

Příklady otázek – Automaty a jazyky

Reálné otázky ze SZZ 2023–2026 (I1)

Jakub Klesa

Pokryté předměty: NTIN071 (Automaty a gramatiky)

Obsah

1	Převod bezkontextové gramatiky na automat	1
2	Zásobníkový automat	2
3	Reprezentace bezkontextového jazyka	3
4	Doplňek regulárního výrazu	4
5	Průnik bezkontextového a regulárního jazyka	5
6	Deterministické konečné automaty	7
7	Porovnání modulárních hashů	8

Poznámka. Tento dokument obsahuje reálné otázky z minulých termínů SZZ (2023–2026) zařazené do okruhu I1. U otázek s obrázkem je grafika oříznutá z originálního PDF (viz odkaz na zdroj). Text, číselná data a tabulky jsou přepsané věrně. Zadání i oficiální náčrty řešení pocházejí ze sad zveřejněných na webu MFF UK:

<https://www.mff.cuni.cz/cs/studenti/bakalarske-studium/statni-zaverecne-zkousky/bakalarske-statni-zkousky-studijniho-programu-informatika>

Náčrty označené jako doplněné jsou vlastní, oficiální sada je k dané otázce neobsahuje.

1 Převod bezkontextové gramatiky na automat

(Reálná otázka: léto 2023, č. 2 – Převod bezkontextové gramatiky na automat ($CFG \rightarrow$ zásobníkový automat))

Zadání.

1. Uveďte definici *bezkontextové gramatiky*.
2. Sestrojte bezkontextovou gramatiku G generující následující jazyk:

$$L = \{a^i b^j \mid i, j \geq 0 \text{ a platí } 2i = 3j\}.$$

3. Převeďte gramatiku G z předchozí části na zásobníkový automat přijímající prázdným zásobníkem. (Pokud se vám nepodařilo gramatiku sestavit, sestrojte libovolný zásobníkový automat přijímající jazyk L prázdným zásobníkem.)

Náčrt řešení.

(V PDF bez náčrtu řešení; následující řešení je doplněné a numericky ověřené.)

1. *Bezkontextová gramatika* je čtveřice $G = (V, T, \mathcal{P}, S)$, kde V je konečná množina neterminálů (proměnných), T konečná neprázdná množina terminálů ($V \cap T = \emptyset$), $S \in V$ počáteční symbol a $\mathcal{P}$ konečná množina pravidel tvaru $A \rightarrow \omega$, kde $A \in V$ a $\omega \in (V \cup T)^*$.

2. Z podmínky $2i = 3j$ plyne $i = 3k$, $j = 2k$ pro $k \geq 0$, tedy $L = \{a^{3k}b^{2k} \mid k \geq 0\}$. Generuje ho

$$G = (\{S\}, \{a, b\}, \mathcal{P}, S), \quad \mathcal{P} = \{S \rightarrow aaaSbb \mid \lambda\}.$$

(Každé použití prvního pravidla přidá tři a zleva a dvě b zprava; ukončení pravidlem $S \rightarrow \lambda$ dá $a^{3k}b^{2k}$.)

3. Standardní převod gramatiky na jednostavový zásobníkový automat přijímající prázdným zásobníkem: $P = (\{q_0\}, \{a, b\}, \{a, b, S\}, \delta, q_0, S)$, kde

$$\delta(q_0, \lambda, S) = \{(q_0, aaaSbb), (q_0, \lambda)\}, \quad \delta(q_0, a, a) = \{(q_0, \lambda)\}, \quad \delta(q_0, b, b) = \{(q_0, \lambda)\}.$$

Pravidla pro S rozepisují neterminál na vrcholu zásobníku, pravidla pro a, b párují terminál na vrcholu se čteným vstupním symbolem. Automat přijme prázdným zásobníkem právě slova generovaná G .

2 Zásobníkový automat

(Reálná otázka: podzim 2023, č. 2 – Zásobníkový automat (definice + konstrukce PDA))

Zadání.

