeden akceptačný stav v „ ε -chvoste“
 - $[q, 2]$: ako $[q, 0]$, ale iba na konci „ ε -chvostu“
 - $[q, 3]$: ako $[q, 1]$, ale iba na konci „ ε -chvostu“

$[q_1, 0]$	$[q_2, 1]$	$[q_3, 1]$	$\dots$	$[q_{m-1}, 1]$	$[q_m, 1]$
Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
Áno	Áno	Áno	$\dots$	Áno	Nie
Nie	Áno	Nie	$\dots$	Nie	Nie

Riešenie problému 3

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- BUNV nech A dočíta každý vstup do konca a zastane

$q \in K$	q_1	q_2	q_3	$\dots$	q_{m-1}	q_m
$Z \in \Gamma$	Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m
$\delta(q, \varepsilon, Z) \neq \emptyset?$	Áno	Áno	Áno	$\dots$	Áno	Nie
$q \in F?$	Nie	Áno	Nie	$\dots$	Nie	Nie

- Pre každý stav $q \in K$ bude mať A' štyri stavy:
 - $[q, 0]$: zatiaľ žiaden akceptačný stav v „ ε -chvoste“
 - $[q, 1]$: aspoň jeden akceptačný stav v „ ε -chvoste“
 - $[q, 2]$: ako $[q, 0]$, ale iba na konci „ ε -chvostu“
 - $[q, 3]$: ako $[q, 1]$, ale iba na konci „ ε -chvostu“

$[q_1, 0]$	$[q_2, 1]$	$[q_3, 1]$	$\dots$	$[q_{m-1}, 1]$	$[q_m, 1]$	$[q_m, 3]$
Z_1	Z_2	Z_3	$\dots$	Z_{m-1}	Z_m	Z_m
Áno	Áno	Áno	$\dots$	Áno	Áno	Nie
Nie	Áno	Nie	$\dots$	Nie	Nie	

Riešenie problému 3 – formálna konštrukcia

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

Riešenie problému 3 – formálna konštrukcia

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane

Riešenie problému 3 – formálna konštrukcia

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane
- ▶ $K' = K \times \{0, 1, 2, 3\}$

Riešenie problému 3 – formálna konštrukcia

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane
- ▶ $K' = K \times \{0, 1, 2, 3\}$
- ▶ Ak $q_0 \in F$, tak $q'_0 = [q_0, 1]$; inak $q'_0 = [q_0, 0]$

Riešenie problému 3 – formálna konštrukcia

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane
- ▶ $K' = K \times \{0, 1, 2, 3\}$
- ▶ Ak $q_0 \in F$, tak $q'_0 = [q_0, 1]$; inak $q'_0 = [q_0, 0]$
- ▶ $F' = \{[q, 2] \mid q \in K\}$

Riešenie problému 3 – formálna konštrukcia

Veta

Pre každý DPDA A existuje DPDA A' taký, že $L(A') = L(A)^C$.

- ▶ BUNV nech A dočíta každý vstup do konca a zastane
- ▶ $K' = K \times \{0, 1, 2, 3\}$
- ▶ Ak $q_0 \in F$, tak $q'_0 = [q_0, 1]$; inak $q'_0 = [q_0, 0]$
- ▶ $F' = \{[q, 2] \mid q \in K\}$

Pôvodný prechod	Nový prechod
$\delta(p, \varepsilon, Z) = (q, \gamma)$	$\delta'([p, 0], \varepsilon, Z) = ([q, 0], \gamma)$
$q \notin F$	$\delta'([p, 0], \varepsilon, Z) = ([q, 1], \gamma)$
$q \in F$	$\delta'([p, 1], \varepsilon, Z) = ([q, 1], \gamma)$
q ľubovoľné	$\delta'([p, 1], \varepsilon, Z) = ([q, 1], \gamma)$
—————	$\delta'([p, 0], \varepsilon, Z) = ([p, 2], Z)$
$\delta(p, \varepsilon, Z) = \emptyset$	$\delta'([p, 1], \varepsilon, Z) = ([p, 3], Z)$
$\delta(p, c, Z) = (q, \gamma)$	$\delta'([p, 2], c, Z) = ([q, 0], \gamma)$
$c \in \Sigma, q \notin F$	$\delta'([p, 3], c, Z) = ([q, 0], \gamma)$
$c \in \Sigma, q \notin F$	$\delta'([p, 2], c, Z) = ([q, 1], \gamma)$
$c \in \Sigma, q \in F$	$\delta'([p, 3], c, Z) = ([q, 1], \gamma)$
$c \in \Sigma, q \in F$	$\delta'([p, 3], c, Z) = ([q, 1], \gamma)$

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Trieda $\mathcal{L}_{\text{detCF}}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{\text{detCF}} \subsetneq \mathcal{L}_{\text{CF}}$.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Dôkaz.

$\subseteq$: Každý DPDA je súčasne aj PDA.

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Dôkaz.

$\subseteq$: Každý DPDA je súčasne aj PDA.

$\not\subseteq$: Trieda $\mathcal{L}_{detCF}$ je uzavretá na komplement a $\mathcal{L}_{CF}$ nie je.



Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Dôkaz.

$\subseteq$: Každý DPDA je súčasne aj PDA.

$\not\subseteq$: Trieda $\mathcal{L}_{detCF}$ je uzavretá na komplement a $\mathcal{L}_{CF}$ nie je. $\square$

Veta

Je $\mathcal{R} \subsetneq \mathcal{L}_{detCF}$.

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Dôkaz.

$\subseteq$: Každý DPDA je súčasne aj PDA.

$\not\subseteq$: Trieda $\mathcal{L}_{detCF}$ je uzavretá na komplement a $\mathcal{L}_{CF}$ nie je. $\square$

Veta

Je $\mathcal{R} \subsetneq \mathcal{L}_{detCF}$.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Dôkaz.

$\subseteq$: Každý DPDA je súčasne aj PDA.

$\not\supseteq$: Trieda $\mathcal{L}_{detCF}$ je uzavretá na komplement a $\mathcal{L}_{CF}$ nie je. $\square$

Veta

Je $\mathcal{R} \subsetneq \mathcal{L}_{detCF}$.

Dôkaz.

$\subseteq$: Každý DKA možno interpretovať ako DPDA.

Trieda $\mathcal{L}_{detCF}$ a Chomského hierarchia

Veta

Je $\mathcal{L}_{detCF} \subsetneq \mathcal{L}_{CF}$.

Dôkaz.

$\subseteq$: Každý DPDA je súčasne aj PDA.

$\not\subseteq$: Trieda $\mathcal{L}_{detCF}$ je uzavretá na komplement a $\mathcal{L}_{CF}$ nie je. $\square$

Veta

Je $\mathcal{R} \subsetneq \mathcal{L}_{detCF}$.

Dôkaz.

$\subseteq$: Každý DKA možno interpretovať ako DPDA.

$\not\subseteq$: $\{a^n b^n \mid n \in \mathbb{N}\} \in \mathcal{L}_{detCF} - \mathcal{R}$. $\square$

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}^C$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}^C$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}^C$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

Vieme, že $L \in \mathcal{L}_{CF}$. Keby bolo $L \in \mathcal{L}_{detCF}$, musel by byť jazyk $L^C = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ v $\mathcal{L}_{detCF}$, a teda aj v $\mathcal{L}_{CF}$. □

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- Evidentne $L \in \mathcal{L}_{CF}$

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A' “ s prechodmi na b nahradenými prechodmi na c

Príklady jazykov z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A' “ s prechodmi na b nahradenými prechodmi na c

$$L(A') = L \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2n} \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

Príklady jazykov z $\mathcal{L}_{CF}$ – $\mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF}$ – $\mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A' “ s prechodmi na b nahradenými prechodmi na c

$$L(A') = L \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2n} \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

- ▶ Očividne $L(A') \notin \mathcal{L}_{CF}$

Príklady jazykov z $\mathcal{L}_{CF}$ – $\mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} \mid n \in \mathbb{N}\}$ je v $\mathcal{L}_{CF}$ – $\mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Evidentne $L \in \mathcal{L}_{CF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A' “ s prechodmi na b nahradenými prechodmi na c

$$L(A') = L \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2n} \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

- ▶ Očividne $L(A') \notin \mathcal{L}_{CF}$
- ▶ Spor



Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	áno
Označené zjednotenie	
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	
Komplement	
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	áno
Označené zjednotenie	
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	
Komplement	
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a zjednotenie

Trieda $\mathcal{L}_{detCF}$ a zjednotenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zjednotenie.

Trieda $\mathcal{L}_{detCF}$ a zjednotenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zjednotenie.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a zjednotenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zjednotenie.

Dôkaz.

► $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ sú v $\mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a zjednotenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zjednotenie.

Dôkaz.

- ▶ $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$ sú v $\mathcal{L}_{detCF}$
- ▶ Dokázali sme, že jazyk $L_1 \cup L_2$ nie je v $\mathcal{L}_{detCF}$



Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	
Zreťazenie	
Prienik	áno
Prienik s regulárnym jazykom	
Komplement	
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^$ sú jazyky a $a \notin \Sigma$ je nový symbol.*

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^$ sú jazyky a $a \notin \Sigma$ je nový symbol.*

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^$ sú jazyky a $a \notin \Sigma$ je nový symbol.*

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

- Nech $L_1 = L(A_1)$ a $L_2 = L(A_2)$, kde A_1, A_2 sú DPDA

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^$ sú jazyky a $a \notin \Sigma$ je nový symbol.*

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Nech $L_1 = L(A_1)$ a $L_2 = L(A_2)$, kde A_1, A_2 sú DPDA
- ▶ BUNV nech $K_1 \cap K_2 = \emptyset$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ a nech A_2 vždy dočíta vstup do konca a zastane, pričom nevyprázdni zásobník

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^$ sú jazyky a $a \notin \Sigma$ je nový symbol.*

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Nech $L_1 = L(A_1)$ a $L_2 = L(A_2)$, kde A_1, A_2 sú DPDA
- ▶ BUNV nech $K_1 \cap K_2 = \emptyset$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ a nech A_2 vždy dočíta vstup do konca a zastane, pričom nevyprázdni zásobník
- ▶ BUNV nech $K_2 = K'_2 \cup K''_2$, pričom pred vykonaním prvého prechodu na písmeno je A_2 vždy v stave z K'_2 a po jeho vykonaní v stave z K''_2

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^$ sú jazyky a $a \notin \Sigma$ je nový symbol.*

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Nech $L_1 = L(A_1)$ a $L_2 = L(A_2)$, kde A_1, A_2 sú DPDA
- ▶ BUNV nech $K_1 \cap K_2 = \emptyset$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ a nech A_2 vždy dočíta vstup do konca a zastane, pričom nevyprázdni zásobník
- ▶ BUNV nech $K_2 = K'_2 \cup K''_2$, pričom pred vykonaním prvého prechodu na písmeno je A_2 vždy v stave z K'_2 a po jeho vykonaní v stave z K''_2
- ▶ Začneme simulovať automat A_2

Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky a $a \notin \Sigma$ je nový symbol.

