Modifikace deterministických konečných automatů

Vlastimil Košař

15. 12. 2010

Obsah

Obsah

Obsah

Obsah

Motivace

Proč modifikovat deterministický konečný automat (DKA)

Složitost determinizace - $O(2^N)$

Problémy při vyhledávání vzorů

- Reálná pravidla tvoří až tisíce regulárních výrazů (RV)
  - Množství konstrukcí typu $.*$ a $\{M, N\}$ - stavová exploze
- Problematické z hlediska velikosti potřebné paměti
- Nevyužije se cache procesoru

Rozdělení přístupů

Varianty

- Modifikace DKA
  - Modifikují klasický model DKA
- Hybridní
  - Kombinují deterministické a nedeterministické konečné automaty
- Hierarchické
  - Vytvářejí hierarchii automatů

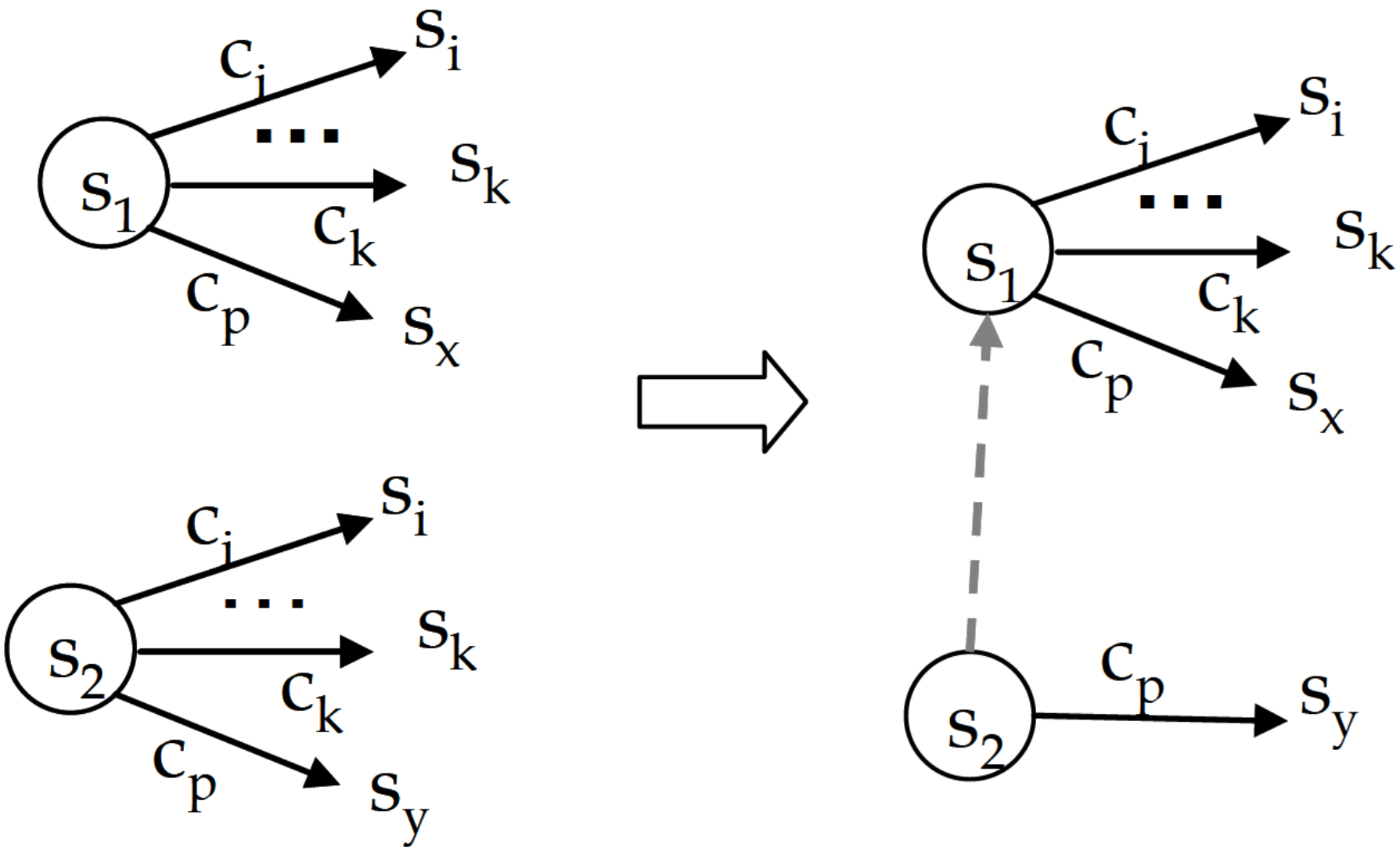
Deterministický konečný automat se zpožděným vstupem