1. Uveďte definici *zásobníkového automatu*.
2. Sestrojte zásobníkový automat A , který přijímá právě řetězce $w \in \{a, b\}^*$, které mají více znaků a než b a zároveň počet a je lichý, tj.

$$L(A) = \{w \mid w \in \{a, b\}^* \ \& \ |w|_a > |w|_b \ \& \ (\exists k \in \mathbb{N}_0) |w|_a = 2k + 1\}.$$

Za částečné řešení se uznává i automat rozpoznávající „více a než b “.

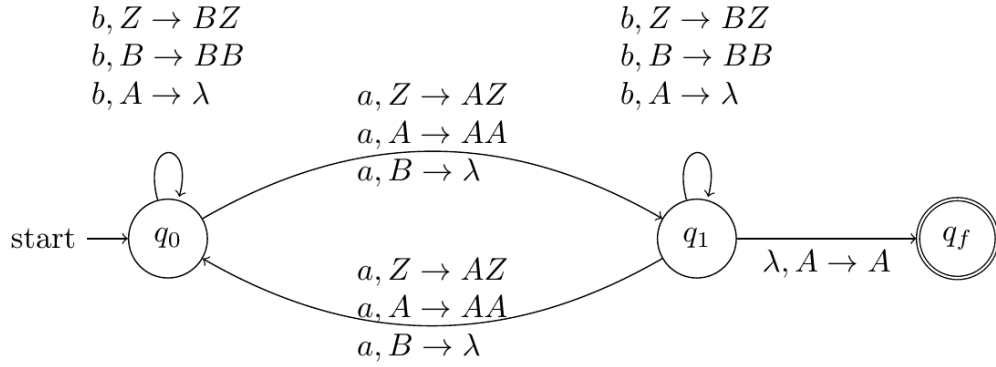
Náčrt řešení.

1. *Zásobníkový automat* je sedmice $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- Q je konečná množina stavů,
- Σ je konečná vstupní abeceda,
- Γ je konečná zásobníková abeceda,
- δ je přechodová funkce $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow \mathcal{P}_{FIN}(Q \times \Gamma^*)$,
- $q_0 \in Q$ je počáteční stav,
- $Z_0 \in \Gamma$ je počáteční zásobníkový symbol,
- $F \subseteq Q$ je množina přijímacích stavů.

V případě přijímání prázdným zásobníkem bez množiny přijímacích stavů.

2. Úkol řeší například zásobníkový automat A se třemi stavy: přijímacím q_f a q_0, q_1 rozlišujícími sudý a lichý počet a . Přebytné znaky a resp. b si ukládáme na zásobník, adekvátně odebíráme. Formálně: $A = (\{q_0, q_1, q_f\}, \{a, b\}, \{Z, A, B\}, \delta, q_0, Z, \{q_f\})$, s přechodovou funkcí δ popsanou grafem



3 Reprezentace bezkontextového jazyka

(Reálná otázka: jaro 2024, č. 1 – Reprezentace bezkontextového jazyka (CFG, zásobníkový automat))

Zadání.

Uvažme následující jazyk nad abecedou $\{0, 1, \#\}$:

$$L = \{w\#s^R \mid w, s \in \{0, 1\}^* \text{ a slovo } s \text{ je podslovem slova } w\}.$$

(Poznámka: s^R označuje slovo s napsané pozpátku; podslovo je souvislý podřetězec, vč. prázdného a celého slova.)

1. Uveďte formální definici bezkontextové gramatiky a formální definici zásobníkového automatu.
2. Sestrojte nějakou bezkontextovou gramatiku generující jazyk L .
3. Sestrojte nějaký zásobníkový automat přijímající jazyk L .

Náčrt řešení.

1. *Bezkontextová gramatika* je $G = (V, T, \mathcal{P}, S)$, kde V a T jsou konečné neprázdné disjunktní množiny, $S \in V$, a $\mathcal{P}$ je konečná množina přepisovacích pravidel tvaru $H \rightarrow \beta$, kde $H \in V$ a $\beta \in (V \cup T)^*$.

