

KONEČNÉ AUTOMATY

Pojem „automat“ je historicky spojen s nějakou konstruktivní, algoritmickou procedurou rozhodující nějaký problém, či abstraktněji řečeno, rozhodující o tom, zda nějaký prvek patří do dané množiny (např. zda vstupní číslo je prvočíslo nebo zda je sudé). V tomto smyslu se mluví i o „Turingově automatu“, ačkoli se dnes ustálil pojem „Turingův stroj“ a termín „konečný automat“ je rezervován pro jednodušší rozhodovací procedury, kterými se budeme zabývat zde.

Pro různé rozhodovací postupy se také objevuje rozlišení mezi determinismem a nedeterminismem. V některých případech má nedeterminismus větší sílu, v některých nikoli, v některých se to neví. V případě konečných automatů odpovídá determinismus a nedeterminismus rozlišení mezi *rozeznatelnými* a *racionálními* podmnožinami monoidu. Klíčovým výsledkem je zde ovšem Kleeneova věta, která ukazuje, že pro volný monoid nad konečnou abecedou (tedy pro monoid slov) je síla deterministických a nedeterministických automatů ekvivalentní. To je důvod, proč se v informatice často obecný rozdíl mezi racionálními a rozeznatelnými množinami ignoruje a mluví se prostě o *regulárních jazycích*. Z algebraického hlediska je ovšem rozlišení důležité a je poučné i pro jazyky nad konečnou abecedou.

Racionální podmnožiny monoidu. Nechť $(M, \cdot, 1_M)$ je monoid, tedy množina M s binární asociativní operací a neutrálním prvkem (jednotkou). Systémem racionálních podmnožin M , značíme $\text{Rat}(M)$, je systém definován induktivně takto:

- $A \in \text{Rat}(M)$ pro všechny konečné množiny A ,
- pokud $A, B \in \text{Rat}(M)$, pak také $A \cup B \in \text{Rat}(M)$,
- pokud $A, B \in \text{Rat}(M)$, pak také $A \cdot B \in \text{Rat}(M)$,
- pokud $A \in \text{Rat}(M)$, pak také $\langle A \rangle \in \text{Rat}(M)$.

Součin množin přitom definujeme $A \cdot B = \{a \cdot b \mid a \in A, b \in B\}$ a uzávěrem $\langle A \rangle$ množiny A míníme nejmenší podmonoid M obsahující A .

Z induktivní definice racionálních množin plyne, že každou racionální množinu lze popsat jako posloupnost operací sjednocení, násobení a uzávěru aplikovaných na konečné množiny. Tuto posloupnost lze zapsat tzv. *regulárním výrazem*.

Regulárním výrazem nad abecedou Σ je každé správně utvořené slovo nad Σ a operačními symboly $\{\mathbf{1}, \mathbf{0}, +, \cdot, *\}$, kde $\mathbf{1}$ a $\mathbf{0}$ jsou konstanty, $+$ a \cdot jsou binární symboly a $*$ je unární symbol.

Regulární výrazy jsou tedy dány následující induktivní definicí.

- $\mathbf{1}$, $\mathbf{0}$ a $a \in \Sigma$ jsou regulární výrazy (nazývané *atomické*);
- jsou-li r a s regulární výrazy, je i
 - $(r \cdot s)$
 - $(r + s)$
 - (r^*)

regulární výraz.

Symbol $*$ se nazývá *Kleeneova hvězda*. Pokud přijmeme obvyklou přednost symbolu násobení před symbolem sčítání a přednost Kleeneovy hvězdy před symboly sčítání a násobení, můžeme vynechávat některé závorky. Symbol násobení také obvykle nahrazujeme prostým zřetěžením. Výrazy $r \cdot r^*$ a $r^* \cdot r$ se obvykle zkracují na r^+ (tzv. *Kleeneovo plus*).

Rozšířený regulární výraz je definován přidáním unárního symbolu $\bar{}$ a pravidla

- je-li r regulární výraz, je i \bar{r} regulární výraz.

Jak již bylo řečeno, každou racionální podmnožinu monoidu M lze definovat regulárním výrazem nad Σ , kde $M = \langle \Sigma \rangle$. A naopak, každý regulární výraz nad Σ definuje nějakou racionální podmnožinu M .

Souvislost je dána následujícími přirozenými pravidly.

