Reducing the size of NFAs by using equivalences and preorders

LUCIAN ILIE ROBERTO SOLIS-OBA SHENG YU

Department of Computer Science, University of Western Ontario
N6A 5B7, London, Ontario, CANADA
e-mails: ilie|solis|syu@csd.uwo.ca

Regular expressions

- powerful tool for describing sets of words
  - programming languages – Perl, Awk, Python
  - editors – emacs, vi
  - lexical analyzers – lex, flex
  - pattern matching tools – egrep

Applications

- pattern recognition
- text retrieval
- computational biology
- linguistics

Regular expressions – Examples

- identifiers:

  $\underbrace{\texttt{[A-Za-z]}}_{letter}\underbrace{\texttt{[A-Za-z0-9]*}}_{letters\ and\ digits}$

- unsigned numbers:

  $\underbrace{\texttt{[0-9]+}}_{digits}\underbrace{\texttt{(.[0-9]+)?}}_{optional\_fraction}\underbrace{\texttt{(E}\overbrace{\texttt{(+|-)?}}^{opt\_sign}\texttt{ [0-9]+)?}}_{optional\_exponent}$

- C comments: /⋆ ... text not including "⋆/" ... ⋆/

  ```
  /\*([^*]|\*+[^/*])*\*+/
  ```

Regular expressions – Examples

- identifiers:

$$\underbrace{\texttt{[A-Za-z]}}_{letter}\underbrace{\texttt{[A-Za-z0-9]*}}_{letters\ and\ digits}$$

- unsigned numbers:

$$\underbrace{\texttt{[0-9]+}}_{digits}\underbrace{\texttt{(.[0-9]+)?}}_{optional\_fraction}\underbrace{\texttt{(E}\overbrace{\texttt{(+|-)?}}^{opt\_sign}\texttt{ [0-9]+)?}}_{optional\_exponent}$$

- C comments: /⋆ ... text not including "⋆/" ... ⋆/

$$/\star\big((A-\{\star\})+\star^{+}(A-\{\star,/\})\big)^{*}\star^{+}/$$

Regular expressions – Implementation

regular expression

⇓

nondeterministic finite automaton (NFA)

⇓

deterministic finite automaton (DFA)

Regular expressions – Implementation

regular expression

⇓

nondeterministic finite automaton (NFA)

⇓ PROBLEM – exponential blow-up here

deterministic finite automaton (DFA)

Regular expressions – Implementation

regular expression

⇓

nondeterministic finite automaton (NFA)

⇓ SOLUTION – reduce NFA first
PROBLEM – exponential blow-up here

deterministic finite automaton (DFA)

Reducing NFAs

- NFA state minimization problem is hard (PSPACE-complete)
  - Jiang, Ravikumar, 1993

Reducing NFAs

- NFA state minimization problem is hard (PSPACE-complete)
  - Jiang, Ravikumar, 1993
- we reduce NFAs and do not minimize

Reducing NFAs

- NFA state minimization problem is hard (PSPACE-complete)
  - Jiang, Ravikumar, 1993
- we reduce NFAs and do not minimize
- idea – state merging

Reducing NFAs by merging states

- using equivalences – Ilie, Yu, 2002
- $M = (Q, A, \delta, I, F)$ is an NFA
- $\equiv_R$ – coarsest equivalence relation over Q such that:

  $(\mathcal{P}_1)$ $\equiv_R \cap (F \times (Q - F)) = \emptyset$

  $(\mathcal{P}_2)$ $\forall p, q \in Q, \forall a \in A, \left(p \equiv_R q \Rightarrow \forall q' \in \delta(q, a), \exists p' \in \delta(p, a), q' \equiv_R p'\right)$

$$\begin{array}{ccc} p & & \\ \Vert\!\vert_R & & \\ q & \xrightarrow{a} & q' \end{array} \quad \Rightarrow \quad \begin{array}{ccc} p & \xrightarrow{a} & p' \\ \Vert\!\vert_R & & \Vert\!\vert_R \\ q & \xrightarrow{a} & q' \end{array}$$

