Determinization of Büchi Automata

王伟锋

计算机科学国家重点实验室
中科院软件所

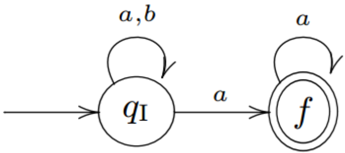
2013 年 5 月 17 日

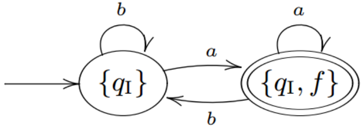
Outline

Background

- The conversion from NFA to DFA uses powerset construction.
- But for Büchi automata, the naive powerset method doesn't work.
- We have already known that this can be done by $NBA \rightarrow SDBA \rightarrow DMA$.
- Safra's construction is another method (more efficient).

Powerset method for $L := \{\alpha \in \{a, b\}^\omega | \sharp_b(\alpha) < \infty\}$





Powerset method doesn't work

- Can we fix the problem by modifying the powerset method?
- The following theorem gives the answer: no.

Powerset method doesn't work

Theorem 1

There exists languages which are accepted by some nondeterministic Buchi automaton but not by any deterministic Buchi automaton.

Proof sketch.

Consider the language $L = \{\alpha \in \Sigma^\omega | \sharp_b(\alpha) < \infty\}$ where $\Sigma = \{a, b\}$. Suppose L can be accepted by some DBA, by the fact that $\forall \sigma \in \Sigma^* \cdot \sigma a^\omega \in L$, we can construct an infinite word $b^{i_1} a^{j_1} b^{i_2} a^{j_2} \cdots$ which is also accepted by this DBA. Thus we get a contradict. □

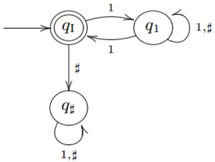
Switch to Rabin or Muller

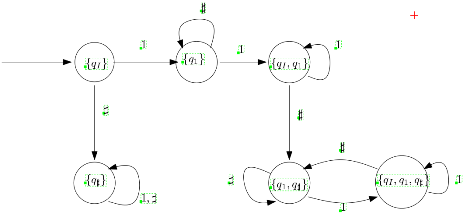
- So, in order to get a deterministic version of a Buchi automata, we should switch to other kinds of ω-automata, e.g. Rabin or Muller Automata (Recall the Rabin and Muller acceptance conditions).
- However, the naive powerset construction still doesn't work.

Recall Rabin and Muller condition

- Rabin:
  - pairs of sets of states (E_i, F_i)
  - ρ is accepting iff there exists an i such that $\mathit{Inf}(\rho) \cap E_i = \emptyset$ and $\mathit{Inf}(\rho) \cap F_i \neq \emptyset$
- Muller:
  - sets of states F_i
  - ρ is accepting iff there exists an i such that $\mathit{Inf}(\rho) = F_i$.

Consider Rabin Condition for $1(11\sharp)^\omega$

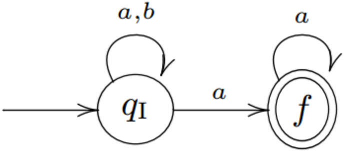


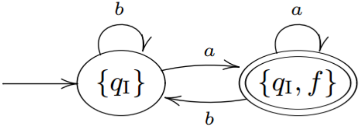


- The weakness of the powerset construction is that the resulting automaton allows for too many "accepting" runs.
- Given a sequence of macrostates, it might be impossible to extract a run of the original Buchi automaton.
- Safra's key idea is to modify the powerset construction so that it satisfies the above condition.