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Nech $L_1 = L(A_1)$ a $L_2 = L(A_2)$, kde A_1, A_2 sú DPDA
- ▶ BUNV nech $K_1 \cap K_2 = \emptyset$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ a nech A_2 vždy dočíta vstup do konca a zastane, pričom nevyprázdni zásobník
- ▶ BUNV nech $K_2 = K'_2 \cup K''_2$, pričom pred vykonaním prvého prechodu na písmeno je A_2 vždy v stave z K'_2 a po jeho vykonaní v stave z K''_2
- ▶ Začneme simulovať automat A_2
- ▶ Kedykoľvek pre nejaké $q \in K'_2$ a $Z \in \Gamma_2$ nevieme spraviť prechod na ε , pridáme prechod na a umožňujúci začať simuláciu automatu A_1



Trieda $\mathcal{L}_{detCF}$ a označené zjednotenie

Veta

Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky a $a \notin \Sigma$ je nový symbol.

Ak $L_1, L_2 \in \mathcal{L}_{detCF}$, tak aj $aL_1 \cup L_2 \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Nech $L_1 = L(A_1)$ a $L_2 = L(A_2)$, kde A_1, A_2 sú DPDA
- ▶ BUNV nech $K_1 \cap K_2 = \emptyset$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ a nech A_2 vždy dočíta vstup do konca a zastane, pričom nevyprázdni zásobník
- ▶ BUNV nech $K_2 = K'_2 \cup K''_2$, pričom pred vykonaním prvého prechodu na písmeno je A_2 vždy v stave z K'_2 a po jeho vykonaní v stave z K''_2
- ▶ Začneme simulovať automat A_2
- ▶ Kedykoľvek pre nejaké $q \in K'_2$ a $Z \in \Gamma_2$ nevieme spraviť prechod na ε , pridáme prechod na a umožňujúci začať simuláciu automatu A_1



Iný druh označeného zjednotenia: $aL_1 \cup bL_2$, kde $a \neq b$ (aj zo Σ)

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a prienik s regulárnym jazykom

Trieda $\mathcal{L}_{detCF}$ a prienik s regulárnym jazykom

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na prienik s regulárnym jazykom.

Trieda $\mathcal{L}_{detCF}$ a prienik s regulárnym jazykom

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na prienik s regulárnym jazykom.

- Funguje rovnaká konštrukcia ako pre $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	
Prienik	
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Trieda $\mathcal{L}_{\det CF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{\det CF}$ nie je uzavretá na zreťazenie.

Trieda $\mathcal{L}_{\det CF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{\det CF}$ nie je uzavretá na zreťazenie.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$
a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$
a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)

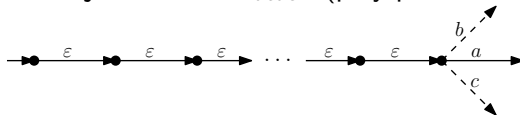
Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)



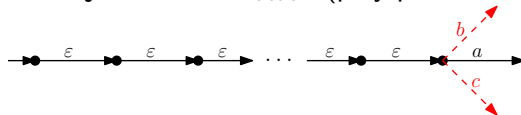
Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)



Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)



Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)



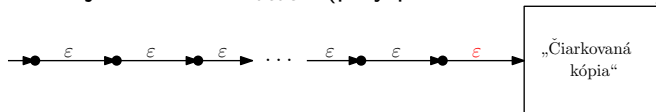
Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)



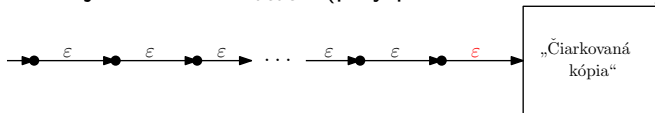
Trieda $\mathcal{L}_{detCF}$ a zreťazenie

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na zreťazenie.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Ak $a \notin \Sigma$ je *nový symbol*, tak $L_3 := aL_1 \cup L_2 \in \mathcal{L}_{detCF}$
- ▶ Jazyk a^* je regulárny, a teda v $\mathcal{L}_{detCF}$
- ▶ Ukážeme, že jazyk a^*L_3 nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, tak aj $a^*L_3 \cap a\Sigma^* = aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$
- ▶ Potom aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$ (prvý prechod na $a \rightsquigarrow$ na ε)



- ▶ Spor



Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a prienik

Trieda $\mathcal{L}_{detCF}$ a prienik

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na prienik.

Trieda $\mathcal{L}_{detCF}$ a prienik

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na prienik.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a prienik

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na prienik.

Dôkaz.

Z uzavretosti na prienik a komplement by podľa De Morganových zákonov vyplývala uzavretosť na zjednotenie – spor. □

Trieda $\mathcal{L}_{detCF}$ a prienik

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na prienik.

Dôkaz.

Z uzavretosti na prienik a komplement by podľa De Morganových zákonov vyplývala uzavretosť na zjednotenie – spor. □

Alternatívny dôkaz.

Trieda $\mathcal{L}_{detCF}$ a prienik

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na prienik.

Dôkaz.

Z uzavretosti na prienik a komplement by podľa De Morganových zákonov vyplývala uzavretosť na zjednotenie – spor. □

Alternatívny dôkaz.

$$L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\} \in \mathcal{L}_{detCF},$$

$$L_2 = \{a^n b^m c^m \mid n, m \in \mathbb{N}\} \in \mathcal{L}_{detCF}, \text{ ale } L_1 \cap L_2 \notin \mathcal{L}_{CF}. \quad \square$$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	
Kladná iterácia	
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a iterácia

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

- Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

- ▶ Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$
- ▶ Jazyk $L_3 := cL_1 \cup L_2$ je v $\mathcal{L}_{detCF}$ (zrejme $c \in L_3$)

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

- ▶ Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$
- ▶ Jazyk $L_3 := cL_1 \cup L_2$ je v $\mathcal{L}_{detCF}$ (zrejme $c \in L_3$)
- ▶ Ukážeme, že L_3^* nie je v $\mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

- ▶ Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$
- ▶ Jazyk $L_3 := cL_1 \cup L_2$ je v $\mathcal{L}_{detCF}$ (zrejme $c \in L_3$)
- ▶ Ukážeme, že L_3^* nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, bol by v $\mathcal{L}_{detCF}$ aj $L_3^* \cap ca^* b^* = cL_1 \cup cL_2$

Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

- ▶ Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$
- ▶ Jazyk $L_3 := cL_1 \cup L_2$ je v $\mathcal{L}_{detCF}$ (zrejme $c \in L_3$)
- ▶ Ukážeme, že L_3^* nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, bol by v $\mathcal{L}_{detCF}$ aj $L_3^* \cap ca^* b^* = cL_1 \cup cL_2$
- ▶ A potom by bol v $\mathcal{L}_{detCF}$ aj jazyk $L_1 \cup L_2$ – spor



Trieda $\mathcal{L}_{detCF}$ a iterácia

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na iteráciu.

Dôkaz.

- ▶ Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$
- ▶ Jazyk $L_3 := cL_1 \cup L_2$ je v $\mathcal{L}_{detCF}$ (zrejme $c \in L_3$)
- ▶ Ukážeme, že L_3^* nie je v $\mathcal{L}_{detCF}$
- ▶ Keby bol, bol by v $\mathcal{L}_{detCF}$ aj $L_3^* \cap ca^* b^* = cL_1 \cup cL_2$
- ▶ A potom by bol v $\mathcal{L}_{detCF}$ aj jazyk $L_1 \cup L_2$ – spor
- ▶ Rovnaký dôkaz možno použiť aj pre kladnú iteráciu



Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a reverz

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Trieda $\mathcal{L}_{\text{detCF}}$ a reverz

Veta

Trieda $\mathcal{L}_{\text{detCF}}$ nie je uzavretá na reverz.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L_3^R$

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L_3^R$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L_3^R$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L_3^R$
- ▶ Nech A' je nedeterministický zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A'' s prechodmi na b nahradenými prechodmi na c

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L_3^R$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A'' s prechodmi na b nahradenými prechodmi na c

$$L(A') = L \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2^n} d \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n d \mid n \in \mathbb{N}\}$$

Trieda $\mathcal{L}_{detCF}$ a reverz

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na reverz.

Dôkaz.

- ▶ Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2^n} a^n \mid n \in \mathbb{N}\}$
- ▶ Nech $L_3 = L_1 \cup dL_2 \in \mathcal{L}_{detCF}$
- ▶ $L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2^n} d \mid n \in \mathbb{N}\}$
- ▶ Ukážeme, že L_3^R nie je v $\mathcal{L}_{detCF}$
- ▶ Sporom, nech existuje DPDA A taký, že $L(A) = L_3^R$
- ▶ Nech A' je **nedeterministický** zásobníkový automat, ktorý:
 - ▶ Pracuje ako A
 - ▶ V akceptačných stavoch môže „prejsť do kópie A' “ s prechodmi na b nahradenými prechodmi na c

$$L(A') = L \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2^n} d \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n d \mid n \in \mathbb{N}\}$$

- ▶ Očividne $L(A') \notin \mathcal{L}_{CF}$ – **spor**



Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	nie
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	nie
Homomorfizmus	
Nevymazávajúci homomorfizmus	
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Trieda $\mathcal{L}_{\det CF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{\det CF}$ nie je uzavretá na homomorfizmus.