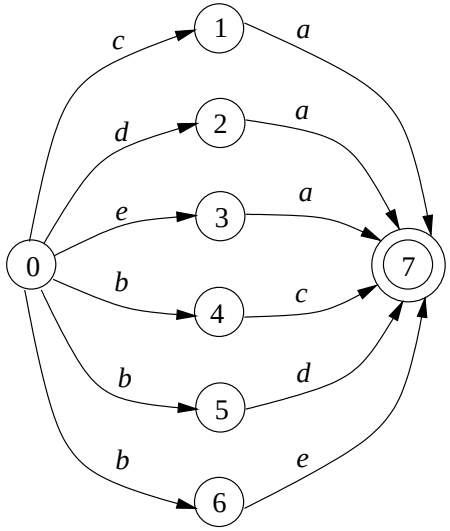
Reducing NFAs by equivalences

- $\equiv_R$ – coarsest equivalence over Q right-invariant with respect to M
- NFA reduction with $\equiv_R$ is trivial
  - merge all states in the same equivalence class
- $\equiv_L$ – defined symmetrically (using the reversed automaton)

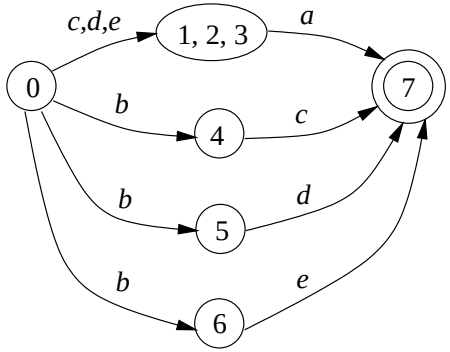
Reducing NFAs by equivalences – example

$\equiv_R$: $\{0\}$
$\{1,2,3\}$
$\{4\}, \{5\}$
$\{6\}, \{7\}$

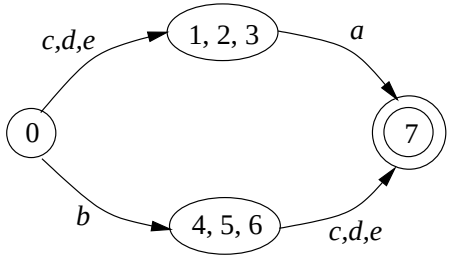
$\equiv_L$: $\{0\}, \{1\}$
$\{2\}, \{3\}$
$\{4,5,6\}$
$\{7\}$



(a)



(b)



(c)

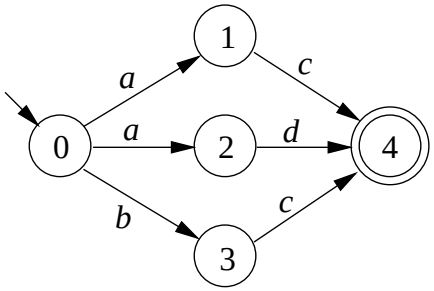
(a) An NFA
(b) Its reduced version using $\equiv_R$
(c) Its reduced version using $\equiv_R$ and $\equiv_L$

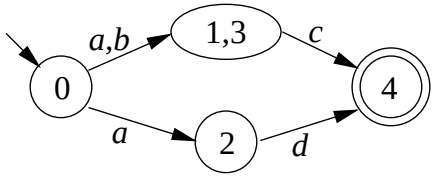
Reducing NFAs by equivalences

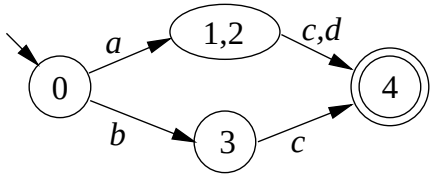
- computing $\equiv_R$ – Ilie, Navarro, Yu, 2004
  - uses a classical algorithm of Paige, Tarjan, 1987
  - automaton with n states and m transitions
  - time $\mathcal{O}(m \log n)$ and space $\mathcal{O}(m+n)$
- regular expression matching (experiments)
  - significant reduction of NFAs (10% – 40%)
  - very big reduction of resulting DFAs (by a factor of up to 10^{-6})
  - little overhead
  - faster regular expression matching
    - better than Glushkov property (Navarro, Raffinot, 2001)

