Notions of Automata Theory

Automata on Finite Words

A *non-deterministic finite automaton* (NFA) over Σ is a tuple $A = \langle S, I, T, F \rangle$ where:

- S is a finite set of *states*,
- $I \subseteq S$ is a set of *initial states*,
- $T \subseteq S \times \Sigma \times S$ is a *transition relation*,
- $F \subseteq S$ is a set of *final states*.

We denote $T(s, \alpha) = \{s' \in S \mid (s, \alpha, s') \in T\}$. When T is clear from the context we denote $(s, \alpha, s') \in T$ by $s \xrightarrow{\alpha} s'$.

Determinism and Completeness

Definition 1 *An automaton* $A = \langle S, I, T, F \rangle$ *is* deterministic *(DFA) iff* $\|I\| = 1$ *and, for each* $s \in S$ *and for each* $\alpha \in \Sigma$, $\|T(s, \alpha)\| \leq 1$.

If A is deterministic we write $T(s, \alpha) = s'$ instead of $T(s, \alpha) = \{s'\}$.

Definition 2 *An automaton* $A = \langle S, I, T, F \rangle$ *is* complete *iff for each* $s \in S$ *and for each* $\alpha \in \Sigma$, $\|T(s, \alpha)\| \geq 1$.

Runs and Acceptance Conditions

Given a finite word $w \in \Sigma^*$, $w = \alpha_1\alpha_2\ldots\alpha_n$, a *run* of A over w is a finite sequence of states $s_1, s_2, \ldots, s_n, s_{n+1}$ such that $s_1 \in I$ and $s_i \xrightarrow{\alpha_i} s_{i+1}$ for all $1 \leq i \leq n$.

A run over w between s_i and s_j is denoted as $s_i \xrightarrow{w} s_j$.

The run is said to be *accepting* iff $s_{n+1} \in F$. If A has an accepting run over w, then we say that A *accepts* w.

The language of A, denoted $\mathcal{L}(A)$ is the set of all words accepted by A.

A set of words $S \subseteq \Sigma^*$ is *recognizable* if there exists an automaton A such that $S = \mathcal{L}(A)$.

Determinism, Completeness, again

Proposition 1 *If A is deterministic, then it has at most one run for each input word.*

Proposition 2 *If A is complete, then it has at least one run for each input word.*

Determinization

Theorem 1 *For every NFA A there exists a DFA A_d such that $\mathcal{L}(A) = \mathcal{L}(A_d)$.*

Let $A_d = \langle 2^S, \{I\}, T_d, \{G \subseteq S \mid G \cap F \neq \emptyset\}\rangle$, where

$$(S_1, \alpha, S_2) \in T_d \iff S_2 = \{s' \mid \exists s \in S_1 \ . \ (s, \alpha, s') \in T\}$$

This definition is known as subset construction

On the Exponential Blowup of Complementation

Theorem 2 *For every $n \in \mathbb{N}$, $n \geq 1$, there exists an automaton A, with $size(A) = n + 1$ such that no deterministic automaton with less than 2^n states recognizes the complement of $\mathcal{L}(A)$.*

Let $\Sigma = \{a, b\}$ and $L = \{uav \mid u, v \in \Sigma^*, |v| = n - 1\}$.

There exists a NFA with exactly $n + 1$ states which recognizes L.

Suppose that $B = \langle S, \{s_0\}, T, F\rangle$, is a (complete) DFA with $\|S\| < 2^n$ that accepts $\Sigma^* \setminus L$.

On the Exponential Blowup of Complementation

$\|\{w \in \Sigma^* \mid |w| = n\}\| = 2^n$ and $\|S\| < 2^n$ (by the pigeonhole principle)

$\Rightarrow \exists uav_1, ubv_2 \,.\, |uav_1| = |ubv_2| = n$ and $s \in S \,.\, s_0 \xrightarrow{uav_1} s$ and $s_0 \xrightarrow{ubv_2} s$

Let s_1 be the (unique) state of B such that $s \xrightarrow{u} s_1$.

Since $|uav_1| = n$, then $uav_1u \in L \Rightarrow uav_1u \notin \mathcal{L}(B)$, i.e. s is not accepting.

On the other hand, $ubv_2u \notin L \Rightarrow ubv_2u \in \mathcal{L}(B)$, i.e. s is accepting, contradiction.