Recall the first example

$$L := \{\alpha \in \{a, b\}^\omega | \sharp_b(\alpha) < \infty\}$$



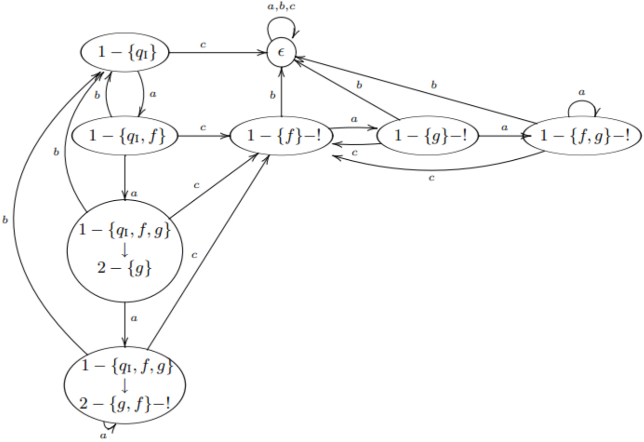


Outline

Overview of Safra's construction

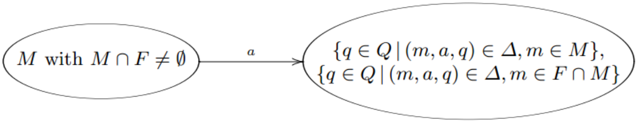
- Each macrostate is a tree (instead of a subset of states in powerset construction).
- Each node of the tree is a subset of states.

How it looks like



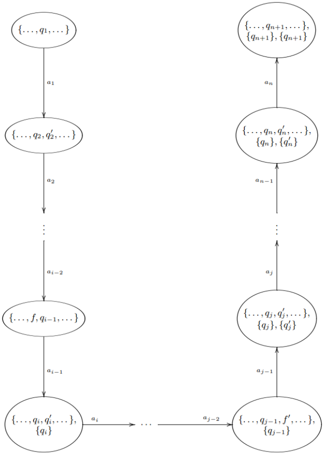
- Trick1: Initialize new runs of macrostates starting from recurring states.
  - Every state in an extra component has a recurring state as predecessor.

Illustration for Trick1



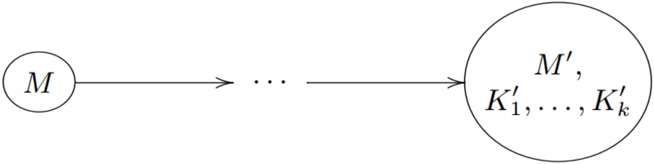
- Trick2: Keep track of joining runs of the nondeterministic Buchi automaton just once.
  - Each macrostate of size n has at most n children.

Illustration for Trick2



- Trick3: If all states in a macrostate have a recurring state as predecessor, delete the corresponding components.
  - Each macrostate of size n is at most n high.

Illustration for Trick3



Outline

- Condition 1: The union of brother macrostates is a proper subset of their parent macrostates.
- Condition 2: Brother macrostates are disjoint.
- The number of nodes in a safra tree is bounded by $|Q|$.

Outline

The Construction

- $V := \{1, 2, \ldots, 2|Q|\}$ as IDs for nodes.
- Q the state space of Büchi automata.
- Why we need $2|Q|$ IDs? (We will see this later)

The Construction

- The initial state q_I' is the Safra tree consisting of the single node 1 labeled with macrostate $\{q_I\}$.

The Construction

- The value of the transition function $\delta(T, a)$ for a given input $a \in \Sigma$ and a Safra tree T with a set N of nodes is computed as follows step by step:
  - **Step 1:** Remove all marks '!' in the Safra tree T.
  - **Step 2:** For every node v with macrostate M such that $M \cap F \neq \emptyset$, create a new node $v' \in (V \setminus N)$, such that v' becomes the youngest son of v and carries the macrostate $M \cap F$.
  - **Step 3:** Apply the powerset construction on every node v, i.e. replace its macrostate M by $\{q \in Q | \exists (m, a, q) \in \Delta : m \in M\}$

The Construction

- The value of the transition function $\delta(T, a)$ for a given input $a \in \Sigma$ and a Safra tree T with a set N of nodes is computed as follows step by step:
  - **Step 4 (horizontal merge):** For every node v with macrostate M and state $q \in M$, such that q also belongs to an older brother of v, remove q from M.
  - **Step 5:** Remove all nodes with empty macrostates.
  - **Step 6 (vertical merge):** For every node whose label is equal to the union of the labels of its sons, remove all the descendants of v and mark v with '!'.

The Construction

- The set of states Q' consists of all reachable Safra trees.

The Construction - acceptance condition

- *Muller* condition: A set $S \subseteq Q'$ of Safra trees is in the system $\mathcal{F}$ of final state sets if for some node $v \in V$ the following holds:

  Muller 1: v appears in all Safra trees of S, and

  Muller 2: v is marked at least once in S.