- Regulárnímu výrazu $\mathbf{0}$ odpovídá prázdná množina;
- regulárnímu výrazu $\mathbf{1}$ odpovídá množina $\{1_M\}$;
- regulárnímu výrazu $a \in \Sigma$ odpovídá množina $\{a\}$;
- regulárnímu výrazu $(r \cdot s)$, kde r je regulární výraz odpovídající množině A a s je regulární výraz odpovídající množině B , odpovídá množina $A \cdot B$;
- regulárnímu výrazu $(r + s)$, kde r je regulární výraz odpovídající množině A a s je regulární výraz odpovídající množině B , odpovídá množina $A \cup B$;
- regulárnímu výrazu r^* , kde r je regulární výraz odpovídající množině A , odpovídá množina $\langle A \rangle$.

Konečná množina $\{m_1, m_2, \dots, m_n\}$, kde

$$m_i = a_{i,1}a_{i,2} \cdots a_{i,k_i},$$

$a_i \in \Sigma$, je přitom definována regulárním výrazem

$$a_{1,1}a_{1,2} \cdots a_{1,k_1} + a_{2,1}a_{2,2} \cdots a_{2,k_2} + \cdots + a_{n,1}a_{n,2} \cdots a_{n,k_n}.$$

Pro rozšířený regulární výraz navíc platí, že

- regulárnímu výrazu \bar{r} , kde r je regulární výraz odpovídající množině A , odpovídá množina $\Sigma^* \setminus A$ (tedy doplněk A v Σ^*).

Poznamenejme, že $\text{Rat}(M)$ obecně není uzavřena na komplement. Z Kleeneovy věty (níže) ovšem vyplývá, že na komplement jsou uzavřeny regulární jazyky (tedy racionální množiny slov nad konečnou abecedou), protože v deterministických automatech lze získat komplement pomocí komplementu množiny přijímajících stavů. Pro jazyky nad konečnou abecedou je tedy možné uvažovat rozšířené regulární výrazy. Uvedená korespondence je rovněž důvodem, proč se racionální množiny někdy označují jako „regulární množiny“ i v obecném případě.

Nedeterministické automaty nad monoidem. *Nedeterministický M -automat* \mathcal{A} je čtveřice (Q, δ, I, F) , kde Q je množina stavů, $\delta \subseteq Q \times M \times Q$ je přechodová relace, $I \subseteq Q$ je množina počátečních stavů a $F \subseteq Q$ je množina přijímajících stavů. Automat se nazývá *konečný*, pokud je δ konečná množina.

Prvek $(q, m, q') \in \delta$ si typicky představujeme jako šipku vedoucí ze stavu q do stavu q' a označenou popiskou m a celý automat jako orientovaný graf s ohodnocenými hranami. Automat *přijímá* prvek $m \in M$, pokud v takto definovaném grafu existuje cesta z počátečního stavu do nějakého přijímajícího stavu jejíž ohodnocení je rovno m . Přesněji, taková *přijímající cesta* prvku m je posloupnost $(q_0, m_1, q_1, m_2, q_2, \dots, m_k, q_k)$, kde $m = m_1 \cdot m_2 \cdots m_k$, $(q_{i-1}, m_i, q_i) \in \delta$ pro všechna $i = 1, 2, \dots, k$, $q_0 \in I$ a $q_k \in F$. Množinu prvků přijímaných nedeterministickým automatem \mathcal{A} značíme $L(\mathcal{A})$. (Říkáme také, že \mathcal{A} *přijímá* $L(\mathcal{A})$). Dvoznačnost mezi termíny „přijímaný prvek“ a „přijímaná množina“ by neměla působit zmatek.) Automaty přijímající stejnou množinu nazýváme *ekvivalentní*.

Všimněme-si, že definice dovoluje tzv. ε -přechody, tedy prvky $(q, 1_M, q')$. To níže umožní některé elegantní konstrukce. Na druhou stranu bychom někdy chtěli uvažovat automaty bez ε -přechodů. Ještě přísněji, pro nějakou podmnožinu $\Sigma \subseteq M$ můžeme uvažovat *nedeterministický M -automat nad Σ* , kde $\delta \subseteq Q \times \Sigma \times Q$.