Delayed Input Deterministic Finite Automata (D2FA)

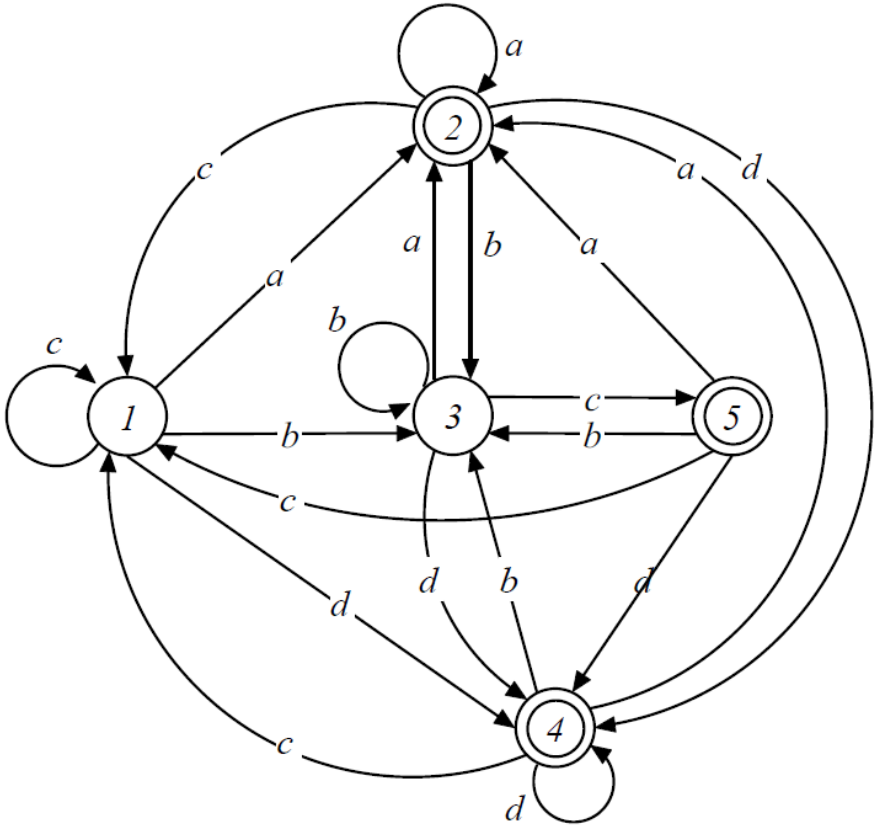
Základní myšlenka

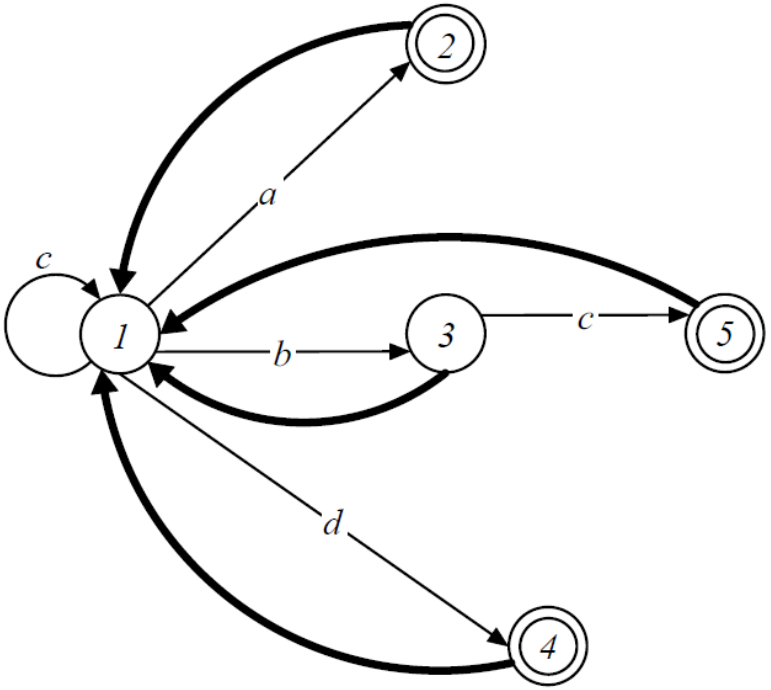
- Mnoho stavů DKA má společnou velkou část přechodů
- Zavedeme tudíž výchozí přechody
  - Provede se, není-li možné provést klasický přechod
- Definujeme algoritmus převodu DKA na ekvivalentní D2FA