Trieda $\mathcal{L}_{\det CF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{\det CF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$
a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$
a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Nech $a, b \notin \Sigma$ sú nové symboly

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$
a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Nech $a, b \notin \Sigma$ sú nové symboly
- ▶ Jazyk $L_3 := aL_1 \cup bL_2$ je v $\mathcal{L}_{detCF}$

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Nech $a, b \notin \Sigma$ sú nové symboly
- ▶ Jazyk $L_3 := aL_1 \cup bL_2$ je v $\mathcal{L}_{detCF}$
- ▶ Nech h je homomorfizmus taký, že $h(a) = h(b) = \varepsilon$ a $h(c) = c$ pre $c \in \Sigma$

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Nech $a, b \notin \Sigma$ sú nové symboly
- ▶ Jazyk $L_3 := aL_1 \cup bL_2$ je v $\mathcal{L}_{detCF}$
- ▶ Nech h je homomorfizmus taký, že $h(a) = h(b) = \varepsilon$ a $h(c) = c$ pre $c \in \Sigma$
- ▶ Potom $h(L_3) = L_1 \cup L_2 \notin \mathcal{L}_{detCF}$. □

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Nech $a, b \notin \Sigma$ sú nové symboly
- ▶ Jazyk $L_3 := aL_1 \cup bL_2$ je v $\mathcal{L}_{detCF}$
- ▶ Nech h je homomorfizmus taký, že $h(a) = h(b) = \varepsilon$ a $h(c) = c$ pre $c \in \Sigma$
- ▶ Potom $h(L_3) = L_1 \cup L_2 \notin \mathcal{L}_{detCF}$. □
- ▶ Pomocou nevymazávajúceho homomorfizmu možno dostať jazyk $aL_1 \cup aL_2$

Trieda $\mathcal{L}_{detCF}$ a homomorfizmus

Veta

Trieda $\mathcal{L}_{detCF}$ nie je uzavretá na homomorfizmus.

Dôkaz.

- ▶ Nech $L_1, L_2 \subseteq \Sigma^*$ sú jazyky také, že $L_1, L_2 \in \mathcal{L}_{detCF}$ a $L_1 \cup L_2 \notin \mathcal{L}_{detCF}$
- ▶ Nech $a, b \notin \Sigma$ sú nové symboly
- ▶ Jazyk $L_3 := aL_1 \cup bL_2$ je v $\mathcal{L}_{detCF}$
- ▶ Nech h je homomorfizmus taký, že $h(a) = h(b) = \varepsilon$ a $h(c) = c$ pre $c \in \Sigma$
- ▶ Potom $h(L_3) = L_1 \cup L_2 \notin \mathcal{L}_{detCF}$. □
- ▶ Pomocou nevymazávajúceho homomorfizmu možno dostať jazyk $aL_1 \cup aL_2$
- ▶ Ak $aL_1 \cup aL_2 \in \mathcal{L}_{detCF}$, tak aj $L_1 \cup L_2 \in \mathcal{L}_{detCF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	nie
Homomorfizmus	nie
Nevymazávajúci homomorfizmus	nie
Inverzný homomorfizmus	

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	nie
Homomorfizmus	nie
Nevymazávajúci homomorfizmus	nie
Inverzný homomorfizmus	

Trieda $\mathcal{L}_{detCF}$ a inverzný homomorfizmus

Trieda $\mathcal{L}_{\det CF}$ a inverzný homomorfizmus

Veta

Trieda $\mathcal{L}_{\det CF}$ je uzavretá na inverzný homomorfizmus.

Trieda $\mathcal{L}_{\det CF}$ a inverzný homomorfizmus

Veta

Trieda $\mathcal{L}_{\det CF}$ je uzavretá na inverzný homomorfizmus.

- Funguje rovnaká konštrukcia ako pre $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	nie
Homomorfizmus	nie
Nevymazávajúci homomorfizmus	nie
Inverzný homomorfizmus	áno

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Veta

Uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$ sú dané nasledujúcou tabuľkou:

Zjednotenie	nie
Označené zjednotenie	áno
Zreťazenie	nie
Prienik	nie
Prienik s regulárnym jazykom	áno
Komplement	áno
Iterácia	nie
Kladná iterácia	nie
Reverz	nie
Homomorfizmus	nie
Nevymazávajúci homomorfizmus	nie
Inverzný homomorfizmus	áno

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

DPDA akceptujúce prázdny zásobník

DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

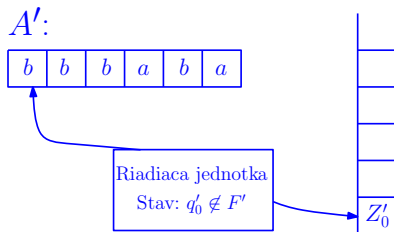
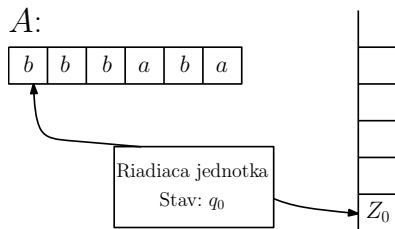
- ▶ Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

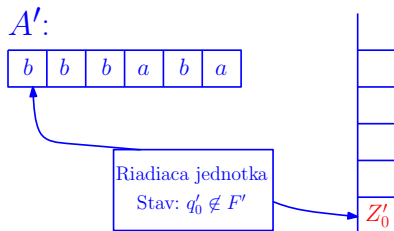
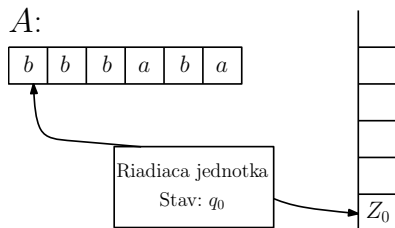


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

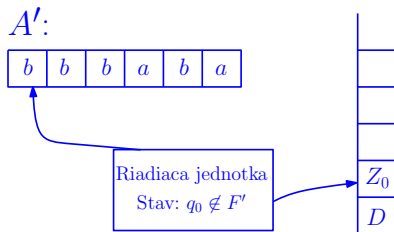
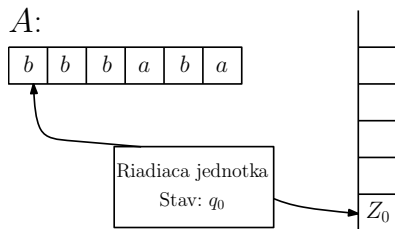


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

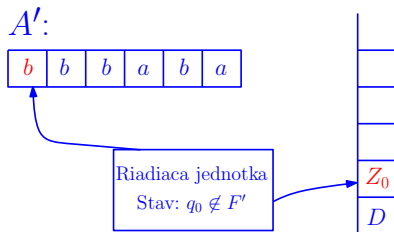
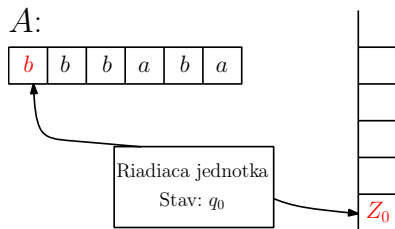


DPDA akceptujúce prázdny zásobníkom

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

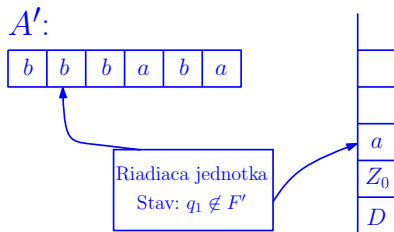
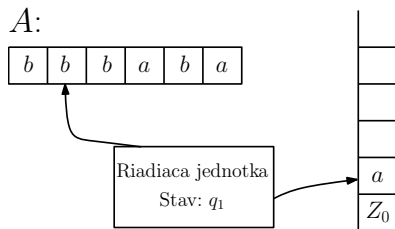


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

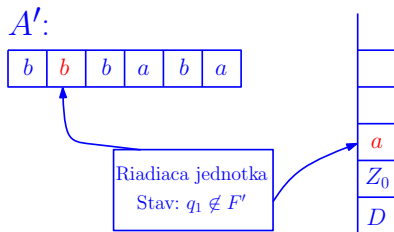
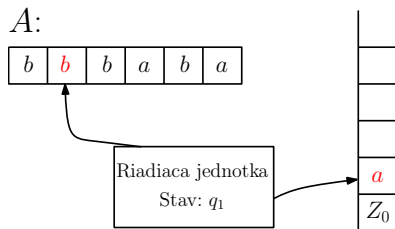


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

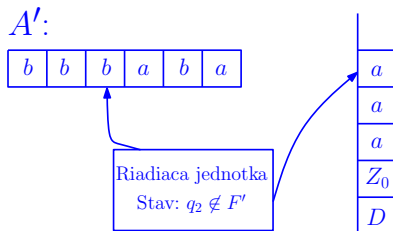
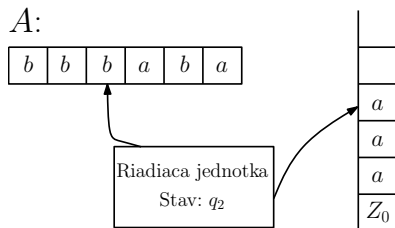


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

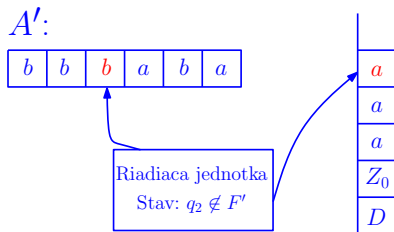
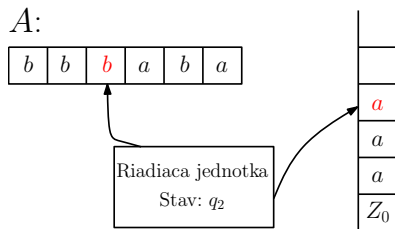


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

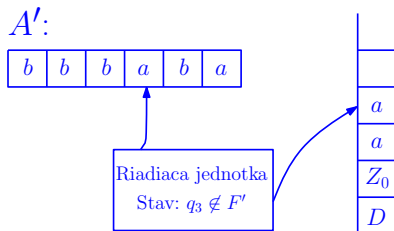
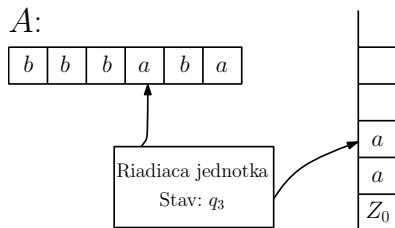


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

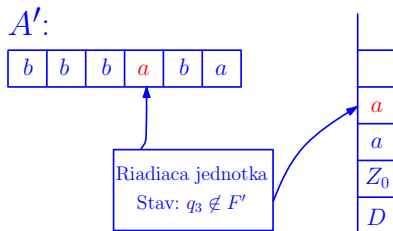
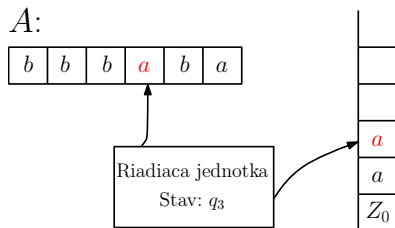


