Büchi Automata and its application to Logic

Yiting Wang

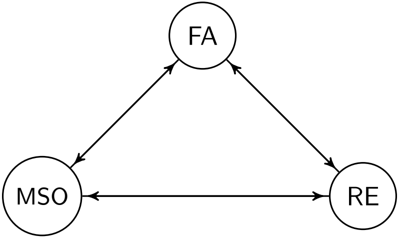
Department of Philosophy, Peking University

2019.11.12

Outline

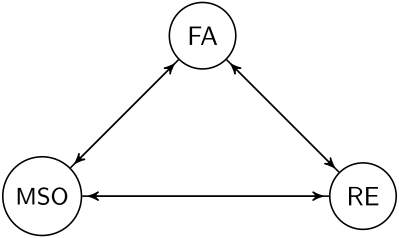
1. Introduction
2. Finite automata and MSO
3. Model Checking for LTL

Background



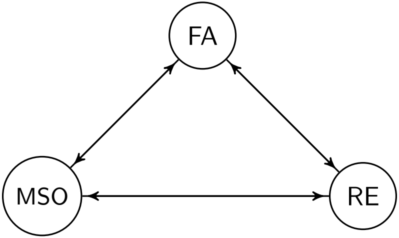
[Huffman, 1954], [Mealy, 1955] and [Moore, 1956] used deterministic finite automata for representing sequential circuits.

Background



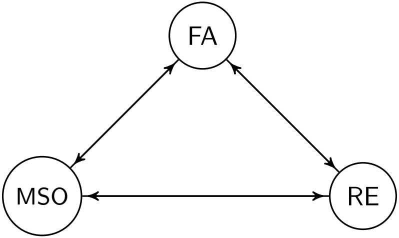
[Kleene, 1956] used regular expressions and show their equivalence to finite automata.

Background



[Büchi, 1960&1962] showed that formulae in $\mathrm{MSO}[\mathrm{S}]$ and finite state automata have the same expressive power.

Background



[Vardi,Wolper, 1986&1994] showed the connection of Büchi automata with linear temporal logic.

Preliminaries

Let Σ be a nonempty alphabet.

A finite word w of length $m \in \omega$ over Σ is a mapping from $\{0, \ldots, m-1\}$ to Σ. We often represent w as $w(0)w(1)\ldots w(m-1)$. With Σ^* we denote the set of finite words over Σ.

An infinite word w over Σ is a mapping from ω to Σ. We often represent w as $w(0)w(1)w(2)\ldots$

With Σ^ω we denote the set of infinite words over Σ.

Cont.

A path π is maximal if π is infinite, or π is a finite path of length n and $(\pi(n-1), u) \notin E$, for all $u \in V$.

A vertex $u \in V$ is reachable from $v \in V$ if there is a path $v_0 v_1 \dots v_n$ with $v = v_0$ and $u = v_n$.

If $n \geq 1$ then we say that u is nontrivially reachable from v.

$R(v)$ denotes the set of vertices that are reachable from v.

Let f and g are total functions from $\mathbb{N}$ to $\mathbb{R}^+$, then $f(n) = O(g(n))$ if there is $n_0 \in \mathbb{N}^+$ and $c \in \mathbb{R}^+$ s.t. when $n \geq n_0$, $f(n) \leq cg(n)$ always holds.

Outline

Deterministic Finite Automata

A deterministic finite automaton (DFA) $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states,
- Σ is an finite alphabet,
- $\delta : Q \times \Sigma \to Q$ is a transition function,
- $q_0 \in Q$ is a initial states, and
- $F \subseteq Q$ is a set of accept states.

A DFA $\mathcal{A}$ is called total if $|\delta(q, a)| = 1$ for all $q \in Q$ and $a \in \Sigma$.
The size of $\mathcal{A}$, denoted $|\mathcal{A}|$, is the number of states and transitions in $\mathcal{A}$, i.e. $|\mathcal{A}| = |Q| + \sum_{q \in Q} \sum_{a \in \Sigma} |\delta(q, a)|$

Non-deterministic Finite Automata

A nondeterministic finite automaton (NFA) $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ where

- Q is a finite set of states,
- Σ is an finite alphabet,
- $\delta : Q \times \Sigma \to 2^Q$ is a transition function,
- $Q_0 \subseteq Q$ is a set of initial states, and
- $F \subseteq Q$ is a set of accept states.