Reducing NFAs by equivalences

- problem – how to use $\equiv_R$ and $\equiv_L$ optimally
- example – best reduction is not unique
  - an NFA and its reduced versions using $\equiv_R$ and $\equiv_L$







Reducing NFAs by merging states

- using preorders – Champarnaud, Coulon, 2003
- same axioms, preorders instead of equivalences
- $\subseteq_R$ – largest (w.r.t. inclusion) preorder over Q with $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$
- $\subseteq_L$ – defined symmetrically
- preorders are stronger than equivalences
  - $p \equiv_R q \quad \Rightarrow \quad p \subseteq_R q, q \subseteq_R p$
  - the converse need not be true
- computing $\subseteq_R$, $\subseteq_L$ – time $\mathcal{O}(mn)$, space $\mathcal{O}(n^2)$
  - Ilie, Navarro, Yu, 2004
  - Champarnaud, Coulon, 2004

Reducing NFAs with preorders

- reduction is complicated
  - after merging two states, we recompute $\subseteq_R$ and $\subseteq_L$
- Champarnaud, Coulon, 2004: p and q can be merged iff

  (i) $p \subseteq_R q$, $q \subseteq_R p$ or

  (ii) $p \subseteq_L q$, $q \subseteq_L p$ or

  (iii) $p \subseteq_R q$, $p \subseteq_L q$

Reducing NFAs with preorders

- reduction is complicated
  - after merging two states, we recompute $\subseteq_R$ and $\subseteq_L$
- Champarnaud, Coulon, 2004: p and q can be merged iff
  - (i) $p \subseteq_R q$, $q \subseteq_R p$ or
  - (ii) $p \subseteq_L q$, $q \subseteq_L p$ or
  - (iii) $p \subseteq_R q$, $p \subseteq_L q$
- (iii) is incorrect – Champarnaud, Coulon, 2005+

Reducing NFAs with preorders

- reduction is complicated
  - after merging two states, we recompute $\subseteq_R$ and $\subseteq_L$
- Champarnaud, Coulon, 2004: p and q can be merged iff
  - (i) $p \subseteq_R q$, $q \subseteq_R p$ or
  - (ii) $p \subseteq_L q$, $q \subseteq_L p$ or
  - (iii) $p \subseteq_R q$, $p \subseteq_L q$
- (iii) is incorrect – Champarnaud, Coulon, 2005+
- correct (iii)

  $p \subseteq_R q$, $p \subseteq_L q$, and $\mathcal{L}(p,p) = \{\varepsilon\}$ (no cycle on p)

Efficient merging of states

- equivalences – how to use $\equiv_R$ and $\equiv_L$ "optimally"
  - "optimally" – to be defined later
    - use as much information as possible from $\equiv_R$ and $\equiv_L$

- preorders – how to use $\subseteq_R$ and $\subseteq_L$ "optimally"
  - "optimally" – to be defined later
    - use as much information as possible from $\subseteq_R$ and $\subseteq_L$

Optimal use of equivalences

- $\equiv_R$ and $\equiv_L$ define the partitions

$$\Pi_R = \{X_1, X_2, \ldots, X_r\}$$
$$\Pi_L = \{X_{r+1}, X_{r+2}, \ldots, X_{r+s}\},$$

- reduction: $X^* = \{X_1^*, X_2^*, \ldots, X_\ell^*\}$
  - X_i^* – set of states merged together
  - the reduced NFA has ℓ states
- each X_i^* is a subset of some $X_{\pi(i)}$, $1 \leq \pi(i) \leq r + s$
- $\{X_{\pi(1)}, X_{\pi(2)}, \ldots, X_{\pi(\ell)}\}$ – set cover for Q from $\Pi_R \cup \Pi_L$.
- "optimal" use of equivalences – optimal solution for the instance $\langle Q, \Pi_R \cup \Pi_L \rangle$ of the set covering problem

Optimal use of equivalences

Theorem 1

Optimal use of equivalences (EQ-reduced NFA) can be computed in polynomial time.