Podmínky na přechodovou funkci můžeme naopak uvolnit a dovolit, aby hrany byly ohodnoceny racionálními množinami, tj. $\delta \subseteq Q \times \text{Rat}(M) \times Q$. Pak mluvíme o *nedeterministickém M -automatu nad $\text{Rat}(M)$* . Přijímající cesta prvku m je pak definována jako posloupnost $(q_0, A_1, q_1, A_2, q_2, \dots, A_k, q_k)$, kde $(q_{i-1}, A_i, q_i) \in \delta$, $m = m_1 \cdot m_2 \cdot \dots \cdot m_k$ pro nějaká $m_i \in A_i$ a $q_k \in F$.

Množiny přijímané konečnými nedeterministickými automaty přesně odpovídají racionálním množinám monoidu M v následujícím smyslu:

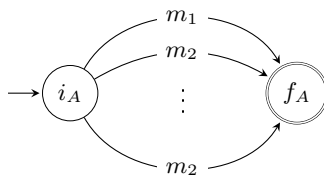
Věta. *Nechť je M monoid, $A \subseteq M$ a $M = \langle \Sigma \rangle$. Následující podmínky jsou ekvivalentní:*

- (1) $A \in \text{Rat}(M)$;
- (2) *existuje konečný nedeterministický M -automat přijímající A ;*
- (3) *existuje konečný nedeterministický M -automat nad Σ s jediným vstupním stavem přijímající A ;*
- (4) *existuje konečný nedeterministický M -automat nad $\text{Rat}(M)$ přijímající A .*

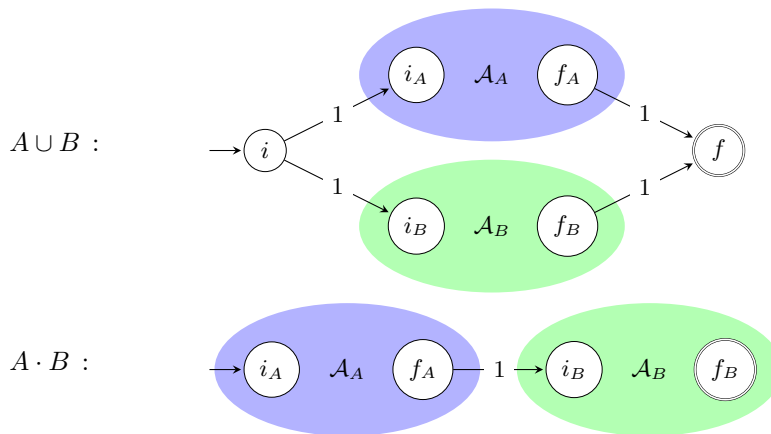
Důkaz.

(1) \Rightarrow (2) Indukcí podle definice racionální množiny zkonstruujeme automat, který jazyk přijímá (jde o tzv. *Thompsonovu konstrukci*).

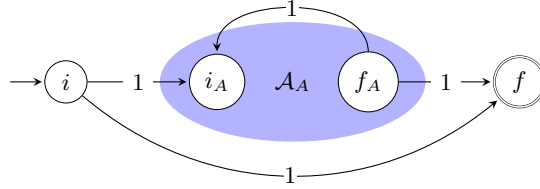
- Je-li $A = \{m_1, m_2, \dots, m_k\}$ konečná množina, má konečný automat tvar



- Máme-li automaty pro množiny A a B , získáme automaty pro $A \cdot B$ a $A \cup B$ takto:



- Automat pro $\langle A \rangle$ získáme z automatu pro A takto:



(Vnější vstupní a přijímající stavy s ε -přechodem jsou zde přidány proto, aby automat přijímal jedničku. Pokud jazyk A už jedničku přijímá, stačilo by přidat pouze hranu vedoucí z f_A do i_A . Všimněme si, že obecně nelze jedničku do jazyka přidat ε -přechodem z i_A do f_A , protože tak bychom mohli nepatřičně přidat prvky odpovídající cestám z f_A do f_A neprocházejícím počátečním stavem.)

(2) \Rightarrow (3) Vzhledem k tomu, že Σ generuje M , lze každý přechod (q, m, q') , kde $m \neq 1$, nahradit přechody $(q, s_0, q_1), (q_1, s_1, q_2), \dots, (q_k, s_k, q')$, kde q_i jsou nové stavy, $s_i \in \Sigma$ a $m = s_1 s_2 \dots s_k$.