Zásobníkový automat je $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde Q , Σ a Γ jsou konečné neprázdné množiny, $\delta: Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow P_{FIN}(Q \times \Gamma^*)$, $q_0 \in Q$, $Z_0 \in \Gamma$, a $F \subseteq Q$ (P_{FIN} značí konečné podmnožiny; F lze vynechat, přijímáme-li prázdným zásobníkem).

2. Jazyk L generuje například bezkontextová gramatika $G = (\{S, X, A\}, \{0, 1, \#\}, \mathcal{P}, S)$ s pravidly $\mathcal{P} = \{S \rightarrow AX, X \rightarrow 0X0 \mid 1X1 \mid A\#, A \rightarrow 0A \mid 1A \mid \lambda\}$ (kde proměnná A generuje libovolné slovo nad $\{0, 1\}$).

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Každá derivace má tvar $S \Rightarrow AX \Rightarrow^* pX \Rightarrow^* puX u^R \Rightarrow^* puA\# u^R \Rightarrow^* puq\# u^R$, kde $p, q, u \in \{0, 1\}^*$ jsou libovolná. Vygenerované slovo je tedy $w\#s^R$ s $w = puq$ a $s = u$, tj. s je podslovem w . Naopak každé slovo z L má takový rozklad $w = puq$, takže ho gramatika vygeneruje.

3. Lze sestrojit převodem gramatiky G . Výsledkem je zásobníkový automat $(\{q_0\}, \{0, 1, \#\}, \{0, 1, \#, S, X, A\}, \delta, q_0, S)$ přijímající jazyk L prázdným zásobníkem, kde přechodová funkce sestává z přechodů odpovídajících přepisovacím pravidlům a z přechodů pro čtení písmen:

$$\delta = \{((q_0, \lambda, H), \beta) \mid H \rightarrow \beta \in \mathcal{P}\} \cup \{((q_0, a, a), \lambda) \mid a \in \{0, 1, \#\}\}.$$

Alternativně lze automat zkonstruovat přímo. Při čtení slova w si jeho znaky ukládáme na zásobník, po přečtení $\#$ nejprve smažeme libovolný počet znaků ze zásobníku, poté čteme s^R a kontrolujeme, zda souhlasí s písmeny na zásobníku, a následně vyprázdníme zásobník. Lze také sestavit automat přijímající koncovým stavem.

4 Doplněk regulárního výrazu

(Reálná otázka: podzim 2024, č. 1 – Doplněk regulárního výrazu (NFA/DFA, podmnožinová konstrukce, doplněk))

Zadání.

Mějme následující regulární výraz nad abecedou $\Sigma = \{a, b\}$:

$$R = ((a + b)(a + b))^* ab.$$

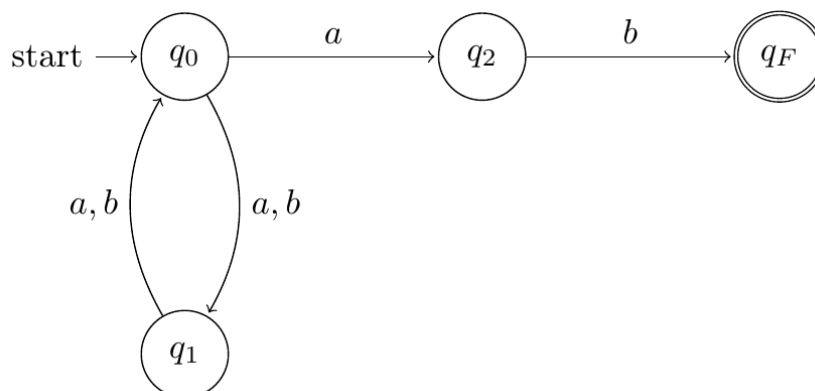
Označme jako L regulární jazyk popsany výrazem R .

1. Sestrojte (co nejmenší) *nedeterministický* konečný automat A rozpoznávající jazyk L .
2. Pomocí podmnožinové konstrukce převedte automat A na *deterministický* konečný automat B .
3. Z automatu B sestrojte *deterministický* konečný automat C rozpoznávající *doplněk* jazyka L .