Completion

Lemma 1 *For every NFA A there exists a complete NFA A_c such that $\mathcal{L}(A) = \mathcal{L}(A_c)$.*

Let $A_c = \langle S \cup \{\sigma\}, I, T_c, F\rangle$, where $\sigma \notin S$ is a new sink state. The transition relation T_c is defined as:

$$\forall s \in S \forall \alpha \in \Sigma \;.\; (s, \alpha, \sigma) \in T_c \iff \forall s' \in S \;.\; (s, \alpha, s') \notin T$$

and $\forall \alpha \in \Sigma \;.\; (\sigma, \alpha, \sigma) \in T_c$.

Closure Properties

Theorem 3 *Let* $A_1 = \langle S_1, I_1, T_1, F_1 \rangle$ *and* $A_2 = \langle S_2, I_2, T_2, F_2 \rangle$ *be two NFA. There exists automata* $\bar{A}_1$, $A_\cup$ *and* $A_\cap$ *that recognize the languages* $\Sigma^* \setminus \mathcal{L}(A_1)$, $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$, *and* $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ *respectivelly.*

Let $A' = \langle S', I', T', F' \rangle$ be the complete deterministic automaton such that $\mathcal{L}(A_1) = \mathcal{L}(A')$, and $\bar{A}_1 = \langle S', I', T', S' \setminus F' \rangle$.

Let $A_\cup = \langle S_1 \cup S_2, I_1 \cup I_2, T_1 \cup T_2, F_1 \cup F_2 \rangle$.

Let $A_\cap = \langle S_1 \times S_2, I_1 \times I_2, T_\cap, F_1 \times F_2 \rangle$ where:

$$(\langle s_1, t_1 \rangle, \alpha, \langle s_2, t_2 \rangle) \in T_\cap \iff (s_1, \alpha, s_2) \in T_1 \text{ and } (t_1, \alpha, t_2) \in T_2$$

Decidability

Given automata A and B:

- **Membership** Given $w \in \Sigma^*$, $w \in \mathcal{L}(A)$?
- **Emptiness** $\mathcal{L}(A) = \emptyset$?
- **Equality** $\mathcal{L}(A) = \mathcal{L}(B)$?
- **Infinity** $\|\mathcal{L}(A)\| < \infty$?
- **Universality** $\mathcal{L}(A) = \Sigma^*$?

Theorem 4 *The emptiness, equality, infinity and universality problems are decidable for automata on finite words.*

Automata on Infinite Words

Definition of Büchi Automata

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet.

A *non-deterministic Büchi automaton* (NBA) over Σ is a tuple $A = \langle S, I, T, F \rangle$, where:

- S is a finite set of *states*,
- $I \subseteq S$ is a set of *initial states*,
- $T \subseteq S \times \Sigma \times S$ is a *transition relation*,
- $F \subseteq S$ is a set of *final states*.

Acceptance Condition

A *run* of a Büchi automaton is defined over an infinite word $w : \alpha_1\alpha_2\ldots$ as an infinite sequence of states $\pi \ : \ s_0s_1s_2\ldots$ such that:

- $s_0 \in I$ and
- $(s_i, \alpha_{i+1}, s_{i+1}) \in T$, for all $i \in \mathbb{N}$.

$$\boxed{\mathrm{inf}(\pi) = \{s \mid s \text{ appears infinitely often on } \pi\}}$$

Run π of A is said to be *accepting* iff $\mathrm{inf}(\pi) \cap F \neq \emptyset$.

The language of A, denoted $\mathcal{L}(A)$, is the set of all words accepted by A.

A language $L \subseteq \Sigma^\omega$ is *recognizable* (or, equivalently *rational*) if there exists a Büchi automaton A such that $L = \mathcal{L}(A)$.

Examples

Let $\Sigma = \{0, 1\}$. Define Büchi automata for the following languages:

1. $L = \{\alpha \in \Sigma^\omega \mid 0 \text{ occurs in } \alpha \text{ exactly once}\}$
2. $L = \{\alpha \in \Sigma^\omega \mid \text{after each } 0 \text{ in } \alpha \text{ there is } 1\}$
3. $L = \{\alpha \in \Sigma^\omega \mid \alpha \text{ contains finitely many 1's}\}$
4. $L = (01)^*\Sigma^\omega$
5. $L = \{\alpha \in \Sigma^\omega \mid 0 \text{ occurs on all even positions in } \alpha\}$

Closure Properties

Closure under union is like in the finite automata case.

Intersection is a bit special.

Complementation of non-deterministic Büchi automata is a complex result.