Zbývá eliminovat ε -přechody. Těm se také někdy říká „spontánní“ přechody, protože za přítomnosti přechodu $(q, 1, q')$ lze stav q kdykoli podle libosti zaměnit za q' . Takové spontánní přechody lze odstranit tím, že již při vstupu do stavu q „donutíme“ automat, aby se „rozhodl“, zda chce spontánní přechod využít, nebo ne. Formálně budeme postupovat takto. Nejprve přidáme nový počáteční stav a z něj jdoucí ε -přechody do dosavadních počátečních stavů, přičemž tento nový počáteční stav bude nadále jediným počátečním stavem. Dále vytvoříme tranzitivní uzávěr ε -přechodů, to znamená, že přidáme nové ε -přechody tak, aby přítomnost přechodů $(q, 1, q')$ a $(q', 1, q'')$ implikovala přítomnost přechodu $(q, 1, q'')$. Nyní pro všechny dvojice přechodů $(s, m, q), (q, \varepsilon, q')$, kde $m \neq 1$, přidáme přechod (s, m, q') a poté všechny ε -přechody odstraníme.

Získáme tak automat v požadovaném tvaru. Ověření, že je ekvivalentní automatu původnímu, je přímočaré. Stačí z přijímající cesty odstranit (nebo do ní pro opačnou inkluzi přidat) příslušné ε -přechody.

(3) \Rightarrow (4) Stačí nahradit každý přechod (q, m, q') přechodem $(q, \{m\}, q')$.

(4) \Rightarrow (1) Nechť jsou stavy automatu q_1, q_2, \dots, q_n . Automat nejprve upravíme tak, aby měl jediný vstupní stav, do kterého nevedou žádné (neprázdné) přechody, a jediný přijímající stav, ve kterém žádné (neprázdné) přechody nezačínají. To lze docílit přidáním nového počátečního (resp. nového koncového) stavu q_0 (resp. q_{n+1}) a přidáním ε -přechodů do původních počátečních (resp. z původních přijímajících) stavů. Dále můžeme předpokládat, že pro libovolnou dvojici stavů (q_i, q_j) , včetně smyček (q_i, q_i) , existuje právě jeden přechod $(q_i, A_{i,j}, q_j)$, protože násobné přechody můžeme sloučit pomocí sjednocení jazyků a pro neexistující přechody definujeme $A_{i,j} = \emptyset$.

Nyní postupně eliminujeme všechny stavy kromě q_0 a q_{n+1} takto: Odstraníme zvolený vrchol q_i a pro všechna $j, k \neq i$ definujeme nové jazyky

$$A'_{j,k} = A_{j,k} \cup A_{j,i} \langle A_{i,i} \rangle A_{i,k}.$$

Je opět přímočaré ověřit, že nově vzniklý automat je ekvivalentní původnímu. Libovolný úsek přijímající cesty tvaru $q_j, m_1, q_i, m_2, q_i, m_3, \dots, q_i, m_\ell, q_k$ v původním automatu totiž odpovídá cestě q_j, m, q_k , kde $m = m_1 m_2 \dots m_\ell \in A'_{j,k}$, a naopak.

Nové jazyky jsou přitom všechny racionální. Po odstranění všech vrcholů tedy zůstane jediný (neprázdný) přechod (q_0, A, q_{n+1}) a $A \in \text{Rat}(M)$. \square

Rozpoznatelné podmnožiny monoidu. Jiným způsobem, jak efektivně popsat podmnožinu monoidu je pojem rozpoznatelnosti. Řekneme, že *monoid* N *rozeznává množinu* $L \subseteq M$, pokud existuje homomorfismus $\varphi : M \rightarrow N$ takový, že $L = \varphi^{-1}\varphi(L)$.

Připomeňme, že *jádrem* homomorfismu φ rozumíme ekvivalenci \sim_φ definovanou:

$$u \sim_\varphi v, \text{ právě když } \varphi(u) = \varphi(v).$$

Přitom platí, že jádro homomorfismu je vždy kongruence; a naopak, každá kongruence \sim na M definuje homomorfismus (*přirozenou projekci*)

$$\varphi : M \rightarrow M/\sim$$

vztahem $\varphi(u) = [u]$, jehož jádrem je právě \sim (kde $[u]$ značí třídu, ve které leží u).