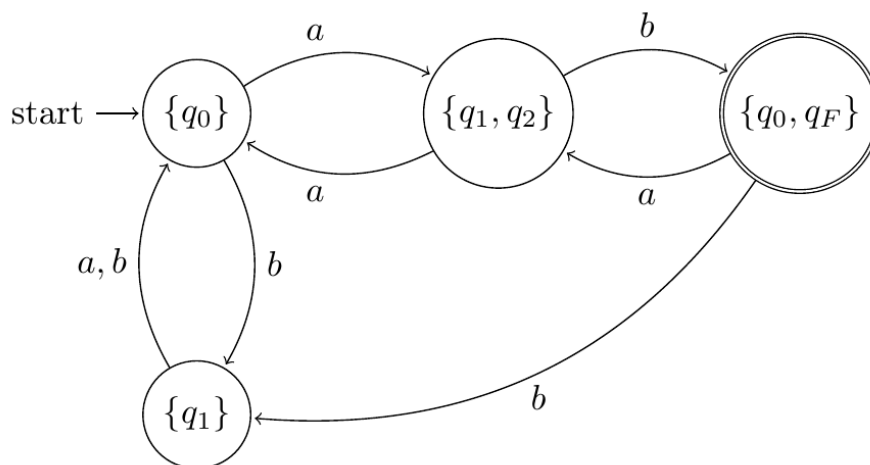
Sestrojené automaty A , B a C znázorněte pomocí stavových diagramů.

Náčrt řešení.

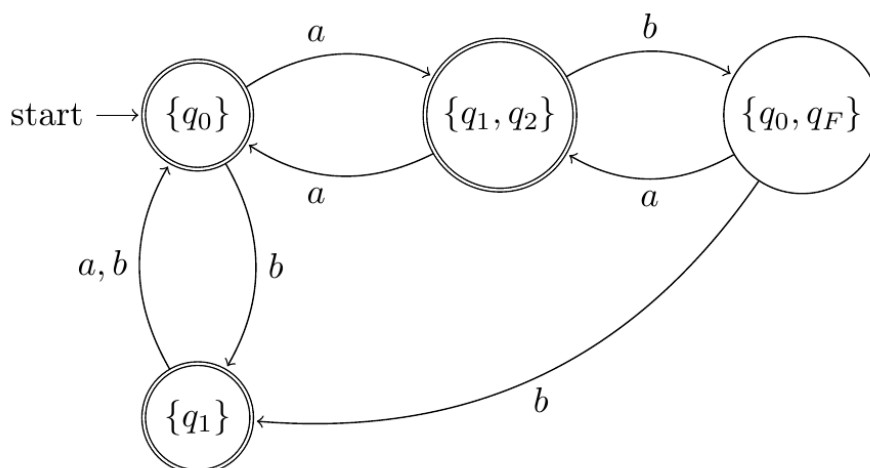
1. Nejmenší NFA má 4 stavy: $A = (\{q_0, q_1, q_2, q_F\}, \{a, b\}, \delta_A, q_0, \{q_F\})$, kde δ_A popíšeme stavovým diagramem:



2. Výsledkem podmnožinové konstrukce je následující DFA: $B = (\{\{q_0\}, \{q_1\}, \{q_1, q_2\}, \{q_0, q_F\}\}, \{a, b\}, \delta_B, q_0, \{\{q_0, q_F\}\})$, kde δ_B popíšeme stavovým diagramem:



3. Stačí zaměnit přijímající a nepřijímající stavy:



(Všimněte si, že přechodová funkce automatu B je totální, proto není třeba přidávat FAIL stav.)

Doplňující vysvětlení (nad rámec oficiálního náčrtu): L tvoří slova sudé délky končící na ab . Stav q_0 znamená „přečten sudý počet znaků“, q_1 „lichý počet“, z q_0 lze na a nedeterministicky zahájit závěrečné ab (přes q_2 do q_F). Méně než 4 stavy NFA mít nemůže: pro dvojice slov $(x_i, y_i) \in \{(\varepsilon, ab), (a, b), (ab, \varepsilon), (b, aab)\}$ je vždy $x_i y_i \in L$, ale pro $i \neq j$ do L nepatří $x_i y_j$ nebo $x_j y_i$. Stavy, ve kterých jsou přijímající výpočty slov $x_i y_i$ po přečtení x_i , proto musí být navzájem různé.