Ilustrace - výchozí přechod



Ilustrace - DKA $\rightarrow$ D2FA





Vysvětlivka

DKA a D2FA pro regulární výraz $a^+ + b^+c + c^*d^+$

Definice D2FA

Definice

Deterministický konečný automat se zpožděným vstupem je pětice $(Q, \Sigma, q_0, \delta, F)$

- Q - konečná množina stavů
- Σ - konečná vstupní abeceda
- $q_0 \in Q$ - počáteční symbol
- $\delta : Qx(\Sigma \cup \{\lambda\}) \rightarrow Q$ - přechodová funkce
- $F \subseteq Q$ - množina koncových stavů

λ značí výchozí přechod

Konfigurace a přechod D2FA

Konfigurace

Konfigurace C $C = (q, w), (q, w) \in Qx\Sigma^*$, kde q je aktuální stav a w je dosud nezpracovaná část vstupního řetězce.

Přechod

$$\vdash \subseteq (Qx\Sigma^*)x(Qx\Sigma^*)$$

která je definována následovně:

$$\begin{aligned}(q, w) \vdash (\acute{q}, \acute{w}) \quad \Leftrightarrow \quad & (w = a\acute{w} \wedge \acute{q} \in \delta(q, a)) \vee \\ & (w = \acute{w} \wedge w = a\hat{w} \wedge \acute{q} \notin \delta(q, a) \wedge \acute{q} \in \delta(q, \lambda))\end{aligned}$$

kde $q, \acute{q} \in Q, a \in \Sigma, w, \acute{w}, \hat{w} \in \Sigma^*$

Ekvivalence DKA a D2FA

Lemma

Předpokládejme D2FA se stavy u a v ($u \neq v$), kde u má přechod označený symbolem a a žádný odchozí výchozí přechod. Pokud platí $\delta(u, a) = \delta(v, a)$, pak je D2FA získaný zavedením výchozího přechodu z u do v a odstraněním přechodu z u do $\delta(u, a)$ ekvivalentní s původním DKA.

Konstrukce D2FA

Konstrukce

- Neexistuje způsob přímé konstrukce D2FA
- Vždy nutno vytvořit DKA
- Každý stav má nejvýše jeden odchozí výchozí přechod
- Výchozí přechody nesmí tvořit cyklus
- Postup konstrukce
  1. Vytvoř graf prostorové redukce
  2. Najdi maximální kostru grafu
  3. Podle kostry vytvoř výchozí přechody

Graf prostorové redukce

Graf prostorové redukce

- Vytvořen z DKA
- Úplný neorientovaný ohodnocený graf

Definice

Graf prostorové redukce je trojice (V, H, c):

- U - konečná množina vrcholů, V = Q
- $H = \{\{u, v\} | u, v \in U\}$ - konečná množina hran
- $c : H \rightarrow N$ - ohodnocení hran,
  $c(\{u, v\}) = |\{a | \forall a \in Q : \delta(u, a) = \delta(v, a)\}| - 1$

Hledání maximální kostry grafu

Vytvoření výchozích přechodů

- Najdeme maximální kostru grafu
- Vybereme vhodný stav a všechny defaultní přechody k němu nasměrujeme
- Problém - dlouhé cesty v minimální kostře grafu
- Řešení:
  1. Omezení průměru kostry grafu - NP-těžký problém
  2. Les maximálních stromů s omezeným průměrem - NP-těžký problém pro $d > 1$
  3. Modifikace Kruskalova algoritmu - vytvaří inkrementáně les stromů s omezeným průměrem - $O(n^2)$