Je tedy vidět, že monoidy rozeznávající množinu L jsou dány kongruencemi monoidu M , pro které platí, že L se rozkládá na třídy této kongruence. Nechť je \sim taková kongruence. Pak $u \sim v$ implikuje $rus \sim rvs$ pro každé r a s , neboli

$$(1) \quad \forall r, s \in M : rus \in L \Leftrightarrow rvs \in L.$$

Přímočaře lze ověřit, že pokud definujeme ekvivalenci $u \sim_L v$ vztahem (1), dostaneme kongruenci. Ta se nazývá *syntaktická kongruence* množiny L , monoid $\text{Synt}(L) := M/\sim_L$ se nazývá *syntaktický monoid* množiny L a příslušná přirozená projekce se nazývá *syntaktický homomorfismus*. Z definice je vidět, že je to největší kongruence, pro kterou se množina L rozkládá na třídy. („Největší“ znamená, že má nejmenší počet tříd a všechny ostatní jsou jejím zjemněním.) Monoid M/\sim_L se nazývá *syntaktický monoid* množiny L , a je to tedy nejmenší monoid rozeznávající množinu L , a to pomocí přirozené projekce $u \mapsto [u]$.

Deterministické automaty nad monoidem. *Deterministický* M -*automat* je čtveřice $\mathcal{A} = (Q, \delta, q_0, F)$. Podobně jako u nedeterministického automatu je Q množina stavů, $q_0 \in Q$ je počáteční stav a $F \subseteq Q$ je množina přijímajících stavů. Namísto přechodové relace ovšem máme *přechodovou funkci* $\delta : Q \times M \rightarrow Q$, která pro všechna $u, v \in M$ a všechna $q \in Q$ splňuje $\delta(\delta(q, u), v) = \delta(q, u \cdot v)$, a $\delta(q, 1_M) = q$. Namísto $\delta(q, u)$ píšeme $q \cdot u$, a podmínky potom mají tvar

$$(q \cdot u) \cdot v = q \cdot (u \cdot v), \\ q \cdot 1 = q.$$

(Všimněte si, že používáme znak násobení ve dvou různých významech.) Deterministický automat se nazývá *konečný*, pokud je množina stavů Q konečná. Automat \mathcal{A} *přijímá* množinu

$$L(\mathcal{A}) = \{u \in M \mid q_0 \cdot u \in F\}.$$

Homomorfismus $\varphi : M \rightarrow N$, s jehož pomocí rozeznává N množinu $L \subseteq M$, přirozeně definuje deterministický automat $\mathcal{A}_\varphi = (\varphi(M), \delta, 1_N, F)$ nad M , kde δ je definováno předpisem $\varphi(u) \cdot v = \varphi(u \cdot v)$ a $F = \varphi(L)$. Snadno ověříme, že $L(\mathcal{A}_\varphi) = L$.

Uvažujme naopak deterministický automat $\mathcal{A} = (Q, \delta, q_0, F)$ nad M . Každé slovo $u \in M$ určuje zobrazení $\delta_u : Q \rightarrow Q$ dané vztahem $\delta_u(q) = q \cdot u$, tzv. *transformaci množiny* stavů. Na transformacích definujeme binární operaci jako skládání zleva:

$$\delta_u \cdot \delta_v := \delta_{uv},$$

čímž vznikne *transformační monoid automatu* \mathcal{A} :

$$T(\mathcal{A}) := (\{\delta_u \mid u \in M\}, \cdot, \text{id}).$$

Snadno ověříme, že monoid $T(\mathcal{A})$ rozeznává $L(\mathcal{A})$ pomocí homomorfismu $\varphi : u \mapsto \delta_u$.

Je-li množina $L \subseteq M$ rozeznávána nějakým konečným monoidem, nazývá se *rozpoznatelná*. Třidu rozpoznatelných podmnožin M značím $\text{Rec}(M)$. Z dosavadních úvah plyne:

Věta. *Nechť $L \subseteq M$. Pak jsou následující podmínky ekvivalentní:*

- L je rozpoznatelný;
- L je přijímán konečným deterministickým automatem nad M ;
- syntaktická kongruence L je konečná.

Minimální automat. Již jsme řekli, že nejmenší monoid N rozeznávající danou množinu $L \subseteq M$ je dán syntaktickou kongruencí jako $N = M/\sim_L$. Prvky N nám současně posloužily jako stavy deterministického automatu, který přijímá L . Můžeme se ptát, zda je tento automat nejmenší možný (tj., zda má nejmenší možný počet stavů).