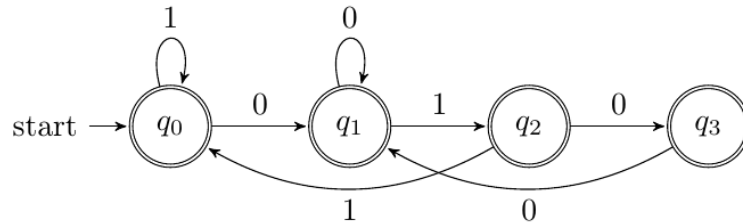
5 Průnik bezkontextového a regulárního jazyka

(Reálná otázka: jaro 2025, č. 1 – Průnik bezkontextového a regulárního jazyka (PDA, $CFG \cap DFA$))

Zadání.

1. Uveďte formální definici zásobníkového automatu.
2. Uvažme následující jazyky nad abecedou $\Sigma = \{0, 1\}$:

- L_1 je jazyk generovaný bezkontextovou gramatikou $G = (\{S\}, \Sigma, \mathcal{P}, S)$ s množinou pravidel $\mathcal{P} = \{S \rightarrow SS \mid 0S1 \mid \epsilon\}$ (kde ϵ značí prázdné slovo),
- L_2 je jazyk rozpoznávaný deterministickým konečným automatem $A = (\{q_0, q_1, q_2, q_3\}, \Sigma, \delta_A, q_0, \{q_0, q_1, q_2, q_3\})$, jehož přechodová funkce δ_A je dána následujícím stavovým diagramem:



Sestrojte zásobníkový automat rozpoznávající průnik $L = L_1 \cap L_2$ jazyků L_1 a L_2 .

Náčrt řešení.

1. Zásobníkový automat je sedmice $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- Q je konečná, neprázdná množina stavů,
- Σ je konečná, neprázdná množina znaků vstupní abecedy,
- Γ je konečná množina znaků zásobníkové abecedy,
- δ je přechodová funkce $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P_{FIN}(Q \times \Gamma^*)$, kde $P_{FIN}(\cdot)$ značí konečnou podmnožinu prvků,
- $q_0 \in Q$ je počáteční stav,
- $Z_0 \in \Gamma$ je počáteční zásobníkový symbol,
- $F \subseteq Q$ množina přijímajících stavů.

V případě přijímání prázdným zásobníkem lze vynechat množinu přijímajících stavů F .

2. Můžeme použít standardní konstrukci. Přijímání automatem A simulujeme pomocí stavů, generování gramatikou G pomocí zásobníku stejně jako při převodu gramatiky na jednostavový PDA. Výsledný zásobníkový automat, přijímající prázdným zásobníkem, je $P = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, S\}, \delta_P, q_0, S)$ kde δ_P je definována následovně:

$$\delta_P(q_i, a, a) = \{(\delta_A(q_i, a), \epsilon)\} \text{ pro } a \in \{0, 1\}, i \in \{0, 1, 2, 3\}$$

$$\delta_P(q_i, \epsilon, S) = \{(q_i, SS), (q_i, 0S1), (q_i, \epsilon)\} \text{ pro } i \in \{0, 1, 2, 3\}$$

První pravidlo představuje čtení vstupního symbolu automatem A a kontrolu s vrcholem zásobníku, druhé pravidlo simuluje jeden krok derivace gramatiky G .

(Alternativně lze přímo sestrojit zásobníkový automat rozpoznávající $L_1 \cap L_2$, nebo sestrojit bezkontextovou gramatiku generující $L_1 \cap L_2$ a následně ji převést na zásobníkový automat.)