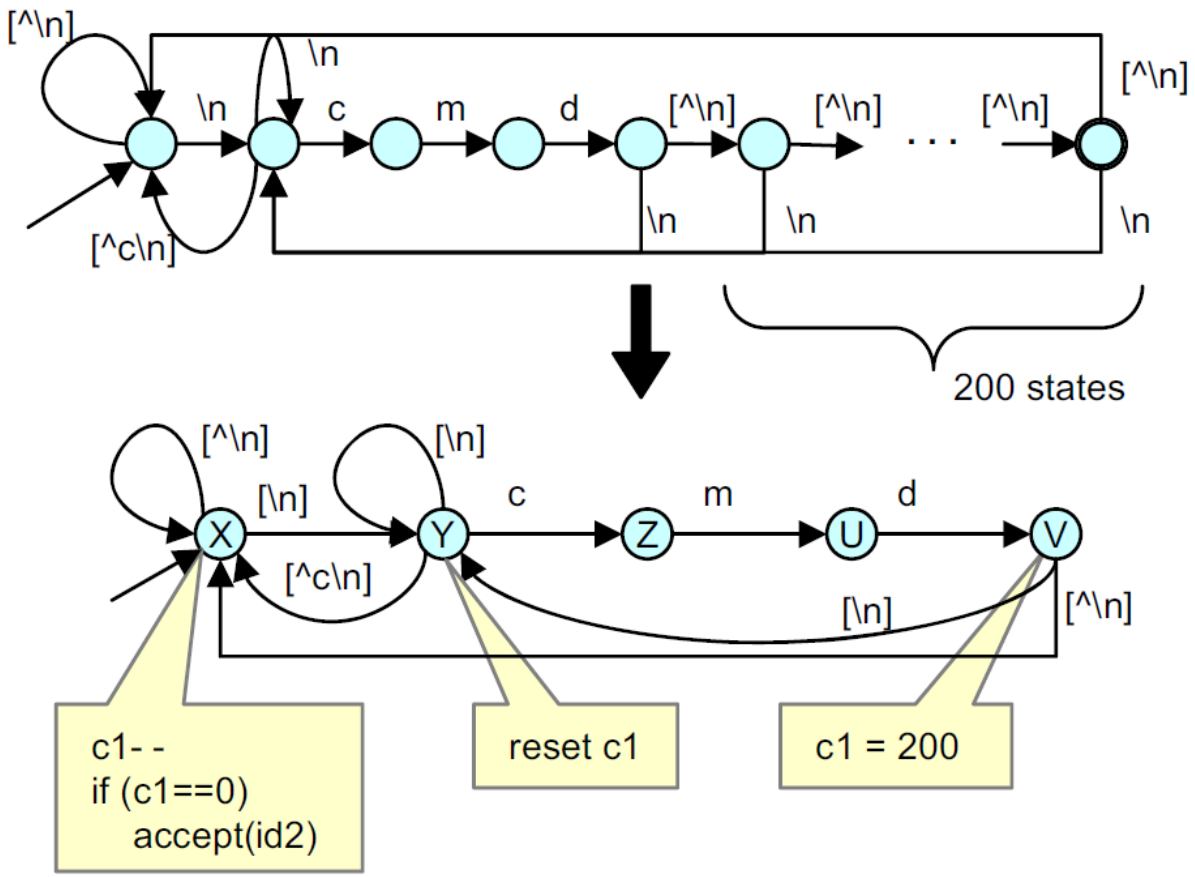
Rozšířený deterministický konečný automat

Extended Deterministic Finite Automata (XFA)

Základní myšlenka

- Umožníme stavům si zapamatovat dodatečné informace
- Zavedeme možnost modifikovat během přechodu dodatečné informace
- V koncovém stavu přijmeme řetězec pouze tehdy, mají-li dodatečné informace určitou hodnotu

Ilustrace - XFA



Vysvětlivka

XFA pro regulární výraz $\backslash ncmd[\hat{\backslash}n]\{200\}$

Definice XFA

Stavová varianta XFA

Definice

Rozšířený deterministický konečný automat je sedmice $(Q, V, \Sigma, \delta, U, (q_0, v_0), F)$

- Q - konečná množina stavů
- V - konečná množina hodnot datové domény (dodatečné informace)
- Σ - konečná vstupní abeceda
- $\delta : Q x \Sigma \rightarrow Q$ - přechodová funkce
- $U : Q x V \rightarrow V$ - funkce aktualizace datové domény, definuje jak je hodnota datové domény aktualizována v jednotlivých stavech
- (q_0, v_0) - počáteční konfigurace, sestává z:
  - q_0 - počáteční stav
  - v_0 - počáteční hodnota datové domény
- $F \subseteq Q x V$ - množina přijímajících konfigurací

Konfigurace a přechod XFA

Konfigurace

Konfigurace C $C = (q, v, w), (q, v, w) \in QxVx\Sigma^*$, kde q je aktuální stav, v je aktualní hodnota datové domény a w je dosud nezpracovaná část vstupního řetězce.