Je-li naopak dán automat (např. co nejmenší) přijímající L , sestrojili jsme k němu monoid rozeznávající L jako monoid transformací. Opět se můžeme ptát, zda se jedná o nejmenší takový monoid.

Je vidět, že odpověď nebude v obou případech kladná. Vyjdeme-li totiž od nějakého rozeznávajícího monoidu N , zkonstruujeme k němu automat a k tomuto automatu jeho monoid transformací, nedostaneme původní N , ale nějaký obecně větší monoid obsahující zobrazení $N \rightarrow N$. Pokud by měl být monoid transformací nejmenším přijímajícím monoidem, muselo by tedy být stavů nejmenšího automatu méně než prvků syntaktického monoidu. Je to možné?

Chceme-li snížit počet stavů na minimum, uvažujme následovně. Stav automatu v daném okamžiku plně určuje, jak se bude automat dále chovat. Určuje zejména, jaké následující vstupy budou přijaty. To je také to jediné, co nás na daném stavu zajímá. Z toho plynou dva poznatky:

- pokud $q_0 \cdot u = q_0 \cdot v$, pak pro jakékoli $w \in M$ platí, že $uw \in L$ právě když $vw \in L$;
- stav $q_u = q_0 \cdot u$ je plně charakterizován jazykem přijímaným z q_u , což je jazyk $L(u) := \{w \in M \mid uw \in L\}$. Jazyk $L(u)$ se často značí $u^{-1}L$ a takové jazyky nazýváme *levé kvocienty* L .

Levé kvocienty, označme jejich množinu Q_L , tedy tvoří stavy minimálního deterministického automatu $\mathcal{A}_L = (Q_L, \delta, 1_M, F)$ přijímajícího L , kde přechodová funkce je definována podobně jako výše předpisem $L(u) \cdot v = L(u \cdot v)$.

Levé kvocienty definují tzv. *Myhillovu-Nerodovu* ekvivalenci \sim_R , kde $u \sim_R v$, právě když $L(u) = L(v)$. Tato ekvivalence typicky není kongruence (ověřte!), nicméně pro každé $s \in M$ platí $u \sim_R v \Rightarrow u \cdot s \sim_R v \cdot s$.

S použitím „lingvistických“ termínů lze říct, že Myhillova-Nerodova ekvivalence ztotožňuje prvky, které mají v L stejný *pravý kontext*. Podobně lze syntaktickou kongruenci definovat jako ekvivalenci ztotožňující prvky se stejným (oboustranným) *kontextem* definovaným pro u jako množina $C_L(u) = \{(r, s) \mid rus \in L\}$. Syntaktická kongruence je tedy zjemněním Myhillovy-Nerodovy ekvivalence.

Otázky z úvodu tohoto oddílu jsou tedy zodpovězeny následující větou.

Věta. *Monoid transformací minimálního automatu je isomorfní syntaktickému monoidu.*

Důkaz. Nechť $\delta_u \in T(\mathcal{A}_L)$ značí transformaci $L(r) \mapsto L(ru)$. Ukážeme, že zobrazení $\delta_u \mapsto [u]$ je isomorfismus $T(\mathcal{A}_L)$ a $\text{Synt}(M)$. Zobrazení je surjektivní homomorfismus, protože

$$\delta_u \cdot \delta_v = \delta_{uv} \mapsto [uv] = [u] \cdot [v].$$

(Poznamenejme, že $[u] \cdot [v]$ zde značí součin tříd kongruence. Všechny předchozí rovnosti tedy triviálně plynou z definic.) Zbývá ukázat, že zobrazení je prosté.

Nerovnost $\delta_u \neq \delta_v$ znamená, že

$$\delta_u(L(r)) = L(ru) \neq L(rv) = \delta_v(L(r))$$

pro nějaké $r \in M$. Existuje tedy $s \in M$, pro které $rus \in L$ není ekvivalentní $rvs \in L$. Tedy $[u] \neq [v]$, což jsme chtěli ukázat. \square

Vztah racionálních a rozpoznatelných podmnožin.

Věta (McKnight). *Inkluze $\text{Rec}(M) \subseteq \text{Rat}(M)$ platí, právě když je monoid M konečně generovaný.*

Důkaz. Je snadné ověřit, že každá racionální množina leží v konečně generované nadmnožině. Současně platí, že M je vždy rozpoznatelná triviálním monoidem $\{1\}$. Pokud je tedy M nekonečně generovaný, je M rozpoznatelná množina, která není racionální.

