Interpreting the Subset Construction Using Finite Sublanguages

Mwawi Msiska   Lynette van Zijl

Department of Mathematical Sciences
Computer Science Division
Stellenbosch University
South Africa

Prague Stringology Conference, 2016

Background

- Motivated by work on 2-way finite automata
  - Complex state-centered conversion algorithms
- Language-centered approach
- Simplest case: NFA to DFA conversion

Finite Automata

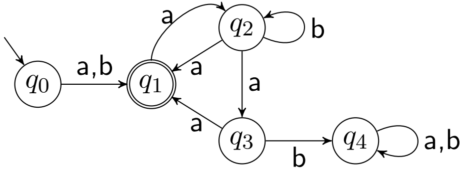
In general

- $M = (Q, \Sigma, \delta, Q_0, F)$
  - Q: finite non-empty set of states
  - Σ: finite set of symbols, the alphabet
  - δ: state transition function
  - Q_0, F: sets of initial and final states, respectively

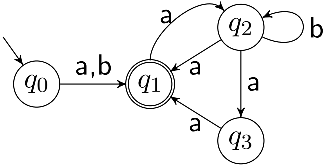
Examples

- NFA: $\delta : Q \times \Sigma \rightarrow 2^Q$
- DFA: $\delta : Q \times \Sigma \rightarrow Q$ and $|Q_0| = 1$

Finite Automata



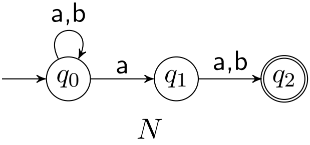
N



$\text{TRIM}(N)$

- Transition: $(p, a, q) | q \in \delta(p, a)$
- Computation:- $q_0, q_1, \ldots, q_k | (q_i, a, q_{i+1})$ is a transition for some $a \in \Sigma$ and all $0 \leq i < k$
- $\mathcal{L}(M)$: the language of the finite automaton M

Finite Exhaustive Languages



Examples

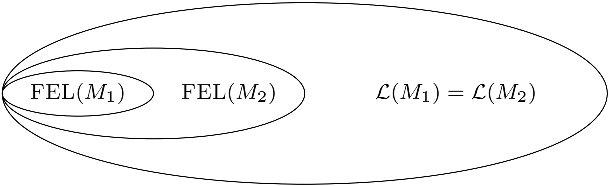
- $L_1 = \{$aa, ab, aaa, aab, baa, bab$\}$
- $L_2 = \{$aaa, aab, baa$\}$

- The following are not FELs
  - $L_3 = \{$aaa, aab, ab$\}$: missing the transition (q_0, b, q_0)
  - $L_4 = \{$aaa, aab, baa, bab, aba$\}$: aba $\notin \mathcal{L}(N)$

Equivalence

Finite automata equivalence

- Let M_1 and M_2 be finite automata
- $M_1 \equiv M_2$ iff $\mathcal{L}(M_1) = \mathcal{L}(M_2)$

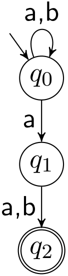


- Can we calculate $\text{FEL}(M_2)$ from M_1?
  - Yes, in the case of NFA vs. DFA

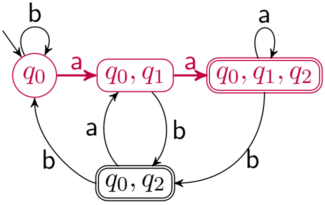
Simulations Using $\mathrm{FEL}(M)$

Example

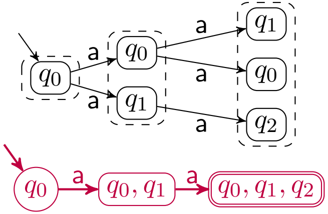
- $\mathrm{FEL}(M) = \{$**aa**, ab, aaa, aab, baa, bab, abaa, abbaa$\}$



N



M



Simulation

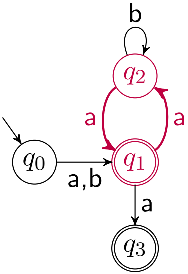
Cycles in a Computation

Definition: simple computation

- Computation that goes though any cycle at most once

Examples

1. **Non-cyclic**: q_0, q_1 **yields** {a, b}
2. **Non-cyclic**: q_0, q_1, q_3 yields {aa, ba}
3. **Cyclic**: q_0, q_1, q_2, q_1, q_3 yields {aaaa, baaa}
4. **Cyclic**: q_0, q_1, q_2, q_2, q_1 yields {aaba, baba}



- $q_0, q_1, q_2, q_1, q_2, q_2, q_1, q_3$ is not a simple computation
- W_δ: union of yields of all accepting simple computations

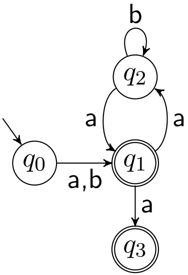
Free Cycles

A cycle is free if

- Yield of non-cyclic computation to first cycle-state does not intersect yield of any other computation, and
- Yield of any computation that completes the cycle does not intersect yield of any other computation

Example: free cycle

- q_1, q_2, q_1

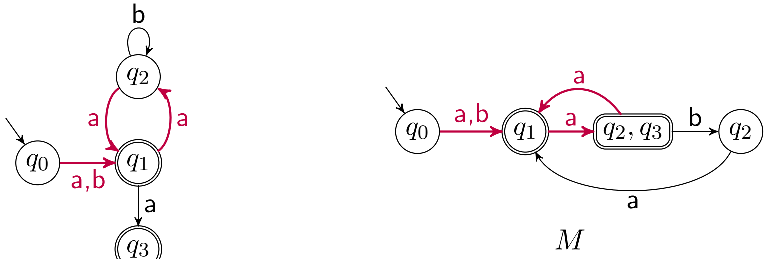


- q_2, q_2 is not free
  - $\text{YIELD}([q_0, q_1, q_2]) = \text{YIELD}([q_0, q_1, q_3])$