Přechod

$$\vdash \subseteq (QxVx\Sigma^*)x(QxVx\Sigma^*)$$

která je definována následovně:

$$(q, v, w) \vdash (\acute{q}, \acute{v}, \acute{w}) \Leftrightarrow w = a\acute{w} \wedge \acute{q} \in \delta(q, a) \wedge \acute{v} \in U(q, v)$$

kde $q, \acute{q} \in Q, a \in \Sigma, v, \acute{v} \in U, w, \acute{w} \in \Sigma^*$

Definice XFA

Přechodová varianta XFA

Definice

Rozšířený deterministický konečný automat je sedmice $(Q, D, \Sigma, \delta, U, (q_0, d_0), F)$

- Q - konečná množina stavů
- D - konečná množina hodnot datové domény
- Σ - konečná vstupní abeceda
- $\delta : Qx\Sigma \rightarrow Q$ - přechodová funkce
- $U : Qx\Sigma xD \rightarrow D$ - funkce aktualizace datové domény, definuje jak je hodnota datové domény aktualizována při přechodech
- (q_0, d_0) - počáteční konfigurace, sestává z:
  - q_0 - počáteční stav
  - d_0 - počáteční hodnota datové domény
- $F \subseteq QxD$ - množina přijímajících konfigurací

Definice

Rozšířený nedeterministický konečný automat je sedmice $(Q, D, \Sigma, \delta, U, QD_0, F)$

- Q - konečná množina stavů
- D - konečná množina hodnot datové domény
- Σ - konečná vstupní abeceda
- $\delta : Qx(\Sigma \cup \{\epsilon\}) \rightarrow Q$ - přechodová funkce
- $U : \delta \rightarrow 2^{DxD}$ - funkce aktualizace datové domény, definuje jak je hodnota datové domény aktualizována při přechodech
- $QD_0 \subseteq QxD$ - množina počátečních konfigurací
- $F \subseteq QxD$ - množina přijímajících konfigurací

Operace nad XFA

Operace

- Eliminace ϵ pravidel
- Determinizace
  1. Determinizace stavů
  2. Determinizace datové domény
- Minimalizace - není možná - neexistuje jediná kanonická minimální forma XFA
- Redukce
  1. Redukce stavů
  2. Redukce datové domény
- Převod přechodové formy XFA na stavovou formu XFA a naopak

Konstrukce XFA

Konstrukce

- Převod z regulárního výrazu na NXFA
- Eliminace ϵ pravidel
- Determinizace
- Redukce
- Převod přechodové formy XFA na stavovou formu XFA
- Kompozice více XFA do jednoho XFA
- Optimalizace
- Efektivní zakódování datové domény

Optimalizace XFA

Optimalizace

- Možno použít mnoho přístupů vyvinutých pro překladače
- Optimalizace datové domény
- Optimalizace aktualizační funkce (programu)