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Stav q_1 znamená, že přečtené slovo končí na 0, stav q_2 na 01 a stav q_3 na 010. Ze stavu q_3 *nevede* přechod na 1, takže A na každém slově s podslovem 0101 uvízne. Protože jsou všechny stavy přijímající, je $L_2 = \{w \in \{0, 1\}^* \mid w \text{ neobsahuje } 0101\}$ a L tvoří vyvážená slova bez podslova 0101 (např. 0011 $\in L$, ale 0101 $\in L_1 \setminus L_2$). Automat P vyprázdní zásobník právě tehdy, když $S \Rightarrow^* w$, tj. $w \in L_1$; chybějící přechod $\delta_A(q_3, 1)$ se v P projeví tak, že P nemůže přečíst další znak a výpočet skončí neúspěchem. Každý stav, do kterého se P dostane, je přitom přijímající stav A , proto při přijímání prázdným zásobníkem kontrola koncového stavu není potřeba. Kdyby některý stav A přijímající nebyl, vložili bychom pod S nové dno Z_0 a dovolili ho odebrat přechodem $\delta_P(q, \epsilon, Z_0) = \{(q, \epsilon)\}$ jen ve stavech $q \in F_A$.

6 Deterministické konečné automaty

(Reálná otázka: léto 2025, č. 1 – Deterministické konečné automaty (jazyk DFA, rozšířená přechodová funkce))

Zadání.

1. Uveďte definici jazyka $L(A)$ přijímaného daným deterministickým konečným automatem $A = (Q, \Sigma, \delta, q_0, F)$ včetně definice jeho rozšířené přechodové funkce δ^* .
2. Necht $S \subseteq \Sigma^*$ je neprázdná konečná množina slov, Q je množina všech prefixů slov z S (včetně prázdného slova ε a slov z S) a $A = (Q, \Sigma, \delta, \varepsilon, S)$ je deterministický konečný automat s množinou stavů Q . Definujte jeho přechodovou funkci δ tak, aby přijímal jazyk

$$L(A) = \{w \in \Sigma^* \mid w \text{ obsahuje nějaké } s \in S \text{ (jako podslovo)}\}.$$

3. Sestrojte deterministický konečný automat přijímající jazyk

$$\{w \in \{0, 1\}^* \mid w \text{ obsahuje } 010 \text{ nebo končí na } \underline{10}\}.$$

Náčrt řešení.

1. $L(A) := \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$, kde $\delta^*: Q \times \Sigma^* \rightarrow Q$ je definována induktivně

$$\delta^*(q, \varepsilon) := q,$$

$$\delta^*(q, wx) := \delta(\delta^*(q, w), x)$$

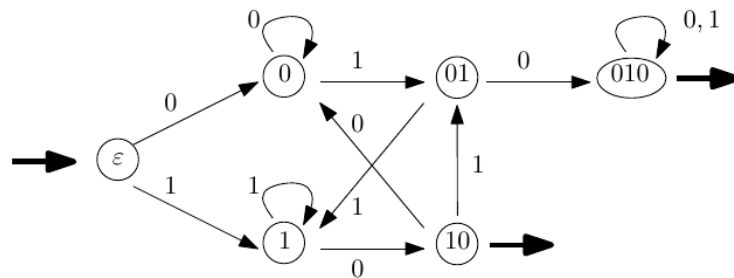
pro každé $q \in Q$, $w \in \Sigma^*$ a $x \in \Sigma$.

2. Pro každé $w \in Q$ a $x \in \Sigma$,

$$\delta(w, x) := \begin{cases} w, & \text{pokud } w \in S, \\ s, & \text{pokud } w \notin S \text{ a } wx \text{ končí na nějaké } s \in S \text{ (vezmi např. nejkratší),} \\ v, & \text{kde } v \text{ je nejdelší sufix slova } wx, \text{ který je v } Q, \text{ v ostatních případech.} \end{cases}$$

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Dokud automat nenašel žádné $s \in S$, je ve stavu v = nejdelší sufix dosud přečteného slova, který leží v Q (je prefixem nějakého slova z S). Tento invariant se přenáší: jakýkoli neprázdný sufix prodlouženého textu, který leží v Q , má tvar tx , kde t je opět prefix slova z S a sufix předchozího textu; podle invariantu je tedy $|t| \leq |w|$ a t je sufixem w , takže stačí hledat mezi sufixy wx . Ze stejného důvodu každý výskyt nějakého $s \in S$ končící právě čteným znakem je sufixem wx , takže ho zachytí druhý případ. Jakmile je s nalezeno, automat zůstane v přijímajícím stavu (první případ), protože podslovo už ze slova nezmizí.

