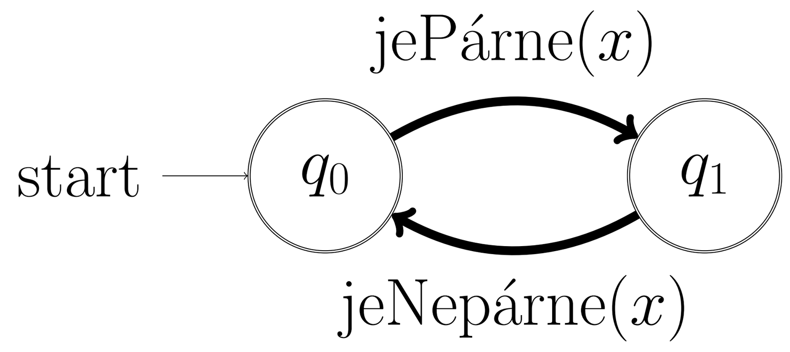
Simulačný algoritmus pre symbolické automaty

Juraj Síč

Symbolické automaty

Symbolické automaty sú podobné klasickým automatom, ale narozdiel od nich sú parametrizované logickou teóriou nad doménou symbolov. Na prechodoch sú formule z tejto teórie s jednou voľnou premennou, ktoré reprezentujú potencionálne nekonečné množiny symbolov.



Tento automat príjma jazyk $\mathcal{L} = \{l : l$ je postupnosť čísli̇c, kde sa strieda párna s nepárnou$\}$.

Simulácia

Nech $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Simulácia na M je kváziusporiadanie $\preceq \in Q \times Q$ také, že:

(i) $\preceq \cap (F \times (Q - F)) = \emptyset$,

(ii) pre $p, q \in Q$, $a \in \Sigma$, ak $p \preceq r$, potom pre každý prechod $p \xrightarrow{a} p'$ existuje $r \xrightarrow{a} r'$ taký, že $p' \preceq r'$.

Ak nejaký stav q simuluje stav p, znamená to, že q príjma rovnaký jazyk ako p. Toto sa dá využiť pri dokazovaní jazykovej inklúzie alebo pri redukovaní počtu stavov nedeterministického automatu.

Algoritmus na klasickom konečnom automate

Algoritmus, z ktorého sme vychádzali, počíta doplnok k simulácii na klasickom konečnom automate. Podľa definície $\preceq$ je jej doplnok $\not\preceq$ najmenšia relácia nad Q taká, že:

(i)' $(F \times (Q - F)) \subseteq \not\preceq$,

(ii)' pre $i, j \in Q$ je $i \not\preceq j$ práve vtedy, keď existujú $a \in \Sigma$ a prechod $i \xrightarrow{a} i'$ také, že pre každé $j' \in Q$, pre ktoré existuje prechod $j \xrightarrow{a} j'$, platí $i' \not\preceq j'$.