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

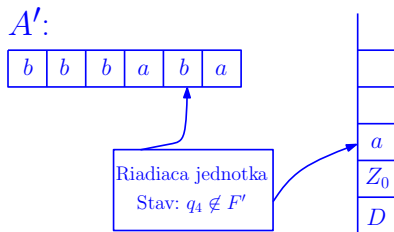
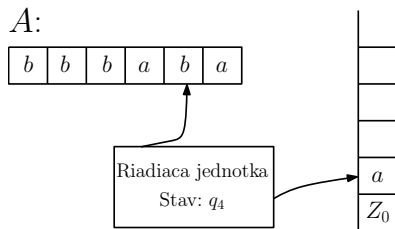


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

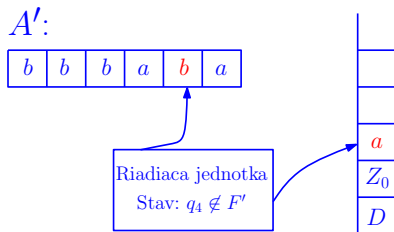
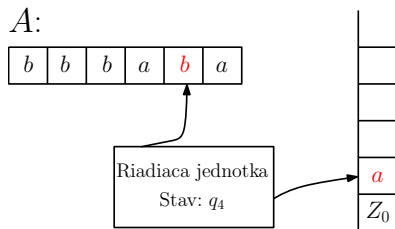


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

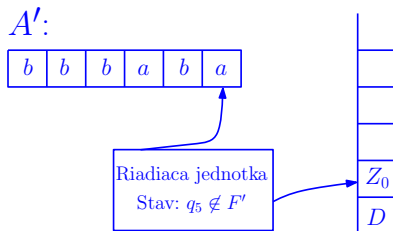
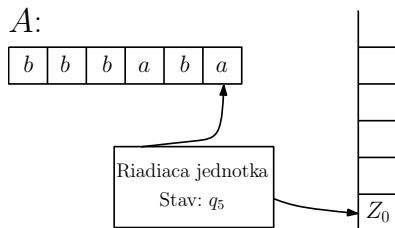


DPDA akceptujúce prázdny zásobníkom

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

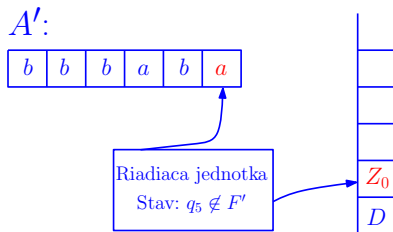
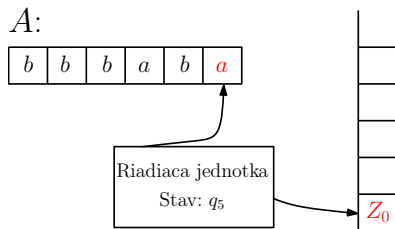


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

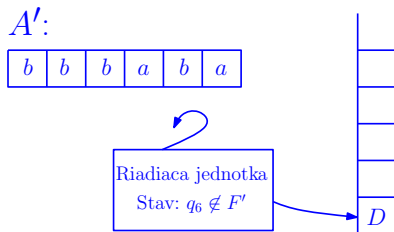
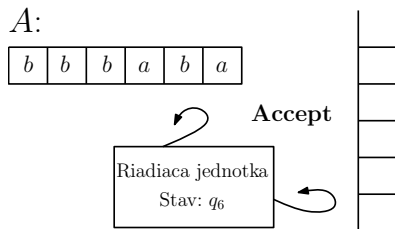


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

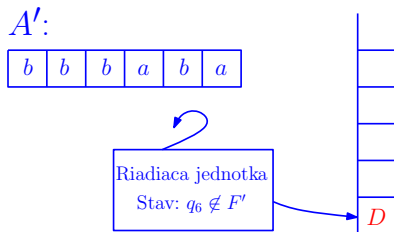
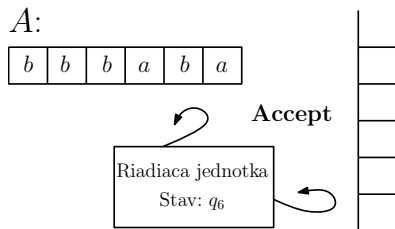


DPDA akceptujúce prázdny zásobník

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA

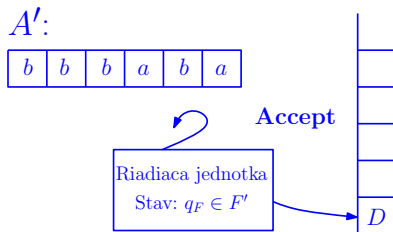
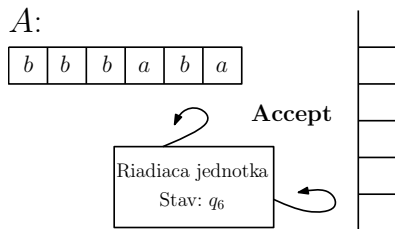


DPDA akceptujúce prázdny zásobníkom

Veta

Nech A je DPDA. Potom existuje DPDA A' taký, že $L(A') = N(A)$.

- Funguje rovnaká konštrukcia s „umelým dnom“ ako pre PDA



DPDA akceptujúce prázdny zásobník

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.*

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.*

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.*

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

Dôkaz.

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^*$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

Dôkaz.

- Nech $u \in N(A)$ a $v \in \Sigma^+$

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^*$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

Dôkaz.

- ▶ Nech $u \in N(A)$ a $v \in \Sigma^+$
- ▶ Nutne $(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ pre nejaké $q \in K$

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^*$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

Dôkaz.

- ▶ Nech $u \in N(A)$ a $v \in \Sigma^+$
- ▶ Nutne $(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ pre nejaké $q \in K$
- ▶ Potom aj $(q_0, uv, Z_0) \vdash^* (q, v, \varepsilon)$

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^*$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

Dôkaz.

- ▶ Nech $u \in N(A)$ a $v \in \Sigma^+$
- ▶ Nutne $(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ pre nejaké $q \in K$
- ▶ Potom aj $(q_0, uv, Z_0) \vdash^* (q, v, \varepsilon)$
- ▶ Každý výpočet na uv sa musí dostať do konfigurácie (q, v, ε)

DPDA akceptujúce prázdny zásobník

Definícia

Jazyk $L \subseteq \Sigma^*$ je bezprefixový, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ je $uv \notin L$ kedykoľvek $u \in L$.

Veta

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom je jazyk $N(A)$ bezprefixový.

Dôkaz.

- ▶ Nech $u \in N(A)$ a $v \in \Sigma^+$
- ▶ Nutne $(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ pre nejaké $q \in K$
- ▶ Potom aj $(q_0, uv, Z_0) \vdash^* (q, v, \varepsilon)$
- ▶ Každý výpočet na uv sa musí dostať do konfigurácie (q, v, ε)
- ▶ Z tejto konfigurácie nemožno pokračovať: $uv \notin N(A)$ □

DPDA akceptujúce prázdny zásobník

DPDA akceptujúce prázdny zásobník

Veta

Existuje jazyk $L \in \mathcal{L}_{\text{detCF}}$, pre ktorý neexistuje žiaden DPDA A taký, že $N(A) = L$.

DPDA akceptujúce prázdny zásobník

Veta

Existuje jazyk $L \in \mathcal{L}_{\text{detCF}}$, pre ktorý neexistuje žiaden DPDA A taký, že $N(A) = L$.

Dôkaz.

DPDA akceptujúce prázdny zásobník

Veta

Existuje jazyk $L \in \mathcal{L}_{\text{detCF}}$, pre ktorý neexistuje žiaden DPDA A taký, že $N(A) = L$.

Dôkaz.

- Nech $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$

DPDA akceptujúce prázdny zásobníkom

Veta

Existuje jazyk $L \in \mathcal{L}_{\text{detCF}}$, pre ktorý neexistuje žiaden DPDA A taký, že $N(A) = L$.

Dôkaz.

- ▶ Nech $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$
- ▶ Ľahko zostrojiť DPDA A taký, že $L(A) = L$

DPDA akceptujúce prázdny zásobníkom

Veta

Existuje jazyk $L \in \mathcal{L}_{\text{detCF}}$, pre ktorý neexistuje žiaden DPDA A taký, že $N(A) = L$.

Dôkaz.

- ▶ Nech $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$
- ▶ Ľahko zostrojiť DPDA A taký, že $L(A) = L$
- ▶ $\varepsilon \in L$ a súčasne $ab \in L$

DPDA akceptujúce prázdny zásobníkom

Veta

Existuje jazyk $L \in \mathcal{L}_{detCF}$, pre ktorý neexistuje žiaden DPDA A taký, že $N(A) = L$.

Dôkaz.

- ▶ Nech $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$
- ▶ Ľahko zostrojiť DPDA A taký, že $L(A) = L$
- ▶ $\varepsilon \in L$ a súčasne $ab \in L$
- ▶ Tvrdenie potom vyplýva z predchádzajúcej vety



DPDA akceptujúce prázdny zásobník

DPDA akceptujúce prázdny zásobník

Veta

Nech $L \in \mathcal{L}_{\text{detCF}}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

DPDA akceptujúce prázdny zásobník

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

DPDA akceptujúce prázdny zásobník

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)

DPDA akceptujúce prázdny zásobníkom

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)
- ▶ BUNV nech B nemá prechody na ε z akceptačných stavov

DPDA akceptujúce prázdny zásobníkom

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)
- ▶ BUNV nech B nemá prechody na ε z akceptačných stavov
- ▶ Z akceptačných stavov nový prechod na $\$$ umožňujúci postupne vyprázdniť zásobník



DPDA akceptujúce prázdny zásobníkom

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)
- ▶ BUNV nech B nemá prechody na ε z akceptačných stavov
- ▶ Z akceptačných stavov nový prechod na $\$$ umožňujúci postupne vyprázdniť zásobník



Veta

K jazyku $L \in \mathcal{L}_{detCF}$ existuje DPDA A taký, že $L = N(A)$ práve vtedy, keď je L bezprefixový.

DPDA akceptujúce prázdny zásobníkom

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)
- ▶ BUNV nech B nemá prechody na ε z akceptačných stavov
- ▶ Z akceptačných stavov nový prechod na $\$$ umožňujúci postupne vyprázdniť zásobník



Veta

K jazyku $L \in \mathcal{L}_{detCF}$ existuje DPDA A taký, že $L = N(A)$ práve vtedy, keď je L bezprefixový.

Dôkaz.

DPDA akceptujúce prázdny zásobníkom

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)
- ▶ BUNV nech B nemá prechody na ε z akceptačných stavov
- ▶ Z akceptačných stavov nový prechod na $\$$ umožňujúci postupne vyprázdniť zásobník



Veta

K jazyku $L \in \mathcal{L}_{detCF}$ existuje DPDA A taký, že $L = N(A)$ práve vtedy, keď je L bezprefixový.