3. Automat $A = (\{\varepsilon, 0, 1, 01, 10, 010\}, \{0, 1\}, \delta, \varepsilon, \{10, 010\})$ je na obrázku níže.



Doplňující vysvětlení (nad rámec oficiálního náčrtu): Stav 010 je absorbující (podslovo 010 nalezeno), kdežto ze stavu 10 automat po dalším znaku odejde (na 0 do stavu 0, na 1 do 01), takže přes stav 10 přijme jen slovo, které na 10 skutečně končí.

7 Porovnání modulárních hashů

(Reálná otázka: jaro 2026, č. 1 – Porovnání modulárních hashů (definice DFA, konstrukce DFA))
Zadání.

Mějme dané prvočíslo $p \geq 2$. Pro slovo $w \in \{0, 1\}^*$ nechť $h(w)$ je zbytek modulo p z nezáporného celého čísla, jehož binární reprezentace je w , přičemž povolujeme úvodní nuly. Přesněji, definujeme funkci $h: \{0, 1\}^* \rightarrow \{0, 1, \dots, p-1\}$ následovně:

$$h(w) = \left(\sum_{i=1}^{|w|} w_i 2^{|w|-i} \right) \bmod p,$$

kde $w = w_1 w_2 \dots w_{|w|}$ a $w_i \in \{0, 1\}$. (Pro prázdné slovo je tedy $h(\varepsilon) = 0$.)

Nyní uvažte následující jazyk nad abecedou $\Sigma = \{0, 1, \#\}$:

$$L_p = \{u\#v \mid u, v \in \{0, 1\}^* \text{ a platí } h(u) = h(v)\}.$$

1. Uveďte formální definici *deterministického konečného automatu (DFA)*.
2. Formálně popište konstrukci DFA A_p , který rozpoznává jazyk L_p , v závislosti na parametru p .
3. Nakreslete stavový diagram automatu A_2 (rozpoznávajícího jazyk L_2).

Náčrt řešení.

1. DFA je pětice $A = (Q, \Sigma, \delta, q_0, F)$, kde:

- Q je konečná neprázdná množina stavů,
- Σ je konečná neprázdná abeceda,
- $\delta: Q \times \Sigma \rightarrow Q$ je (parciální) přechodová funkce,
- $q_0 \in Q$ je počáteční stav,
- $F \subseteq Q$ je množina přijímajících stavů.

2. DFA $A_p = (Q, \Sigma, \delta, q_0, F)$ zkonstruujeme takto:

- $Q = [p] \cup [p] \times [p]$ kde $[p] = \{0, 1, \dots, p-1\}$ (stav i znamená, že jsme přečetli prefix u' slova u a $i = h(u')$, stav (i, j) znamená, že jsme přečetli prefix v' slova v , $i = h(u)$ a $j = h(v')$).
- $\Sigma = \{0, 1, \#\}$,
- $q_0 = 0$,
- $F = \{(i, i) \mid i \in \{0, 1, \dots, p-1\}\}$,
- přechodová funkce δ je definována takto (kde $i, j \in [p]$):
 - $\delta(i, 0) = 2i \bmod p$,
 - $\delta(i, 1) = (2i + 1) \bmod p$,
 - $\delta(i, \#) = (i, 0)$,
 - $\delta((i, j), 0) = (i, 2j \bmod p)$,
 - $\delta((i, j), 1) = (i, (2j + 1) \bmod p)$,

a zbývající přechody jsou nedefinované (anebo vedou do implicitního mrtvého stavu, tzv. žumpy).

3. Stavový diagram automatu A_2 :