Je-li naopak M generovaný konečnou množinou Σ a $\text{Synt}(L)$ je konečný, pak lze přechodovou funkci deterministického automatu \mathcal{A}_L nahradit konečnou přechodovou relací, která obsahuje všechny trojice (q, u, q') , kde $s \in \Sigma$, $q \in Q_L$ a kde $q' = q \cdot u$ je dáno přechodovou funkcí \mathcal{A}_L . \square

Druhá implikace předchozího důkazu ukazuje rozdíl mezi konečností deterministického a nedeterministického automatu. Chápeme-li automat jako orientovaný graf s ohodnocenými hranami, znamená konečnost deterministického automatu pouze konečný počet vrcholů, nikoli konečný počet hran. Definice přechodové funkce naopak pro nekonečný monoid M počítá s nekonečným počtem hran vycházejícím z každého vrcholu. Přejít ke konečnému nedeterministickému automatu spočívá v redukci těchto hran pouze na ty, které jsou ohodnoceny generujícími prvky.

Věta (Kleene). *Pro konečnou abecedu Σ platí $\text{Rec}(\Sigma^*) = \text{Rat}(\Sigma^*)$.*

Důkaz. Díky McKnightově větě zbývá ukázat, že každá racionální podmnožina Σ^* je rozpoznatelná. Díky charakterizacím $\text{Rat}(M)$ a $\text{Rec}(M)$ pomocí automatů stačí ukázat, že nedeterministický Σ^* -automat $\mathcal{A} = (Q, \delta, I, F)$ nad Σ lze transformovat na deterministický Σ^* -automat.

Množinou stavů hledaného deterministického automatu bude potenční množina $\mathcal{P}(Q)$, počátečním stavem je I a množinou přijímajících stavů je $F' = \{S \mid S \cap F \neq \emptyset\}$. Klíčovým krokem důkazu je následující definice přechodové funkce. Nechť $S \in \mathcal{P}(Q)$, a $a \in \Sigma$. Pak

$$S \cdot a = \{q \in Q \mid (q', a, q) \in \delta \text{ pro nějaké } q' \in S\}.$$

Na základě tohoto předpisu dostáváme z asociativního požadavku na deterministický automat indukční definici $S \cdot v = (S \cdot u) \cdot a$, kde $v = ua$, $v \in \Sigma^*$, $a \in \Sigma$, přičemž $S \cdot \varepsilon = S$ pro všechna $S \in \mathcal{P}(Q)$. Tato definice je korektní díky tomu, že každé slovo má jednoznačný rozklad na součin písmen.

Je nyní snadné ověřit (indukcí), že $I \cdot v \in F'$, právě když existuje přijímající cesta v \mathcal{A} . \square

- Ukažte, že $\{0\}$ není rozpoznatelnou množinou monoidu $(\mathbb{Z}, +)$ (a inkluze z věty je tedy v tomto případě ostrá).
- Co selže při pokusu rozšířit nedeterministický automat přijímající $\{0\}$ v $(\mathbb{Z}, +)$ na deterministický?
- Je $\{0\}$ rozpoznatelnou množinou monoidu $(\mathbb{N}, +)$?
- Je \mathbb{N} racionální množinou monoidu (\mathbb{N}, \cdot) ?

Poznámka: Monoidy, pro které platí Kleeneova věta, se nazývají *Kleeneho monoidy*. Viděli jsme, že volný, konečně generovaný monoid je Kleeneho monoid. Je možné tuto podmínku zeslabit? Předpokládejme, že Σ je konečná generující množina monoidu M . Definice přechodové funkce na generátorech je vždy korektní. Problematická je jednoznačnost definice $S \cdot u$, pokud má u více faktorizací na generátory. To se můžeme pokusit obejít následujícím způsobem. Definujeme

$$S \cdot u = \bigcup_{u=v \cdot a} (S \cdot v) \cdot a.$$

Neboli: $S \cdot u$ je množina stavů, do kterých se lze dostat z nějakého stavu S nějakou faktorizací $u = v \cdot a$, kde $a \in \Sigma$. To je induktivní definice, která je korektní, pokud na M existuje uspořádání $<$ kompatibilní s monoidovou operací, tj. splňující, že $v < v \cdot a$ pro všechna $v \in M$ a všechna $a \in \Sigma$.