The transition function δ can be identified with the relation $\to \subseteq Q \times \Sigma \times Q$ given by $q \xrightarrow{a} q'$ iff $q' \in \delta(q, a)$.

$\delta^*(q, \epsilon) = \{q\}$, $\delta^*(q, a) = \delta(q, a)$, and

$$\delta^*(q, a_1 a_2 \ldots a_n) = \bigcup_{q' \in \delta(q, a_1)} \delta^*(q', a_2 \ldots a_n)$$

Run and Language

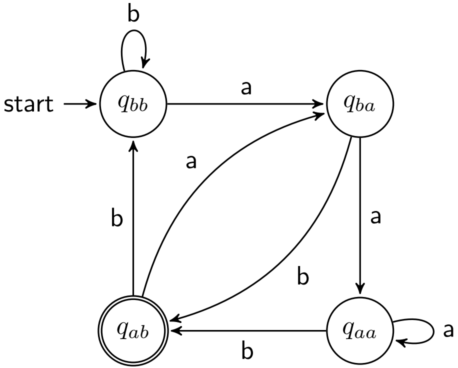
Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ be an NFA and $w = a_1 \ldots a_n \in \Sigma^*$ a finite word. A run for w in $\mathcal{A}$ is a finite sequence of states $q_0 q_1 \ldots q_n$ such that $q_0 \in Q_0$ and $q_i \stackrel{a_{i+1}}{\to} q_{i+1}$ for all $0 \leq i < n$.

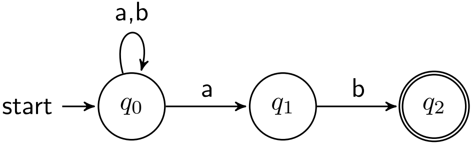
Run $q_0 q_1 \ldots q_n$ is called accepting if $q_n \in F$.

A finite word $w \in \Sigma^*$ is called accepted by $\mathcal{A}$ if there exists an accepting run for w. The accepted language of $\mathcal{A}$, denoted $\mathcal{L}(\mathcal{A})$, is the set of finite words in Σ^* accepted by $\mathcal{A}$, i.e.

$$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \text{ there exists an accepting run for } w \text{ in } \mathcal{A}\}$$

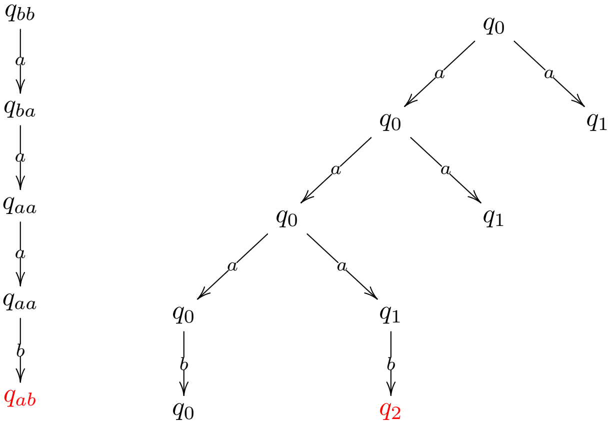
Example for DFA and NFA





$\mathcal{L}((a+b)^*ab)$, where $\Sigma = \{a, b\}$

Computation History



Determinization of NFA

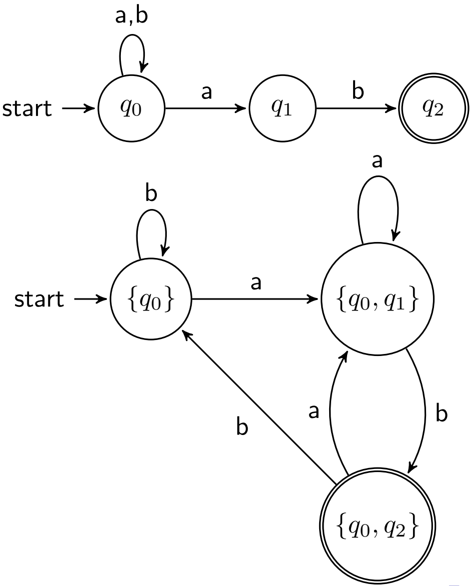
For a given NFA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ we can construct an equivlent total DFA $\mathcal{A}' = (Q^{'} \subseteq 2^Q, \Sigma, \delta', Q_0, F')$ where

$$F' = \{Q^* \subseteq Q \mid Q^* \in Q' \text{ and } Q \cap F \neq \emptyset\}$$

and the transition function $\delta : 2^Q \times \Sigma \rightarrow 2^Q$ is defined

$$\delta'(Q^*, a) = \bigcup_{q \in Q^*} \delta(q, a)$$

An example for determinization



Some properties of NFA

Languages recognized by NFA are closed under union, complement and homomorphism.

- Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, Q_1', F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, Q_2', F_2)$. Let $A = (Q, \Sigma, \delta, Q_0, F)$ where
    - $Q = Q_1 \dot{\cup} Q_2$
    - $Q_0 = Q_1' \dot{\cup} Q_2'$
    - $F = F_1 \dot{\cup} F_2$
    - $\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{if } q \in Q_1 \\ \delta_2(q,a) & \text{if } q \in Q_2 \end{cases}$
- Let h be homomorphisim between Σ and Γ. Let $w' = h(w)$. If $\mathcal{A} = (Q, \Gamma, \delta', q_0, F)$ recognize w, then we can construct $\mathcal{A}' = h(\mathcal{A}) = (Q, \Gamma, \delta', q_0, F)$, where $\delta'(q, h(a)) = \delta(q, a)$. $\mathcal{A}'$ is a NFA which recognizes w'.
- Complement part can be easily shown by constructing an equivalent total DFA.

Words as structures

Let $\mathfrak{W} = \left(W, S^{\mathfrak{W}}, (P_a^{\mathfrak{W}})_{a\in\Sigma}\right)$ be a structure with vocabulary $\sigma = \{S\} \cup \{P_a \mid a \in \Sigma\}$. $\mathfrak{W}$ is called an finite-word with alphabet Σ, if

(1) S is binary and all P_a are monadic,
(2) $W = \{0, \ldots, n-1\}, n \in \mathbb{N}$ is the set of word positions,
(3) $S^{\mathfrak{W}} = \{(n, n+1) \mid n \in W\}$ is the successor relation, and
(4) $P_a^{\mathfrak{M}} = \{i \in dom(w) \mid i$th position carry $a\}$ and $P_a^{\mathfrak{M}}$ form a partition of W.

If (1) and (3) are satisfied, $\mathfrak{W}$ is called an extended (finite) word.

MSO(Syntax)

The formulae of monadic second-order logic of vocabulary σ, denoted $MSO[\sigma]$, are defined simultaneously for all vocabularies σ by induction.

(1) If $R, S \in \sigma$ are monadic, then $R \subseteq S$ is in $\mathrm{MSO}\ [\sigma]$

(2) If $R_1, \ldots, R_k \in \sigma$ are monadic and $S \in \sigma$ has arity k, then $SR_1 \ldots R_k$ is in $\mathrm{MSO}[\sigma]$.

(3) If ϕ and ψ are in $\mathrm{MSO}\ [\sigma]$, then so are $\neg\phi, \phi \vee \psi$ and $\phi \wedge \psi$.

(4) If ϕ is in MSO $[\sigma \dot{\cup} \{R\}]$ and R is monadic, then $\exists R\ \phi$ and $\forall R\ \phi$ are in $\mathrm{MSO}[\sigma]$. Note that in this case the parameter σ changes.

MSO(Semantics)

The satisfaction relation model is defined for all vocabularies σ, all σ -structures $\mathfrak{A}$ and all $\phi \in \mathrm{MSO}[\sigma]$ along the same induction.

(1) $\mathfrak{A} \models R \subseteq S$ iff $R^{\mathfrak{A}} \subseteq S^{\mathfrak{A}}$.

(2) $\mathfrak{A} \models SR_1 \ldots R_k$ iff $S^{\mathfrak{A}} \cap (R_1^{\mathfrak{A}} \times \ldots \times R_k^{\mathfrak{A}}) \neq \emptyset$ or in other words iff there are individuals $a_1 \in R_1^{\mathfrak{A}}, \ldots, a_k \in R_k^{\mathfrak{A}}$ such that $(a_1, \ldots, a_k) \in S^{\mathfrak{A}}$.

(3) $\mathfrak{A} \models \neg\phi$ iff not $\mathfrak{A} \models \phi$

(4) $\mathfrak{A} \models \phi \vee \psi$ iff $\mathfrak{A} \models \phi$ or $\mathfrak{A} \models \psi$ holds.