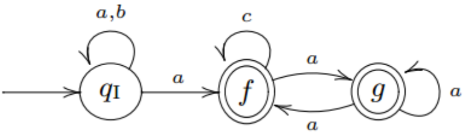
The Construction - acceptance condition

- *Rabin* condition: A pair (E_v, F_v), $v \in V$, where

  Rabin 1: E_v consists of all Safra trees without a node v, and

  Rabin 2: F_v consists of all Safra trees with node v marked '!'.

Example



Example

Computing $\delta(1 - \{q_I\}, a)$:

Step 1	Step 2	**Step 3**
$1 - \{q_I\}$	$1 - \{q_I\}$	$1 - \{q_I, f\}$

Computing $\delta(1 - \{q_I\}, c)$:

Step 1	Step 2	**Step 3**	Step 4	**Step 5**
$1 - \{q_I\}$	$1 - \{q_I\}$	$1 - \emptyset$	$1 - \emptyset$	ϵ

Computing $\delta(1 - \{q_I, f\}, c)$:

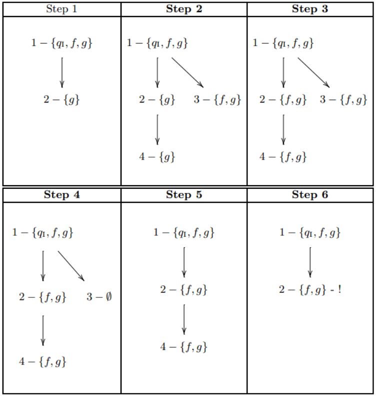
Step 1	**Step 2**	**Step 3**	Step 4	Step 5	**Step 6**
$1 - \{q_I, f\}$	$1 - \{q_I, f\}$ ↓ $2 - \{f\}$	$1 - \{f\}$ ↓ $2 - \{f\}$	$1 - \{f\}$ ↓ $2 - \{f\}$	$1 - \{f\}$ ↓ $2 - \{f\}$	$1 - \{f\}$ - !

Example

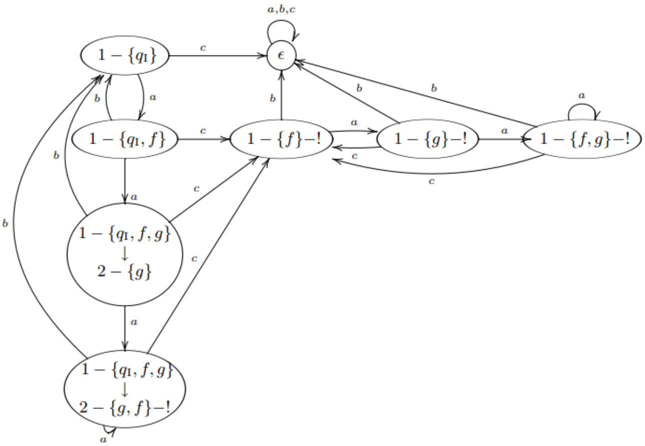
Computing $\delta(1 - \{q_1, f, g\}, a)$:

$$\downarrow$$

$$2 - \{g, f\} - !$$



Example


a,b,c
1 – {q1}
c
ϵ
b
a
b
1 – {q1, f}
c
1 – {f}–!
a
1 – {g}–!
a
1 – {f, g}–!
b
b
a
c
c
b
a
c
1 – {q1, f, g}
↓
2 – {g}
b
c
a
1 – {q1, f, g}
↓
2 – {g, f}–!
a

Correctness

- For a Buchi Automaton $\mathcal{B}$, and the Muller Automaton $\mathcal{M}$ obtained via Safra's Construction:
  - Completeness : $L(\mathcal{B}) \subseteq L(\mathcal{M})$
  - Soundness : $L(\mathcal{M}) \subseteq L(\mathcal{B})$

Completeness ($L(\mathcal{B}) \subseteq L(\mathcal{M})$)

Proof.