Free Cycles

Lemma

- *Let $C_p = p_0, p_1, \ldots, p_i, \ldots, p_j, \ldots, p_k$ be an accepting cyclic simple computation of an NFA N, where $CY = p_i, \ldots, p_j$ is a free cycle*
- *Then an equivalent DFA, M, has a computation isomorphic to C_p*



N M

Free Cycles

Theorem

- *If all cycles of an NFA* $N = (Q, \Sigma, \delta, S, F)$ *are free, then* $W_\delta = \textsc{fel}(M)$, *where* M *is the equivalent DFA*

Proof.

- All non-cyclic computations of N are trivially in M
- By previous lemma, all cyclic computations of N are also in M

□

Non-free cycles

Definition: wrap value of a non-free cycle

- Let $p_i, \ldots, p_j$ be a non-free cycle
- $r = \text{WRAP}(p_i, \ldots, p_j)$ if $p_i, (p_{i+1}, \ldots, p_j)^r$ is a non-free cycle and $p_i, (p_{i+1}, \ldots, p_j)^{r+1}$ is a free cycle

Lemma

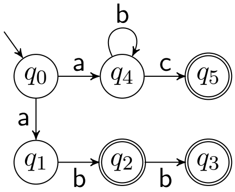
- *If a cycle has a finite wrap value, r, then $\text{FEL}(M)$ contains the yield of accepting computations that complete the cycle up to $(r+2)$ times*

- Finite wrap value indicates interaction between a cycle and a non-cyclic computation

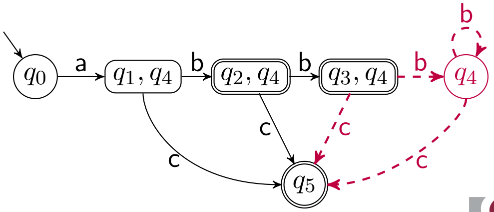
Non-Free Cycles: finite wrap value

Example

- $W_\delta = \{\text{ac, abb, abc}\}$
- $W^{2+2} = \{\text{abb}\textbf{c}, \text{abb}\textbf{bc}, \text{abb}\textbf{bbc}\}$
- $\textsc{fel}(M) = W_\delta \cup W^{2+2}$



N



M

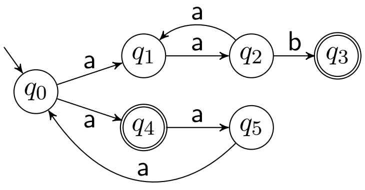
Non-Free Cycles: infinite wrap

- What if the wrap value is infinite?
  - Indicates interaction of several cycles – **overlap**

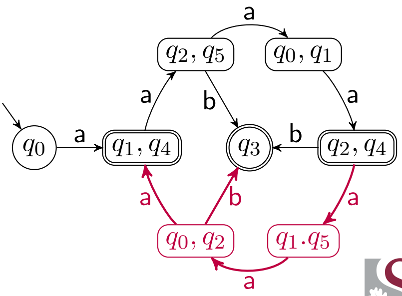
Overlapping Cycles

Example

- $W_\delta = \{$a, aab, aaaa, aaaab$\}$
- $W_{\text{overlap}} = \{$aaaa**aaa**, aaaa**aab**, aaaa**aaa**ab$\}$



N



M

Overlapping Cycles

Lemma

- *Let* $C_1, C_2, \ldots, C_m$ *be overlapping cycles in an NFA* N
- *Let* $l_i = |C_i| - 1$ *for all* $1 \leq i \leq m$
- *Then an equivalent DFA* M *has a cycle* C*, such that* $g = |C| - 1 = \textsc{lcm}(l_1, l_2, \ldots, l_m)$
- $\textsc{fel}(M)$ *includes NFA computations that traverse* C_i $2, \ldots, (g/l_i)$ *times*

Corollary

- *The length of the longest cycle in a DFA equivalent to an* n*-state NFA is* $O(e^{\sqrt{n \log n}})$

Conclusion

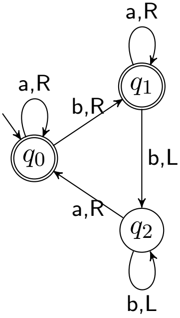
- Demonstrated an alternative to the subset construction
  - More complex derivation of $\mathrm{FEL}(M)$
  - The conversion algorithm is simple, once we have $\mathrm{FEL}(M)$
- Framework for other conversions
  - Effect of cycles on $\mathrm{FEL}(M_2)$
- The language-centered approach is more general than the state-centered approach

- Thank you

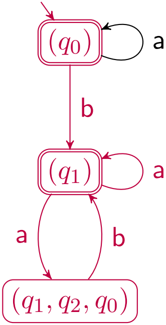
2-Way DFA to NFA Conversion

Example

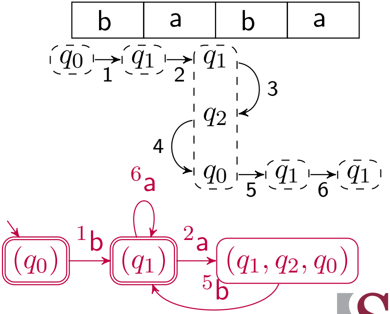
- $\text{FEL}(N) = \{\epsilon, \text{a}, \textbf{baba}\}$



T



N



Simulation