Dôkaz.

$\Rightarrow$: Už dokázané

DPDA akceptujúce prázdny zásobníkom

Veta

Nech $L \in \mathcal{L}_{detCF}$ je jazyk. Potom existuje DPDA A taký, že $N(A) = L\$$, kde $\$$ je nový symbol.

Dôkaz.

- ▶ Jazyk $L = L(B)$ pre nejaký DPDA B , ktorý nikdy nevyprázdni zásobník („umelé dno“)
- ▶ BUNV nech B nemá prechody na ε z akceptačných stavov
- ▶ Z akceptačných stavov nový prechod na $\$$ umožňujúci postupne vyprázdniť zásobník



Veta

K jazyku $L \in \mathcal{L}_{detCF}$ existuje DPDA A taký, že $L = N(A)$ práve vtedy, keď je L bezprefixový.

Dôkaz.

$\Rightarrow$: Už dokázané

$\Leftarrow$: Ako vo vete hore, ale namiesto $\$$ čítame ε



Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Normálny tvar pre DPDA

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.*

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.*

- ▶ Jednotlivé prechody „prerobiť na postupnosti prechodov“

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.*

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak*

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.*

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak*

- (i) $\beta = \varepsilon$, alebo*

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak

- (i) $\beta = \varepsilon$, alebo
- (ii) $\beta = Z$, alebo

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak

- (i) $\beta = \varepsilon$, alebo
- (ii) $\beta = Z$, alebo
- (iii) $\beta = ZY$ pre nejaké $Y \in \Gamma$.

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak

- (i) $\beta = \varepsilon$, alebo
- (ii) $\beta = Z$, alebo
- (iii) $\beta = ZY$ pre nejaké $Y \in \Gamma$.

- Nech B je v normálnom tvare z predchádzajúcej lemy

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak

- (i) $\beta = \varepsilon$, alebo
- (ii) $\beta = Z$, alebo
- (iii) $\beta = ZY$ pre nejaké $Y \in \Gamma$.

- Nech B je v normálnom tvare z predchádzajúcej lemy
- Zaved'me „umelé dno“ D

Normálny tvar pre DPDA

Lema

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak $|\beta| \leq 2$.

- Jednotlivé prechody „prerobiť na postupnosti prechodov“

Veta

Nech B je DPDA. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L(B)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\beta \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \beta)$, tak

- (i) $\beta = \varepsilon$, alebo
- (ii) $\beta = Z$, alebo
- (iii) $\beta = ZY$ pre nejaké $Y \in \Gamma$.

- Nech B je v normálnom tvare z predchádzajúcej lemy
- Zaved'me „umelé dno“ D
- Potom stačí pamätať si vrch zásobníka v stave

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Predpovedajúce stroje

Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.

Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

Predpovedajúce stroje

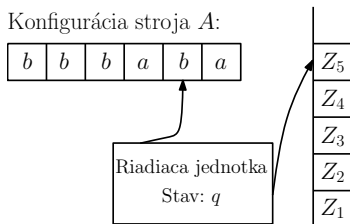
- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:

Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

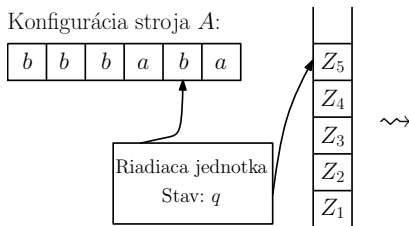
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

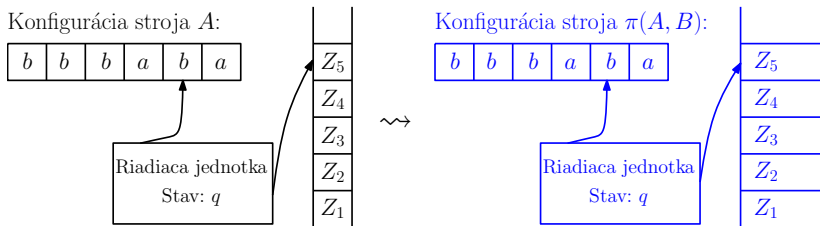
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

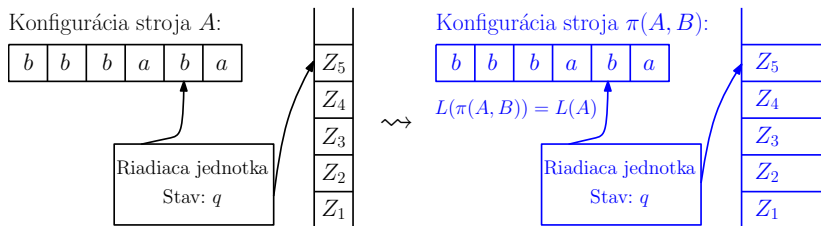
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

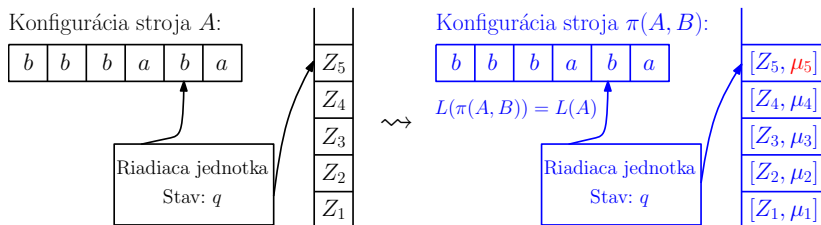
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

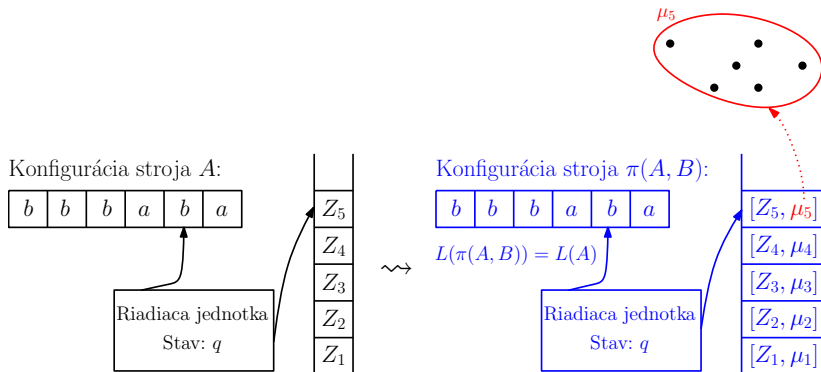
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

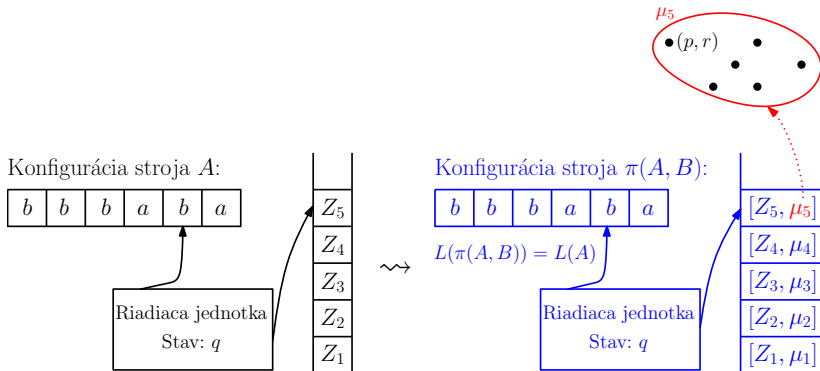
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

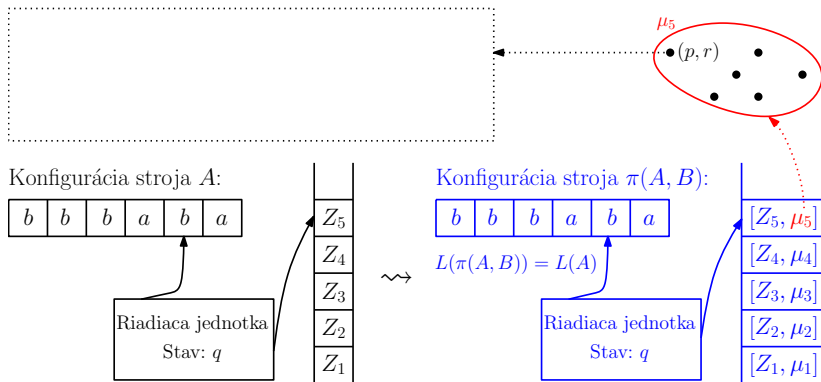
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:



Predpovedajúce stroje

- Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:

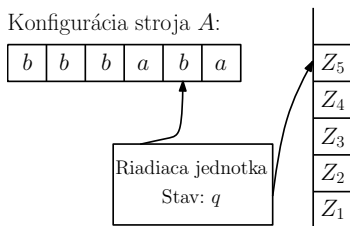


Predpovedajúce stroje

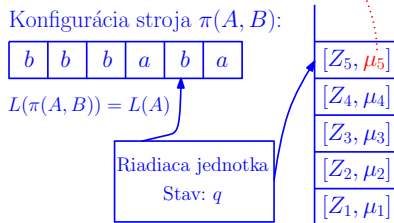
- Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:

$(p, r) \in K_A \times K_B$ je v μ_5 práve vtedy, keď $\exists w \in \Sigma^*$:



$\rightsquigarrow$



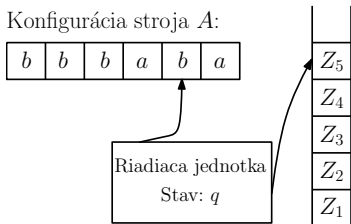
Predpovedajúce stroje

- Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

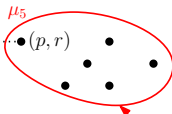
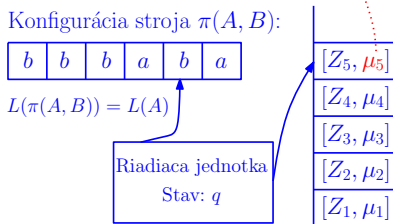
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:

$(p, r) \in K_A \times K_B$ je v μ_5 práve vtedy, keď $\exists w \in \Sigma^*$:

(i): $(p, w, Z_1 \dots Z_5) \vdash_A^* (p_F, \varepsilon, \alpha)$, kde $p_F \in F_A$ a $\alpha \in \Gamma^*$



$\rightsquigarrow$



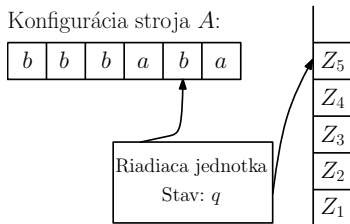
Predpovedajúce stroje

- Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

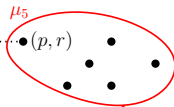
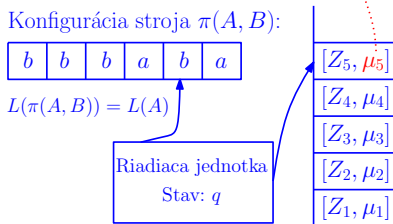
Predpovedajúci stroj (angl. *Predicting Machine*) $\pi(A, B)$:

$(p, r) \in K_A \times K_B$ je v μ_5 práve vtedy, keď $\exists w \in \Sigma^*$:

- (i): $(p, w, Z_1 \dots Z_5) \vdash_A^* (p_F, \varepsilon, \alpha)$, kde $p_F \in F_A$ a $\alpha \in \Gamma^*$
- (ii): $(r, w) \vdash_B^* (r_F, \varepsilon)$, kde $r_F \in F_B$



$\rightsquigarrow$



Konštrukcia predpovedajúceho stroja

Konštrukcia predpovedajúceho stroja

Označenie

Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA.