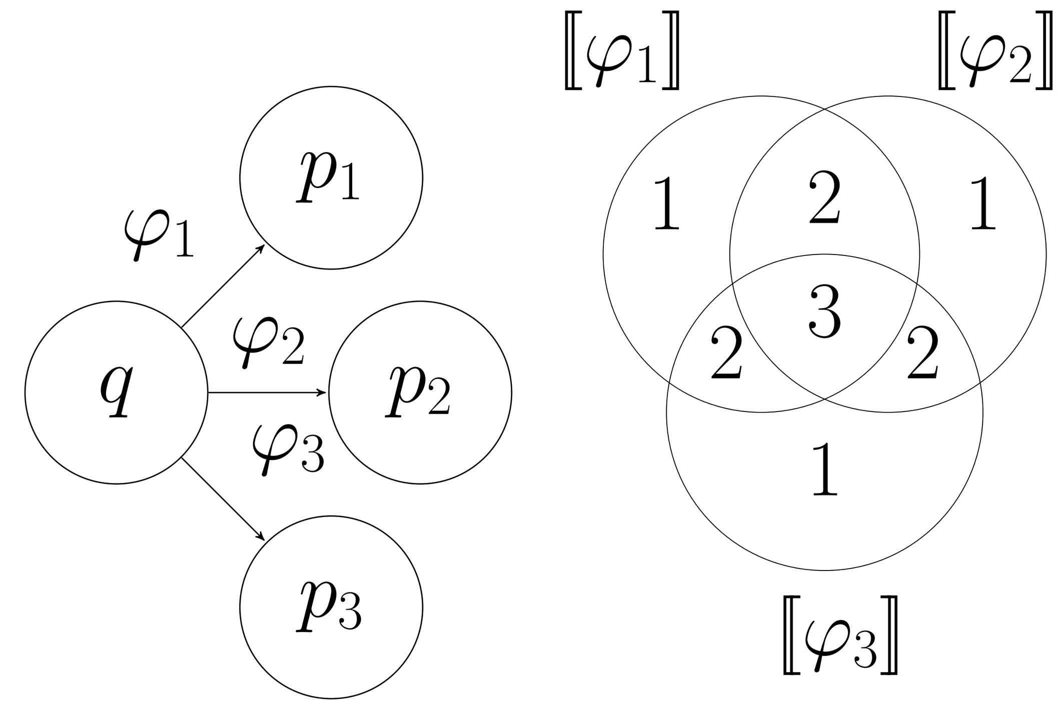
Začnú s reláciou v bode (i)' a potom túto reláciu postupne zväčšujú pomocou (ii)'. Teda ak spracovávajú dvojicu (i, j), potrebujú otestovať, či existuje $t \in Q$ a $a \in \Sigma$, také, že $t \xrightarrow{a} j$ a zároveň je j posledný stav, ktorý by mohol simulovať i. Aby sa toto urýchlilo, vypočítavajú si matice počítadiel $N(a)$, pre $a \in \Sigma$, kde $N(a)$ je $|Q| \times |Q|$ matica taká, že

$$N(a)_{it} = |\{t \xrightarrow{a} j, i \not\preceq l\}|$$

na konci výpočtu algoritmu pre všetky $i, t \in Q$.

Mintermy

V pôvodnom algoritme si vypočítavali matice počítadiel pre každý symbol z abecedy. Toto nie je pre symbolické automaty prijateľné, takže pre ne si vytvoríme počítadlá nad predikátmi. Pre každý stav $q \in Q$ si zavedieme *rozklad na lokálne mintermy*, čo je najhrubší rozklad domény symbolov, ktorý je kompatibilný s predikátmi nad prechodmi vychádzajúcimi z q. Predikáty nad počítadlami budú tieto mintermy.



Algoritmus pre symbolický automat

Input: SFA M
Output: $\not\preceq$

```
for q ∈ F do
    Vypočítaj všetky lokálne mintermy ψ stavu q a
    inicializuj všetky N(ψ)
ω ← ∅, C ← NEWQUEUE()
// inicializácia relácie doplnku
for i ∈ F do
    for j ∈ Q − F do
        ω ← ω ∪ {(i, j)}
        ENQUEUE(C, (i, j))
// zväčšovanie tejto relácie
while C ≠ ∅ do
    (i, j) ← DEQUEUE(C)
    S ← pre(i), T ← pre(j)
    for t ∈ T where t -φtj-> j do
        for ψ where ψ is minterm of t and
        IsSat(ψ ∧ φtj) do
            N(ψ)_it ← N(ψ)_it − 1
            if N(ψ)_it = 0 then
                for s ∈ S where s -φsi-> i and IsSat(ψ ∧ φsi)
                do
                    if (s, t) ∉ ω then
                        ω ← ω ∪ {(s, t)}
                        ENQUEUE(C, (s, t))
return ω
```

Algoritmus 1: Simulácia SKA. Očakáva, že M je čistý, normalizovaný a úplný.