Deterministic BA are not closed under complement

Closure under Intersection

Let $A_1 = \langle S_1, I_1, T_1, F_1 \rangle$ and $A_2 = \langle S_2, I_2, T_2, F_2 \rangle$

Build $A_\cap = \langle S, I, T, F \rangle$:

- $S = S_1 \times S_2 \times \{1, 2, 3\}$,
- $I = I_1 \times I_2 \times \{1\}$,
- the definition of T is the following:
  - $((s_1, s_1', 1), a, (s_2, s_2', 1)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_1 \notin F_1$
  - $((s_1, s_1', 1), a, (s_2, s_2', 2)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_1 \in F_1$
  - $((s_1, s_1', 2), a, (s_2, s_2', 2)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_1' \notin F_2$
  - $((s_1, s_1', 2), a, (s_2, s_2', 3)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_1' \in F_2$
  - $((s_1, s_1', 3), a, (s_2, s_2', 1)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$
- $F = S_1 \times S_2 \times \{3\}$

The Emptiness Problem

Theorem 5 *Given a Büchi automaton A, $\mathcal{L}(A) \neq \emptyset$ iff there exist $u, v \in \Sigma^*$, $|u|, |v| \leq \|A\|$, such that $uv^\omega \in \mathcal{L}(A)$.*

In practical terms, A is non-empty iff there exists a state s which is reachable both from an initial state and from itself.

Q: Is the membership problem decidable for Büchi automata?

Deterministic Büchi Automata

$$\boxed{\omega\text{-languages recognized by NBA} \supset \omega\text{-languages recognized by DBA}}$$

Q: Why classical subset construction does not work for Büchi automata?

Let $A = \langle S, I, T, F \rangle$ and $A_d = \langle 2^S, \{I\}, T_d, \{Q \mid Q \cap F \neq \emptyset\} \rangle$.

Let $u_0 u_1 u_2 \ldots \in \mathcal{L}(A)$ be an infinite word. In A_d this gives:

$$I \xrightarrow{u_0} Q_1 \xrightarrow{u_1} Q_2 \xrightarrow{u_2} \ldots$$

where each $Q_i \cap F$. However this does not necessarily correspond to an accepting path in A!

Deterministic Büchi Automata

Let $W \subseteq \Sigma^*$. Define $\overrightarrow{W} = \{\alpha \in \Sigma^\omega \mid \alpha(0,n) \in W \text{ for infinitely many } n\}$

Theorem 6 *A language $L \subseteq \Sigma^\omega$ is recognizable by a deterministic Büchi automaton iff there exists a rational language $W \subseteq \Sigma^*$ such that $L = \overrightarrow{W}$.*

If $L = \mathcal{L}(A)$ then $W = \mathcal{L}(A')$ where A' is the DFA with the same definition as A, and with the finite acceptance condition.

Deterministic Büchi Automata

Theorem 7 *There exists a Büchi recognizable language that can be recognized by no deterministic Büchi automaton.*

$\Sigma = \{a, b\}$ and $L = \{\alpha \in \Sigma^\omega \mid \#_a(\alpha) < \infty\} = \Sigma^* b^\omega$.

Suppose $L = \overrightarrow{W}$ for some $W \subseteq \Sigma^*$.

$b^\omega \in L \Rightarrow b^{n_1} \in W$

$b^{n_1} a b^\omega \in L \Rightarrow b^{n_1} a b^{n_2} \in W$

$\cdots$

$b^{n_1} a b^{n_2} a \ldots \in \overrightarrow{W} = L$, contradiction.

Deterministic BA are not closed under complement

Theorem 8 *There exists a DBA A such that no DBA recognizes the language $\Sigma^\omega \setminus \mathcal{L}(A)$.*

$\Sigma = \{a, b\}$ and $L = \{\alpha \in \Sigma^\omega \mid \#_a(\alpha) < \infty\} = \Sigma^* b^\omega$.

Let $V = \Sigma^* a$. There exists a DFA A such that $\mathcal{L}(A) = V$.

There exists a deterministic Büchi automaton B such that $\mathcal{L}(A) = \overrightarrow{V}$

But $\Sigma^\omega \setminus \overrightarrow{V} = L$ which cannot be recognized by any DBA.

Complementation of non-deterministic BA

- Languages recognized by non-deterministic BA are closed under complement
- Original proof by Büchi using Ramsey Theorem
- Optimal $2^{O(n \log n)}$ complexity by Safra Algorithm
- Lower bound of $n!$

LTL Model Checking

System verification using LTL

- Let K be a model of a reactive system (finite computations can be turned into infinite ones by repeating the last state infinitely often)
- Given an LTL formula φ over a set of atomic propositions $\mathcal{P}$, specifying all bad behaviors, we build a Büchi automaton A_φ that accepts all sequences over $2^{\mathcal{P}}$ satisfying φ.
- Check whether $\mathcal{L}(A_\varphi) \cap \mathcal{L}(K) = \emptyset$. In case it is not, we obtain a counterexample.
- Alternatively, if φ specifies all good behaviors, we check $\mathcal{L}(A_{\neg\varphi}) \cap \mathcal{L}(K) = \emptyset$.