Konštrukcia predpovedajúceho stroja

Označenie

Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA.

- ▶ *Pre $p \in K_A$ a $Z \in \Gamma$ píšeme $A_{p,Z} = (K_A, \Sigma, \Gamma, \delta_A, p, Z, F_A)$.*

Konštrukcia predpovedajúceho stroja

Označenie

Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA.

- ▶ *Pre $p \in K_A$ a $Z \in \Gamma$ píšeme $A_{p,Z} = (K_A, \Sigma, \Gamma, \delta_A, p, Z, F_A)$.*
- ▶ $N_p(A) = \{w \in \Sigma^* \mid (p_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)\}.$

Konštrukcia predpovedajúceho stroja

Označenie

Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA.

- ▶ Pre $p \in K_A$ a $Z \in \Gamma$ píšeme $A_{p,Z} = (K_A, \Sigma, \Gamma, \delta_A, p, Z, F_A)$.
- ▶ $N_p(A) = \{w \in \Sigma^* \mid (p_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)\}$.

Označenie

Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA.

Konštrukcia predpovedajúceho stroja

Označenie

Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA.

- ▶ Pre $p \in K_A$ a $Z \in \Gamma$ píšeme $A_{p,Z} = (K_A, \Sigma, \Gamma, \delta_A, p, Z, F_A)$.
- ▶ $N_p(A) = \{w \in \Sigma^* \mid (p_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)\}$.

Označenie

Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA.

- ▶ Pre $r \in K_B$ píšeme $B_r = (K_B, \Sigma, \delta_B, r, F_B)$.

Konštrukcia predpovedajúceho stroja

Označenie

Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA.

- ▶ Pre $p \in K_A$ a $Z \in \Gamma$ píšeme $A_{p,Z} = (K_A, \Sigma, \Gamma, \delta_A, p, Z, F_A)$.
- ▶ $N_p(A) = \{w \in \Sigma^* \mid (p_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)\}$.

Označenie

Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA.

- ▶ Pre $r \in K_B$ píšeme $B_r = (K_B, \Sigma, \delta_B, r, F_B)$.
- ▶ $L_r(B) = \{w \in \Sigma^* \mid (r_0, w) \vdash^* (r, \varepsilon)\}$.

Konštrukcia predpovedajúceho stroja

Konštrukcia predpovedajúceho stroja

- Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná
- ▶ $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, kde:

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná
- ▶ $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, kde:
 - ▶ $\mu_0 = \{(p, r) \in K_A \times K_B \mid L(A_{p, Z_0}) \cap L(B_r) \neq \emptyset\}$

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná
- ▶ $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, kde:
 - ▶ $\mu_0 = \{(p, r) \in K_A \times K_B \mid L(A_{p, Z_0}) \cap L(B_r) \neq \emptyset\}$

Pôvodný prechod		Nové prechody
$\delta_A(p, z, Z) = (p', \varepsilon)$	$\forall \mu \in 2^{K_A \times K_B}:$	$\delta(p, z, [Z, \mu]) = (p', \varepsilon)$
$\delta_A(p, z, Z) = (p', Z)$	$\forall \mu \in 2^{K_A \times K_B}:$	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu])$
$\delta_A(p, z, Z) = (p', ZY)$	$\forall \mu \in 2^{K_A \times K_B}:$	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu][Y, \eta])$

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná
- ▶ $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, kde:
 - ▶ $\mu_0 = \{(p, r) \in K_A \times K_B \mid L(A_{p, Z_0}) \cap L(B_r) \neq \emptyset\}$

Pôvodný prechod		Nové prechody
$\delta_A(p, z, Z) = (p', \varepsilon)$	$\forall \mu \in 2^{K_A \times K_B}:$	$\delta(p, z, [Z, \mu]) = (p', \varepsilon)$
$\delta_A(p, z, Z) = (p', Z)$	$\forall \mu \in 2^{K_A \times K_B}:$	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu])$
$\delta_A(p, z, Z) = (p', ZY)$	$\forall \mu \in 2^{K_A \times K_B}:$	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu][Y, \eta])$

- ▶ Množina η pozostáva z dvojíc $(\bar{p}, \bar{r}) \in K_A \times K_B$ takých, že:

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná
- ▶ $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, kde:
 - ▶ $\mu_0 = \{(p, r) \in K_A \times K_B \mid L(A_{p, Z_0}) \cap L(B_r) \neq \emptyset\}$

Pôvodný prechod		Nové prechody
$\delta_A(p, z, Z) = (p', \varepsilon)$	$\forall \mu \in 2^{K_A \times K_B}$:	$\delta(p, z, [Z, \mu]) = (p', \varepsilon)$
$\delta_A(p, z, Z) = (p', Z)$	$\forall \mu \in 2^{K_A \times K_B}$:	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu])$
$\delta_A(p, z, Z) = (p', ZY)$	$\forall \mu \in 2^{K_A \times K_B}$:	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu][Y, \eta])$

- ▶ Množina η pozostáva z dvojíc $(\bar{p}, \bar{r}) \in K_A \times K_B$ takých, že:
 - a) Bud' $L(A_{\bar{p}, Y}) \cap L(B_{\bar{r}}) \neq \emptyset$

Konštrukcia predpovedajúceho stroja

- ▶ Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je DPDA v n.t.
- ▶ Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je DKA
- ▶ Prázdnosť bezkontextových jazykov je rozhodnuteľná
- ▶ $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, kde:
 - ▶ $\mu_0 = \{(p, r) \in K_A \times K_B \mid L(A_{p, Z_0}) \cap L(B_r) \neq \emptyset\}$

Pôvodný prechod		Nové prechody
$\delta_A(p, z, Z) = (p', \varepsilon)$	$\forall \mu \in 2^{K_A \times K_B}$:	$\delta(p, z, [Z, \mu]) = (p', \varepsilon)$
$\delta_A(p, z, Z) = (p', Z)$	$\forall \mu \in 2^{K_A \times K_B}$:	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu])$
$\delta_A(p, z, Z) = (p', ZY)$	$\forall \mu \in 2^{K_A \times K_B}$:	$\delta(p, z, [Z, \mu]) = (p', [Z, \mu][Y, \eta])$

- ▶ Množina η pozostáva z dvojíc $(\bar{p}, \bar{r}) \in K_A \times K_B$ takých, že:
 - a) Bud' $L(A_{\bar{p}, Y}) \cap L(B_{\bar{r}}) \neq \emptyset$
 - b) Alebo pre nejaké $(s, t) \in \mu$ je $N_s(A_{\bar{p}, Y}) \cap L_t(B_{\bar{r}}) \neq \emptyset$

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Operácie MIN a MAX

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

► $w \in L$

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$
- ▶ Žiaden *vlastný* prefix slova w nie je v L

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$
- ▶ Žiaden *vlastný* prefix slova w nie je v L

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MAX(L)$ je definovaný nasledovne:*

$$MAX(L) = \{w \in L \mid \forall u \in \Sigma^* : wu \in L \Rightarrow u = \varepsilon\}.$$

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$
- ▶ Žiaden *vlastný* prefix slova w nie je v L

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MAX(L)$ je definovaný nasledovne:*

$$MAX(L) = \{w \in L \mid \forall u \in \Sigma^* : wu \in L \Rightarrow u = \varepsilon\}.$$

Jazyk $MAX(L)$ teda obsahuje slová w také, že:

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:*

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$
- ▶ Žiaden *vlastný* prefix slova w nie je v L

Definícia

Nech $L \subseteq \Sigma^$ je jazyk. Jazyk $MAX(L)$ je definovaný nasledovne:*

$$MAX(L) = \{w \in L \mid \forall u \in \Sigma^* : wu \in L \Rightarrow u = \varepsilon\}.$$

Jazyk $MAX(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$

Operácie MIN a MAX

Definícia

Nech $L \subseteq \Sigma^*$ je jazyk. Jazyk $MIN(L)$ je definovaný nasledovne:

$$MIN(L) = \{w \in L \mid \forall u, v \in \Sigma^* : (w = uv \wedge u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk $MIN(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$
- ▶ Žiaden *vlastný* prefix slova w nie je v L

Definícia

Nech $L \subseteq \Sigma^*$ je jazyk. Jazyk $MAX(L)$ je definovaný nasledovne:

$$MAX(L) = \{w \in L \mid \forall u \in \Sigma^* : wu \in L \Rightarrow u = \varepsilon\}.$$

Jazyk $MAX(L)$ teda obsahuje slová w také, že:

- ▶ $w \in L$
- ▶ Slovo w nie je *vlastným* prefixom žiadneho slova z L

Trieda $\mathcal{L}_{detCF}$ a operácia MIN

Trieda $\mathcal{L}_{detCF}$ a operácia MIN

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MIN.

Trieda $\mathcal{L}_{detCF}$ a operácia MIN

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MIN.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a operácia MIN

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MIN.

Dôkaz.

Stačí zrušiť všetky prechody z akceptačných stavov.



Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Dôkaz.

Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Dôkaz.

- ▶ Nech A je DPDA akceptujúci jazyk L

Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Dôkaz.

- ▶ Nech A je DPDA akceptujúci jazyk L
- ▶ Nech B je DKA akceptujúci jazyk Σ^+ :

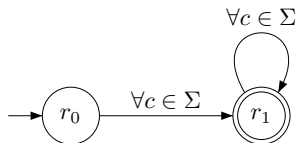
Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Dôkaz.