(5) $\mathfrak{A} \models \exists X\ \phi$ iff there is $R \subseteq A$ s.t. $\mathfrak{A}, [X \to R] \models \phi$.
$\mathfrak{A} \models \forall X\ \phi$ iff for all $R \subseteq A$ s.t. $\mathfrak{A}, [X \to R] \models \phi$.

Some shorthands

$$
\begin{array}{lll}
X = \emptyset & \text{for} & \forall Y\ X \subseteq Y \\
\mathrm{sing}(x) & \text{for} & \neg x = \emptyset \wedge \forall X (X \subseteq x \rightarrow (x \subseteq X \vee X = \emptyset)) \\
x \in P & \text{for} & \mathrm{sing}(x) \wedge x \subseteq P \\
P = Q & \text{for} & P \subseteq Q \wedge Q \subseteq P
\end{array}
$$

$Incl(P)$ means $y \in P^{\mathfrak{W}}$ implies $x \in P^{\mathfrak{W}}$ for all word positions $x \leq y$

$$Incl(P) = \forall x \forall y ((sing(x) \wedge Sxy \wedge y \in P \rightarrow x \in P))$$

$$x \leq y := sing(y) \wedge \forall P (Incl(P) \wedge y \in P \rightarrow x \in P)$$

Büchi Theorem (over finite word)

Büchi Theorem [Büchi, 1960]

A language of finite words is recognizable by a finite state automaton iff it is MSO-definable and both conversions from automata to formulae and vice versa are effective.

From NFA to MSO formulae

- $\phi_{\mathcal{A}} := \exists \bar{R}(Part \wedge Init \wedge Trans \wedge Accept)$
  $State_q(x) := x \in R_q$
- $Part := \forall x(sing(x) \rightarrow \bigvee_{q \in Q} State_q(x))$
- $Init := \exists x(State_{q_I}(x) \wedge \forall y(sing(y) \rightarrow x \leq y))$
- $Trans := \forall x \forall y(sing(x) \wedge sing(y) \wedge S(x,y) \rightarrow \bigvee_{(q,a,q') \in \delta} (State_q(x) \wedge$
  $x \in P_a \wedge State_{q'}(y)))$
- $Accept : \forall x(\forall y(y \leq x \wedge \bigvee_{q \in F} State_q(x)))$

From MSO formulae to NFA

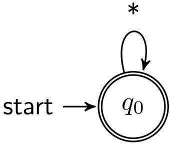
We proceed using induction on ϕ. In order to apply induction, the statement has to be modified such that not only infinite words, but also extended words are permitted. We have to express how extended words may be represented by words. An MSO-formula $\phi(X_1, \ldots, X_n)$ with at most the free variables $X_1, \ldots, X_n$ is interpreted in a word with n designated subsets $P_1, \ldots, P_n$. Such a model represents a word over expanded alphabet $A' = A \times \{0,1\}^n$, where the label $(a, c_1, \ldots, c_n)$ of position p indicates that p carries label a from A and that p belong to P_j iff $c_j = 1$. For instance, the ω-word model $w^* = (w, P_1, P_2)$ where $w = abbaaaa\ldots, P_1$ is the set of even numbers and P_2 is the set of prime numbers, will be identified with the follow ω-word over $\{a,b\} \times \{0,1\}^2$ where

w a b b a a a a

P_1 1 0 1 0 1 0 1 ...

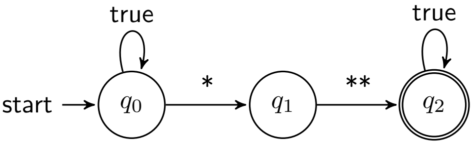
P_2 0 0 1 1 0 1 0

Base Case: If $\phi := X_1 \subseteq X_2$, we have $\mathfrak{W}' \vDash \phi$ iff at every position x the following condition holds: whenever 1 occurs in the first additional 0-1 component it also occurs in the second additional component. Therefore the automaton $\mathfrak{A}_\phi$ verifies that the labels $(\Sigma, 1, 0)$ does not occur in W'. We set $\mathcal{A}_\phi = (\{q\}, q, \Sigma \times \{0,1\}^2, \delta, \{q\})$