Optimal use of equivalences

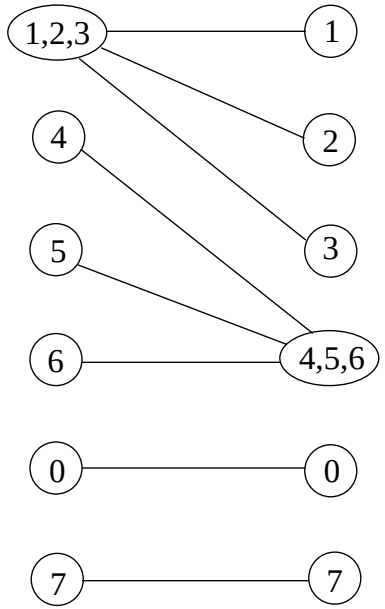
Theorem 1

*Optimal use of equivalences (*EQ*-reduced NFA) can be computed in polynomial time.*

- time $O(n^{3/2} + m \log n)$
- proof idea
  - set covering
  - minimum vertex cover in bipartite graphs
  - maximum matching in bipartite graphs
    - $\mathcal{O}(e\sqrt{v})$ (Hopcroft, Karp, 1973)

Optimal use of equivalences

$\equiv_L$: $\{0\}$
$\{1,2,3\}$
$\{4\}, \{5\}$
$\{6\}, \{7\}$



$\equiv_R$: $\{0\}, \{1\}$
$\{2\}, \{3\}$
$\{4,5,6\}$
$\{7\}$

associated bipartite graph

Optimal use of preorders

- $p \cong_R q$ iff $p \subseteq_R q$ and $q \subseteq_R p$
- $p \cong_L q$ iff $p \subseteq_L q$ and $q \subseteq_L p$
  - $\cong_R$ and $\cong_L$ – equivalence relations (coarser than $\equiv_R$ and $\equiv_L$)
  - induce the partitions π_R and π_L
- $p \preceq q$ iff $p \subseteq_R q$, $p \subseteq_L q$ and $\mathcal{L}(p,p) = \{\varepsilon\}$
  - $\preceq$ – partial order
  - induces a family $\pi_P = \{P_1, P_2, \ldots, P_k\}$ (need not be partition)
  - each P_i has a unique maximal element m_i
  - p, q belong to the same set P_i iff $p \preceq m_i$ and $q \preceq m_i$
- "optimal" use of preorders – optimal solution for the instance $\langle Q, \pi_R \cup \pi_L \cup \pi_P \rangle$ of the set covering problem

Optimal use of preorders

Theorem 2

Optimal use of preorders (PRE-reduced NFA) is NP-hard.

Optimal use of preorders

Theorem 2

Optimal use of preorders (PRE-reduced NFA) is NP-hard.

- proof idea
  - consider only the instances for which π_P is a partition
  - vertex covering problem on 3-partite hypergraphs
  - 3-SAT

Conclusions and further research

- equivalences
  - useful, easy to compute
  - easy to combine "optimally"
  - iterate our algorithm for equivalences
  - merge only some equivalent states (and then iterate)
  - stronger definition of "optimal"
- preorders
  - potentially more powerful than equivalences
  - too expensive to combine "optimally"
  - weaker definition of "optimal"
  - efficient approximation algorithms

Conclusions and further research

- experiments – compare:
  - single equivalences
  - combined equivalences
  - preorders (random merging)
  - iterated versions