Let $\alpha \in L(\mathcal{B})$, then there exists one run ρ' of α on $\mathcal{M}$. We claim there is at least one node v in the Safra trees of ρ' such that
(i) v-from certain point on- is a node of all Safra trees in ρ' and
(ii) v is marked '!' infinitely often.
(Proof: Consider candidates from the root node to its children.)
From the two claims, we can conclude that ρ' is an accepting run of $\mathcal{M}$, consequently $L(\mathcal{B}) \subseteq L(\mathcal{M})$ holds. □

Soundness ($L(\mathcal{M}) \subseteq L(\mathcal{B})$)

Theorem 2 (König's Infinity Lemma)

An infinite rooted tree which is finitely branching has an infinite path.

Soundness ($L(\mathcal{M}) \subseteq L(\mathcal{B})$)

Proof.

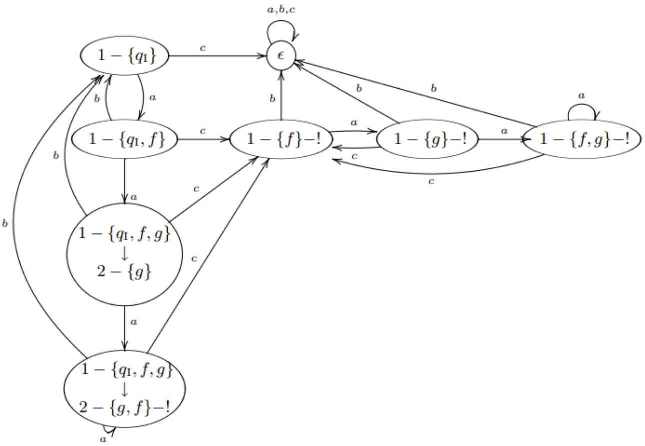
Since ρ' is an accepting run of α over $\mathcal{M}$, it satisfies the (Muller) accepting conditions.

Consider two Safra trees T and U in the sequence ρ', on which v is marked '!', and v is not marked '!' on any other tree between T and U.(How the child nodes appeared and then disappeared?)

We construct fragment runs (passing some recurring state) from the two v's. By König's Lemma, there is an infinite run in the NBA. □

- Why we need $2|Q|$ IDs? (Answer: for correctness)
  - If v is merged into other nodes, then the ID cannot appear in the next tree.
  - So the continuity of the ID means the continuity of the node.
  - With just $|Q|$ IDs, we cannot achieve that effect.

Example: Consider the state sequence by $ac(aac)^\omega$

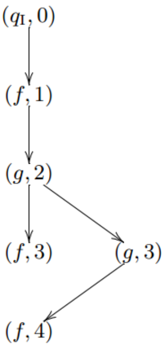


Example

The word $ac(aac)^\omega$ leads to the following sequence of macrostates:

- $S_0 = \{q_I\}$
- $S_{3i+1} = \{f\}, i \geq 0$
- $S_{3i+2} = \{g\}, i \geq 0$
- $S_{3i+3} = \{f, g\}, i \geq 0.$

Example



Outline

Theorem 3

Safra's construction converts a nondeterministic Büchi automaton with n states into a deterministic Muller automaton or into a deterministic Rabin automaton with $2^{O(n*log(n))}$ states.

Proof.

We describe an encoding of a Safra tree

- For each q in NBA, we record the deepest node that contains q, resulting in a function $\{q_1, \ldots, q_n\} \rightarrow \{0, 1, 2, \ldots, 2n\}$.
- Parent relation : $\{1, 2, \ldots, 2n\} \rightarrow \{0, 1, 2, \ldots, 2n\}$
- Next-older brother relation: $\{1, 2, \ldots, 2n\} \rightarrow \{0, 1, 2, \ldots, 2n\}$
- The marks '!': $\{1, 2, \ldots, 2n\} \rightarrow \{0, 1\}$

Each Safra tree correspond to a combination of such encoding, and the number of combinations of such encodings is bounded by $(2n+1)^{n+3*2n} = (2n+1)^{7n} = 2^{log((2n+1)^{7n})} = 2^{7n(log(2n+1))} \in 2^{O(nlog(n))}$

□

Corollary 4 (Optimality of Safra's Construction)

There is no conversion of Büchi automata with n states into deterministic Rabin automata with $2^{O(n)}$ states.

It is open whether Safra's construction can be improved for Muller automata.

Thanks!