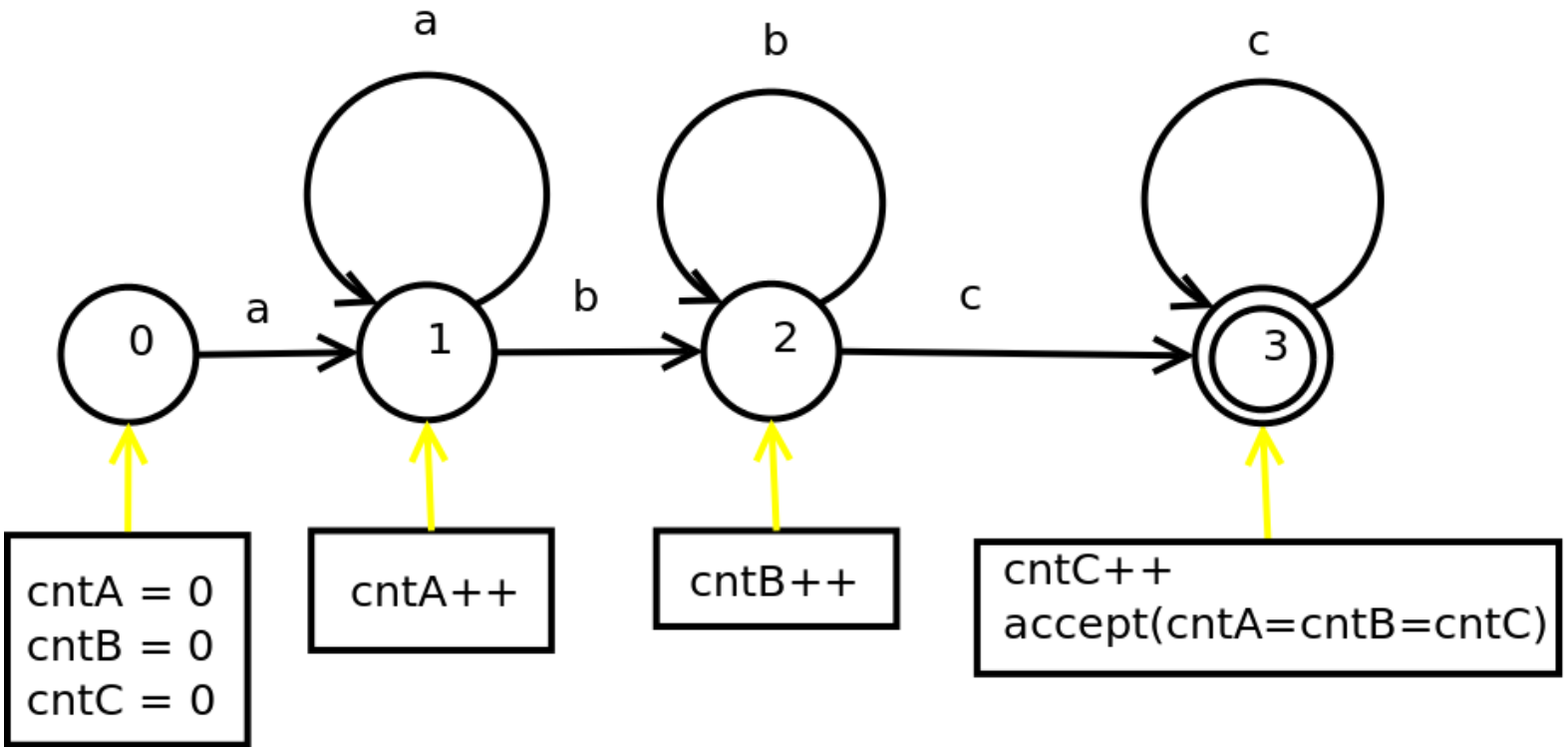
Typy

- Využívání běhových informací
- Spojování nezávislých proměnných
- Spojování instrukcí

Modifikace pro zvýšení síly

Modifikace

- Nebudeme trvat na tom, aby datová doména byla konečná



Vysvětlivka

- Modifikovaný XFA přijímající jazyk $L = \{a^n b^n c^n | n > 0\}$

Zhodnocení D2FA

Výhody

- Redukce potřebné paměti pro uložení tabulky přechodů až o 95%

Nevýhody

- Nutnost konstrukce DKA
- Snížení počtu přijímaných znaků za krok

Zhodnocení XFA

Výhody

- Redukce potřebné paměti pro uložení tabulky přechodů až o 90%
- Možnost konstrukce po jednotlivých regulárních výrazech
- Po optimalizacích rychlejší než D2FA

Nevýhody

- Náročná konstrukce celkového XFA
- Pomalé neoptimalizované XFA

Reference

D2FA

- S. Kumar et al, "Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection", in ACM SIGCOMM'06, Pisa, Italy, September 12-15, 2006.

XFA

- R. Smith, C. Estan, and S. Jha. XFA: Faster signature matching with extended automata. In IEEE Symposium on Security and Privacy, May 2008
- R. Smith et al, "Deflating the Big Bang: Fast and Scalable Deep Packet Inspection with Extended Finite Automata", in ACM SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.

Otázky