Generalized Büchi Automata

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet.

A *generalized Büchi automaton* (GBA) over Σ is $A = \langle S, I, T, \mathcal{F} \rangle$, where:

- S is a finite set of *states*,
- $I \subseteq S$ is a set of *initial states*,
- $T \subseteq S \times \Sigma \times S$ is a *transition relation*,
- $\mathcal{F} = \{F_1, \ldots, F_k\} \subseteq 2^S$ is a set of *sets of final states*.

A run π of a GBA is said to be *accepting* iff, for all $1 \leq i \leq k$, we have

$$\inf(\pi) \cap F_i \neq \emptyset$$

GBA and BA are equivalent

Let $A = \langle S, I, T, \mathcal{F} \rangle$, where $\mathcal{F} = \{F_1, \ldots, F_k\}$.

Build $A' = \langle S', I', T', F' \rangle$:

- $S' = S \times \{1, \ldots, k\}$,
- $I' = I \times \{1\}$,
- $(\langle s, i \rangle, a, \langle t, j \rangle) \in T'$ iff $(s, t) \in T$ and:
  - $j = i$ if $s \notin F_i$,
  - $j = (i \mod k) + 1$ if $s \in F_i$.
- $F' = F_1 \times \{1\}$.

The idea of the construction

Let $K = \langle S, s_0, \rightarrow, L\rangle$ be a Kripke structure over a set of atomic propositions $\mathcal{P}$, $\pi : \mathbb{N} \rightarrow S$ be an infinite path through K, and φ be an LTL formula.

To determine whether $K, \pi \models \varphi$, we label π with sets of subformulae of φ in a way that is compatible with LTL semantics.

Then $K, \pi \models \varphi$ if such a labeling exists

Negation Normal Form

- Negation occurs only on atomic propositions

$$
\begin{aligned}
\neg(\varphi \mathcal{U} \psi) &= \neg\varphi \mathcal{R} \neg\psi \\
\neg(\varphi \mathcal{R} \psi) &= \neg\varphi \mathcal{U} \neg\psi \\
\neg\Box\varphi &= \Diamond\neg\varphi \\
\neg\Diamond\varphi &= \Box\neg\varphi
\end{aligned}
$$

- Example

$$\neg\Box p \vee \Diamond(\neg(a\mathcal{U}b \wedge \Box c)) = \Diamond\neg p \vee \Diamond(\neg a\mathcal{R}\neg b \vee \Diamond\neg c)$$

Closure

Let φ be an LTL formula written in negation normal form.

The *closure* of φ is the set $Cl(\varphi) \in 2^{\mathcal{L}(LTL)}$:

- $\varphi \in Cl(\varphi)$
- $\bigcirc\psi \in Cl(\varphi) \Rightarrow \psi \in Cl(\varphi)$
- $\psi_1 \bullet \psi_2 \in Cl(\varphi) \Rightarrow \psi_1, \psi_2 \in Cl(\varphi)$, for all $\bullet \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$.

Example 1 $Cl(\Diamond p) = Cl(\top \mathcal{U} p) = \{\Diamond p, p, \top\}\Box$

Q: What is the size of the closure relative to the size of φ ?

Labeling rules

Given a path $\pi : \mathbb{N} \rightarrow 2^{\mathcal{P}}$ in a Kripke structure $K = \langle S, s_0, \rightarrow, L \rangle$ and φ, we define the labeling $\tau : \mathbb{N} \rightarrow 2^{Cl(\varphi)}$ as follows:

- for $p \in \mathcal{P}$, if $p \in \tau(i)$ then $p \in \pi(i)$, and if $\neg p \in \tau(i)$ then $p \notin \pi(i)$
- if $\psi_1 \wedge \psi_2 \in \tau(i)$ then $\psi_1 \in \tau(i)$ and $\psi_2 \in \tau(i)$
- if $\psi_1 \vee \psi_2 \in \tau(i)$ then $\psi_1 \in \tau(i)$ or $\psi_2 \in \tau(i)$