- ▶ Nech A je DPDA akceptujúci jazyk L
- ▶ Nech B je DKA akceptujúci jazyk Σ^+ :



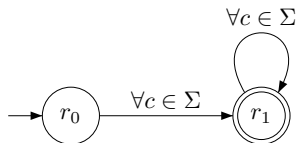
Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Dôkaz.

- ▶ Nech A je DPDA akceptujúci jazyk L
- ▶ Nech B je DKA akceptujúci jazyk Σ^+ :



- ▶ Nech $\pi(A, B)$ je predpovedajúci stroj

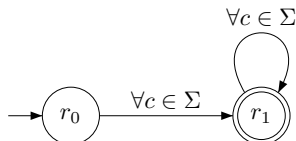
Trieda $\mathcal{L}_{detCF}$ a operácia MAX

Veta

Trieda $\mathcal{L}_{detCF}$ je uzavretá na operáciu MAX.

Dôkaz.

- ▶ Nech A je DPDA akceptujúci jazyk L
- ▶ Nech B je DKA akceptujúci jazyk Σ^+ :



- ▶ Nech $\pi(A, B)$ je predpovedajúci stroj
- ▶ Ten stačí upraviť tak, aby mohol akceptovať iba ak je v stave $p \in F$ a súčasne je zásobník buď prázdny, alebo je na jeho vrchu $[Z, \mu]$ také, že $(p, r_0) \notin \mu$



Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- Sporom, nech L je v $\mathcal{L}_{detCF}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$
- ▶ Ak $0 < i \leq j$, je v L_3 prefix $(ab)^i(ba)^i$ tohto slova

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$
- ▶ Ak $0 < i \leq j$, je v L_3 prefix $(ab)^i(ba)^i$ tohto slova
- ▶ Ak $i > j$, nie je v L_3 žiaden vlastný prefix $(ab)^i(ba)^j(ab)^j(ba)^i$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$
- ▶ Ak $0 < i \leq j$, je v L_3 prefix $(ab)^i(ba)^i$ tohto slova
- ▶ Ak $i > j$, nie je v L_3 žiaden vlastný prefix $(ab)^i(ba)^j(ab)^j(ba)^i$
- ▶ Preto $L_4 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; 0 \leq j < i\}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$
- ▶ Ak $0 < i \leq j$, je v L_3 prefix $(ab)^i(ba)^i$ tohto slova
- ▶ Ak $i > j$, nie je v L_3 žiaden vlastný prefix $(ab)^i(ba)^j(ab)^j(ba)^i$
- ▶ Preto $L_4 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; 0 \leq j < i\}$
- ▶ Nech $h(a) = ab$ a $h(b) = ba$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$
- ▶ Ak $0 < i \leq j$, je v L_3 prefix $(ab)^i(ba)^i$ tohto slova
- ▶ Ak $i > j$, nie je v L_3 žiaden vlastný prefix $(ab)^i(ba)^j(ab)^j(ba)^i$
- ▶ Preto $L_4 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; 0 \leq j < i\}$
- ▶ Nech $h(a) = ab$ a $h(b) = ba$
- ▶ $L_5 := h^{-1}(L_4) = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}; 0 \leq j < i\}$ je v $\mathcal{L}_{detCF}$

Príklad jazyka z $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$

Tvrdenie

Jazyk $L = \{ww^R \mid w \in \{a, b\}^*\}$ je v $\mathcal{L}_{CF} - \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Sporom, nech L je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a, b\}^+\}$ je v $\mathcal{L}_{detCF}$
- ▶ Potom aj $L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^*$ je v $\mathcal{L}_{detCF}$
- ▶ Zrejme $L_3 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; i > 0 \vee j > 0\}$
- ▶ Jazyk $L_4 := \text{MIN}(L_3) \cap a\{a, b\}^*$ je tiež v $\mathcal{L}_{detCF}$
- ▶ Ak $i = 0$ a $j > 0$, slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ nie je v $a\{a, b\}^*$
- ▶ Ak $0 < i \leq j$, je v L_3 prefix $(ab)^i(ba)^i$ tohto slova
- ▶ Ak $i > j$, nie je v L_3 žiaden vlastný prefix $(ab)^i(ba)^j(ab)^j(ba)^i$
- ▶ Preto $L_4 = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i, j \in \mathbb{N}; 0 \leq j < i\}$
- ▶ Nech $h(a) = ab$ a $h(b) = ba$
- ▶ $L_5 := h^{-1}(L_4) = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}; 0 \leq j < i\}$ je v $\mathcal{L}_{detCF}$
- ▶ Spor, pretože L_5 nie je ani len v $\mathcal{L}_{CF}$ (pumpovacia lema) $\square$

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R		R
$L(A)$ je konečný	R		R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R		N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno		N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R		R
$L(A)$ je konečný	R		R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R		N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno		N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{\text{detCF}}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{\text{detCF}}$	$\mathcal{L}_{\text{CF}}$
$L(A)$ je $\emptyset$	R		R
$L(A)$ je konečný	R		R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R		N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^c$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{\text{detCF}}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{\text{detCF}}$	$\mathcal{L}_{\text{CF}}$
$L(A)$ je $\emptyset$	R		R
$L(A)$ je konečný	R		R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R		N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^c$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R		N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R		N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Problém ekvivalencie s regulárnym jazykom R

Problém ekvivalence s regulárnym jazykom R

Veta

Nasledujúci problém je rozhodnuteľný:

Vstup: DPDA A , DKA B .

Výstup: „Áno“ práve vtedy, keď $L(A) = L(B)$.

Problém ekvivalencie s regulárnym jazykom R

Veta

Nasledujúci problém je rozhodnuteľný:

Vstup: DPDA A , DKA B .

Výstup: „Áno“ práve vtedy, keď $L(A) = L(B)$.

Dôkaz.

Problém ekvivalencie s regulárnym jazykom R

Veta

Nasledujúci problém je rozhodnuteľný:

Vstup: DPDA A , DKA B .

Výstup: „Áno“ práve vtedy, keď $L(A) = L(B)$.

Dôkaz.

$$\blacktriangleright L(A) = L(B) \iff (L(A) \cap L(B)^c) \cup (L(A)^c \cap L(B)) = \emptyset$$

Problém ekvivalencie s regulárnym jazykom R

Veta

Nasledujúci problém je rozhodnuteľný:

Vstup: DPDA A , DKA B .

Výstup: „Áno“ práve vtedy, keď $L(A) = L(B)$.

Dôkaz.

- ▶ $L(A) = L(B) \iff (L(A) \cap L(B)^c) \cup (L(A)^c \cap L(B)) = \emptyset$
- ▶ Trieda $\mathcal{L}_{detCF}$ je (efektívne) uzavretá na komplement a prienik s regulárnym jazykom

Problém ekvivalencie s regulárnym jazykom R

Veta

Nasledujúci problém je rozhodnuteľný:

Vstup: DPDA A , DKA B .

Výstup: „Áno“ práve vtedy, keď $L(A) = L(B)$.

Dôkaz.

- ▶ $L(A) = L(B) \iff (L(A) \cap L(B)^c) \cup (L(A)^c \cap L(B)) = \emptyset$
- ▶ Trieda $\mathcal{L}_{detCF}$ je (efektívne) uzavretá na komplement a prienik s regulárnym jazykom
- ▶ Trieda $\mathcal{L}_{CF}$ je (efektívne) uzavretá na zjednotenie

Problém ekvivalencie s regulárnym jazykom R

Veta

Nasledujúci problém je rozhodnuteľný:

Vstup: DPDA A , DKA B .

Výstup: „Áno“ práve vtedy, keď $L(A) = L(B)$.

Dôkaz.

- ▶ $L(A) = L(B) \iff (L(A) \cap L(B)^c) \cup (L(A)^c \cap L(B)) = \emptyset$
- ▶ Trieda $\mathcal{L}_{detCF}$ je (efektívne) uzavretá na komplement a prienik s regulárnym jazykom
- ▶ Trieda $\mathcal{L}_{CF}$ je (efektívne) uzavretá na zjednotenie
- ▶ Prázdnosť je pre *bezkontextové* jazyky rozhodnuteľná



Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno		N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R		N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Jazyk platných výpočtov Turingovho stroja

Jazyk platných výpočtov Turingovho stroja

Definícia

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk platných výpočtov $L_{PV}(A)$ stroja A pozostáva zo slov

$$\begin{cases} w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k, & k \text{ párne} \\ w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

Jazyk platných výpočtov Turingovho stroja

Definícia

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk platných výpočtov $L_{PV}(A)$ stroja A pozostáva zo slov

$$\begin{cases} w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k, & k \text{ párne} \\ w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

(i) Každé w_i , $i = 0, \dots, k$, je konfigurácia stroja A

Jazyk platných výpočtov Turingovho stroja

Definícia

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk platných výpočtov $L_{PV}(A)$ stroja A pozostáva zo slov

$$\begin{cases} w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k, & k \text{ párne} \\ w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

- (i) Každé w_i , $i = 0, \dots, k$, je konfigurácia stroja A*
- (ii) Konfigurácia w_0 je počiatočná*

Jazyk platných výpočtov Turingovho stroja

Definícia

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk platných výpočtov $L_{PV}(A)$ stroja A pozostáva zo slov

$$\begin{cases} w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k, & k \text{ párne} \\ w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

- (i) Každé w_i , $i = 0, \dots, k$, je konfigurácia stroja A*
- (ii) Konfigurácia w_0 je počiatočná*
- (iii) Konfigurácia w_k je akceptačná*

Jazyk platných výpočtov Turingovho stroja

Definícia

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk platných výpočtov $L_{PV}(A)$ stroja A pozostáva zo slov

$$\begin{cases} w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k, & k \text{ párne} \\ w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

- (i) Každé w_i , $i = 0, \dots, k$, je konfigurácia stroja A*
- (ii) Konfigurácia w_0 je počiatočná*
- (iii) Konfigurácia w_k je akceptačná*
- (iv) Pre $i = 1, \dots, k$ je $w_{i-1} \vdash w_i$*

Jazyk platných výpočtov Turingovho stroja

Definícia

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk platných výpočtov $L_{PV}(A)$ stroja A pozostáva zo slov

$$\begin{cases} w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k, & k \text{ párne} \\ w_0 \# w_1^R \# w_2 \# w_3^R \# \dots \# w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

- (i) Každé w_i , $i = 0, \dots, k$, je konfigurácia stroja A*
- (ii) Konfigurácia w_0 je počiatočná*
- (iii) Konfigurácia w_k je akceptačná*
- (iv) Pre $i = 1, \dots, k$ je $w_{i-1} \vdash w_i$*

Lema

Nech A je deterministický Turingov stroj. Potom existujú jazyky $L_1, L_2 \in \mathcal{L}_{\text{detCF}}$ také, že $L_{PV}(A) = L_1 \cap L_2$.