Labeling rules

$$\begin{array}{rcl} \varphi \mathcal{U} \psi & \Longleftrightarrow & \psi \vee (\varphi \wedge \bigcirc(\varphi \mathcal{U} \psi)) \\ \varphi \mathcal{R} \psi & \Longleftrightarrow & \psi \wedge (\varphi \vee \bigcirc(\varphi \mathcal{R} \psi)) \end{array}$$

- if $\bigcirc \psi \in \tau(i)$ then $\psi \in \tau(i+1)$
- if $\psi_1 \mathcal{U} \psi_2 \in \tau(i)$ then either $\psi_2 \in \tau(i)$, or $\psi_1 \in \tau(i)$ and $\psi_1 \mathcal{U} \psi_2 \in \tau(i+1)$
- if $\psi_1 \mathcal{R} \psi_2 \in \tau(i)$ then $\psi_2 \in \tau(i)$ and either $\psi_1 \in \tau(i)$ or $\psi_1 \mathcal{R} \psi_2 \in \tau(i+1)$

Interpreting labelings

A sequence π satisfies a formula φ if one can find a labeling τ satisfying:

- the labeling rules above

- $\varphi \in \tau(0)$, and

- if $\psi_1 \mathcal{U} \psi_2 \in \tau(i)$, then for some $j \geq i$, $\psi_2 \in \tau(j)$ (the eventuality condition)

Example

π :	p	p	p			$\ldots$
				q	q	
τ :	$p\mathcal{U}q$	$p\mathcal{U}q$	$p\mathcal{U}q$	$p\mathcal{U}q$		$\ldots$
	p	p	p	q		
	$\bigcirc(p\mathcal{U}q)$	$\bigcirc(p\mathcal{U}q)$	$\bigcirc(p\mathcal{U}q)$			

Building the GBA $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$

The automaton A_φ is the set of labeling rules + the eventuality condition(s) !

- $\Sigma = 2^{\mathcal{P}}$ is the alphabet
- $S \subseteq 2^{Cl(\varphi)}$, such that, for all $s \in S$:
  - $\varphi_1 \wedge \varphi_2 \in s \Rightarrow \varphi_1 \in s$ and $\varphi_2 \in s$
  - $\varphi_1 \vee \varphi_2 \in s \Rightarrow \varphi_1 \in s$ or $\varphi_2 \in s$
- $I = \{s \in S \mid \varphi \in s\}$,
- $(s, \alpha, t) \in T$ iff:
  - for all $p \in \mathcal{P}$, $p \in s \Rightarrow p \in \alpha$, and $\neg p \in s \Rightarrow p \notin \alpha$,
  - $\bigcirc\psi \in s \Rightarrow \psi \in t$,
  - $\psi_1 \mathcal{U} \psi_2 \in s \Rightarrow \psi_2 \in s$ or $[\psi_1 \in s$ and $\psi_1 \mathcal{U} \psi_2 \in t]$
  - $\psi_1 \mathcal{R} \psi_2 \in s \Rightarrow \psi_2 \in s$ and $[\psi_1 \in s$ or $\psi_1 \mathcal{R} \psi_2 \in t]$

Building the GBA $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$

- for each eventuality $\phi \mathcal{U} \psi \in Cl(\varphi)$, the transition relation ensures that this will appear until the first occurrence of ψ

- it is sufficient to ensure that, for each $\phi \mathcal{U} \psi \in Cl(\varphi)$, one goes infinitely often either through a state in which this does not appear, or through a state in which both $\phi \mathcal{U} \psi$ and ψ appear

- let $\phi_1 \mathcal{U} \psi_1, \ldots \phi_n \mathcal{U} \psi_n$ be the "until" subformulae of φ

  $\mathcal{F} = \{F_1, \ldots, F_n\}$, where:

$$F_i = \{s \in S \mid \phi_i \mathcal{U} \psi_i \in s \text{ and } \psi_i \in s \text{ or } \phi_i \mathcal{U} \psi_i \notin s\}$$

  for all $1 \leq i \leq n$.

Conclusion of the second part

- Model checking is a push-button verification technique
- The main limitation is the size of the system's model
- Practical for hardware systems: boolean variables, finite-state models
- Difficult for software systems: integers, pointers, recursive data structures
- There are several methods to fight state explosion:
  - finite-state systems: partial-order reductions, symmetry reductions
  - infinite-state systems: symbolic representations (automata,logic), abstract interpretation, compositional techniques
- Verification in industry:
  - hardware: Cadence, Synopsis, IBM, Intel, ...
  - software: AbsInt, GrammaTech, Coverity, Polyspace, Monoidics, ...