Problém prázdnoti prieniku

Problém prázdnoty prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Problém prázdnoty prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Dôkaz.

Problém prázdnoti prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Dôkaz.

- Problém prázdnoti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný

Problém prázdnoty prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Dôkaz.

- ▶ Problém prázdnoty je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj

Problém prázdnoty prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Dôkaz.

- ▶ Problém prázdnoty je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj
- ▶ Nech A_1, A_2 sú DPDA také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$

Problém prázdnoti prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Dôkaz.

- ▶ Problém prázdnoti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj
- ▶ Nech A_1, A_2 sú DPDA také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$
- ▶ Zrejme $L_{PV}(B)$ je prázdny práve vtedy, keď $L(B)$ je prázdny

Problém prázdnoti prieniku

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$.

Dôkaz.

- ▶ Problém prázdnoti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj
- ▶ Nech A_1, A_2 sú DPDA také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$
- ▶ Zrejme $L_{PV}(B)$ je prázdny práve vtedy, keď $L(B)$ je prázdny
- ▶ $L(B)$ je prázdny práve vtedy, keď $L(A_1) \cap L(A_2) = \emptyset$ □

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R		N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Problém inklúzie

Problém inklúzie

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \subseteq L(A_2)$.

Problém inklúzie

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \subseteq L(A_2)$.

Dôkaz.

Problém inklúzie

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \subseteq L(A_2)$.

Dôkaz.

- ▶ Redukciou problému prázdnoty prieniku na problém inklúzie

Problém inklúzie

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \subseteq L(A_2)$.

Dôkaz.

- ▶ Redukciou problému prázdnoty prieniku na problém inklúzie
- ▶ Zrejme $L_1 \cap L_2 = \emptyset$ práve vtedy, keď $L_1 \subseteq L_2^C$

Problém inklúzie

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \subseteq L(A_2)$.

Dôkaz.

- ▶ Redukciou problému prázdnoty prieniku na problém inklúzie
- ▶ Zrejme $L_1 \cap L_2 = \emptyset$ práve vtedy, keď $L_1 \subseteq L_2^C$
- ▶ Trieda $\mathcal{L}_{detCF}$ je (efektívne) uzavretá na komplement



Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R	N	N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R	N	N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno		N
$L(A)^C$ je rovnakého typu	áno	áno	N

Viac o jazyku platných výpočtov

Viac o jazyku platných výpočtov

Lema

Nech A je deterministický Turingov stroj, ktorý na každom vstupe urobí aspoň tri kroky. Potom jazyk $L_{PV}(A)$:

Viac o jazyku platných výpočtov

Lema

Nech A je deterministický Turingov stroj, ktorý na každom vstupe urobí aspoň tri kroky. Potom jazyk $L_{PV}(A)$:

- ▶ *Je konečný, ak $L(A)$ je konečný*

Viac o jazyku platných výpočtov

Lema

Nech A je deterministický Turingov stroj, ktorý na každom vstupe urobí aspoň tri kroky. Potom jazyk $L_{PV}(A)$:

- ▶ *Je konečný, ak $L(A)$ je konečný*
- ▶ *Inak nie je bezkontextový*

Problém příslušnosti přieniku do $\mathcal{L}_{detCF}$

Problém príslušnosti prieniku do $\mathcal{L}_{detCF}$

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$.

Problém príslušnosti prieniku do $\mathcal{L}_{detCF}$

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$.

Dôkaz.

Problém príslušnosti prieniku do $\mathcal{L}_{detCF}$

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$.

Dôkaz.

- Problém konečnosti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný

Problém príslušnosti prieniku do $\mathcal{L}_{detCF}$

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Problém konečnosti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj

Problém príslušnosti prieniku do $\mathcal{L}_{detCF}$

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Problém konečnosti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj
- ▶ Nech A_1, A_2 sú DPDA také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$

Problém príslušnosti prieniku do $\mathcal{L}_{detCF}$

Veta

Nasledujúci problém je nerozhodnuteľný:

Vstup: DPDA A_1 , DPDA A_2 .

Výstup: „Áno“ práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$.

Dôkaz.

- ▶ Problém konečnosti je pre rekurzívne vyčísliteľné jazyky nerozhodnuteľný
- ▶ Nech B je deterministický Turingov stroj
- ▶ Nech A_1, A_2 sú DPDA také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$
- ▶ $L(B)$ je konečný práve vtedy, keď $L(A_1) \cap L(A_2) \in \mathcal{L}_{detCF}$ □

Dva příbuzné problémy

Dva príbuzné problémy

Dôsledok

Nasledujúce problémy sú pre deterministické bezkontextové jazyky nerozhodnuteľné:

Dva príbuzné problémy

Dôsledok

Nasledujúce problémy sú pre deterministické bezkontextové jazyky nerozhodnuteľné:

- ▶ Pre $L_1, L_2 \in \mathcal{L}_{\text{detCF}}$ zistiť, či $L_1 \cap L_2 \in \mathcal{L}_{\text{CF}}$

Dva príbuzné problémy

Dôsledok

Nasledujúce problémy sú pre deterministické bezkontextové jazyky nerozhodnuteľné:

- ▶ Pre $L_1, L_2 \in \mathcal{L}_{\text{detCF}}$ zistiť, či $L_1 \cap L_2 \in \mathcal{L}_{\text{CF}}$
- ▶ Pre $L_1, L_2 \in \mathcal{L}_{\text{detCF}}$ zistiť, či $L_1 \cup L_2 \in \mathcal{L}_{\text{detCF}}$

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R	N	N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno	N	N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R		N
$L(A_1) \subseteq L(A_2)$	R	N	N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R^*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno	N	N
$L(A)^C$ je rovnakého typu	áno	áno	N

Problém ekvivalence

Problém ekvivalence

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)

Problém ekvivalence

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)
- ▶ Množstvo publikovaných výsledkov pre špeciálne prípady DPDA

Problém ekvivalence

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)
- ▶ Množstvo publikovaných výsledkov pre špeciálne prípady DPDA
- ▶ Ekvivalentný s niektorými problémami ekvivalence programových schém

Problém ekvivalencie

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)
- ▶ Množstvo publikovaných výsledkov pre špeciálne prípady DPDA
- ▶ Ekvivalentný s niektorými problémami ekvivalencie programových schém
- ▶ Mejtus 1992: problém ekvivalencie *je rozhodnuteľný*

Problém ekvivalencie

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)
- ▶ Množstvo publikovaných výsledkov pre špeciálne prípady DPDA
- ▶ Ekvivalentný s niektorými problémami ekvivalencie programových schém
- ▶ Mejtus 1992: problém ekvivalencie *je rozhodnuteľný*
- ▶ Sénizergues 1997/2001: prvý naozajstný dôkaz (166 strán + Gödelova cena 2002)

Problém ekvivalencie

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)
- ▶ Množstvo publikovaných výsledkov pre špeciálne prípady DPDA
- ▶ Ekvivalentný s niektorými problémami ekvivalencie programových schém
- ▶ Mejtus 1992: problém ekvivalencie *je rozhodnuteľný*
- ▶ Sénizergues 1997/2001: prvý naozajstný dôkaz (166 strán + Gödelova cena 2002)
- ▶ Neskôr rôzne zjednodušenia (Stirling, Sénizergues)

Problém ekvivalencie

- ▶ Niekdajší dlhoročný otvorený problém (od roku 1966)
- ▶ Množstvo publikovaných výsledkov pre špeciálne prípady DPDA
- ▶ Ekvivalentný s niektorými problémami ekvivalencie programových schém
- ▶ Mejtus 1992: problém ekvivalencie *je rozhodnuteľný*
- ▶ Sénizergues 1997/2001: prvý naozajstný dôkaz (166 strán + Gödelova cena 2002)
- ▶ Neskôr rôzne zjednodušenia (Stirling, Sénizergues)
- ▶ Jančar 2010/2012: „krátky“ dôkaz

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R	R*	N
$L(A_1) \subseteq L(A_2)$	R	N	N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno	N	N
$L(A)^C$ je rovnakého typu	áno	áno	N

Rozhodovacie problémy pre jazyky z $\mathcal{L}_{detCF}$

Veta

Nasledujúce vlastnosti deterministických bezkontextových jazykov (alebo ich dvojíc) sú rozhodnuteľné resp. nerozhodnuteľné (podľa tabuľky):

	$\mathcal{R}$	$\mathcal{L}_{detCF}$	$\mathcal{L}_{CF}$
$L(A)$ je $\emptyset$	R	R	R
$L(A)$ je konečný	R	R	R
$L(A_1) = L(A_2)$	R	R*	N
$L(A_1) \subseteq L(A_2)$	R	N	N
$L(A_1) \cap L(A_2) = \emptyset$	R	N	N
$L(A) = R$ pre $R \in \mathcal{R}$	R	R	N
$L(A) \in \mathcal{R}$	áno	R*	N
$L(A_1) \cap L(A_2)$ je rovnakého typu	áno	N	N
$L(A)^C$ je rovnakého typu	áno	áno	N

Deterministické zásobníkové automaty

Definícia

Uzavretosť triedy $\mathcal{L}_{detCF}$ na komplement

Vzťah medzi $\mathcal{L}_{detCF}$ a $\mathcal{L}_{CF}$

Základné uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Akceptácia prázdny zásobníkom

Normálny tvar deterministických zásobníkových automatov

Predpovedajúce stroje

Ďalšie uzáverové vlastnosti triedy $\mathcal{L}_{detCF}$

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Záver

Veci, o ktorých sme nehovorili

Veci, o ktorých sme nehovorili

- ▶ DPDA akceptujúce prázdny zásobník sú ekvivalentné LR(0) gramatikám

Veci, o ktorých sme nehovorili

- ▶ DPDA akceptujúce prázdny zásobníkom sú ekvivalentné LR(0) gramatikám
- ▶ DPDA akceptujúce stavom sú ekvivalentné LR(k) gramatikám $\forall k \geq 1$

Veci, o ktorých sme nehovorili

- ▶ DPDA akceptujúce prázdny zásobník sú ekvivalentné LR(0) gramatikám
- ▶ DPDA akceptujúce stavom sú ekvivalentné LR(k) gramatikám $\forall k \geq 1$
- ▶ DPDA vo všeobecnosti nemožno „odepsilonovať“