$* := (\Sigma, 0, 0), (\Sigma, 0, 1), (\Sigma, 1, 1).$

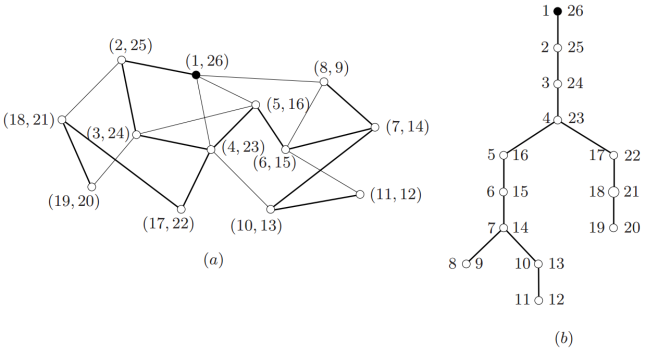
If $\phi := SX_1X_2$, then $\mathfrak{W}' \vDash \phi$ iff there are positions x and y, $x \in X_1$, $y \in X_2$ and $(x, y) \in S^{\mathfrak{W}'}$ and at position x, 1 occurs in the first additional 0-1 component, at position y, 1 occurs in the second additional 0-1 component. We can construct $\mathcal{A}_\phi$ as follows:



$* := (\Sigma, 1, 0), (\Sigma, 1, 1)$.
$** := (\Sigma, 0, 1), (\Sigma, 1, 1)$.

Induction Step is followed by the properties of NFA we have shown.

Check the emptiness of NFA



ω-word

Let $\mathfrak{W} = \left(W, S^{\mathfrak{W}}, (P_a^{\mathfrak{W}})_{a \in \Sigma}\right)$ be a structure with vocabulary $\sigma = \{S\} \cup \{P_a \mid a \in \Sigma\}$. $\mathfrak{W}$ is called an ω-word with alphabet Σ, if

(1) S is binary and all P_a are monadic,

(2) $W = \omega$ is the set of word positions,

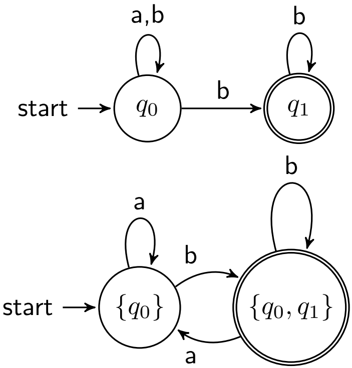
(3) $S^{\mathfrak{W}} = \{(n, n+1) \mid n \in \omega\}$ is the successor relation,

(4) $P_a^{\mathfrak{M}}$ form a partition of W.

Büchi automata

An ω-automaton $\mathcal{B} = (Q, \Sigma, \delta, q_I, F)$ with acceptance component $F \subseteq Q$ is called Büchi automaton if it is used with the following acceptance condition (Büchi acceptance): A word $w \in \Sigma^\omega$ is accepted by $\mathcal{B}$ iff there exists a run r of B on w satisfying the condition: $Inf(r) \cap F \neq \emptyset$ i.e. at least one of the states in F has to be visited infinitely often during the run. $\mathcal{L}(\mathcal{B}) := \{w \in \Sigma^\omega \mid \ \mathcal{B} \text{ accepts } w\}$ is the ω-language recognized by $\mathcal{B}$.
A run r of $\mathcal{B}$ on $w \in \Sigma^\omega$ is an infinite word over Q with $r(0) = q_I$ and $r(i+1) \in \delta(r(i), w(i))$, for all $i \in \omega$. r is accepting if a final state occurs infinitely often in r, i.e. $F \cap Inf(r) = \emptyset$.
$\mathcal{B}$ accepts w if there is an accepting run of $\mathcal{B}$ on w. Otherwise, w is rejected.

Subset construction fails for NBA



NBA is more expressive than DBA

Assume that $\mathcal{L}((a+b)^*b^\omega) = \mathcal{L}(\mathcal{B})$ for some DBA $\mathcal{B} = (Q, \Sigma, \delta, q_0, F)$ with $\Sigma = \{a, b\}$. Since the word $w_1 = b^\omega \in \mathcal{L}((a+b)^*b^\omega)$, there exists an accepting state $q_1 \in F$ and a $n_1 \in \mathbb{N}^+$ such that $\delta^*(q_0, b^{n_1}) = q_1 \in F$. Now consider the word $w_2 = b^{n_1}ab^\omega \in \mathcal{L}((a+b)^*b^\omega)$. Since w_2 is accepted by a, there exists an accepting state $q_2 \in F$ and $n_2 \in \mathbb{N}^+$, such that

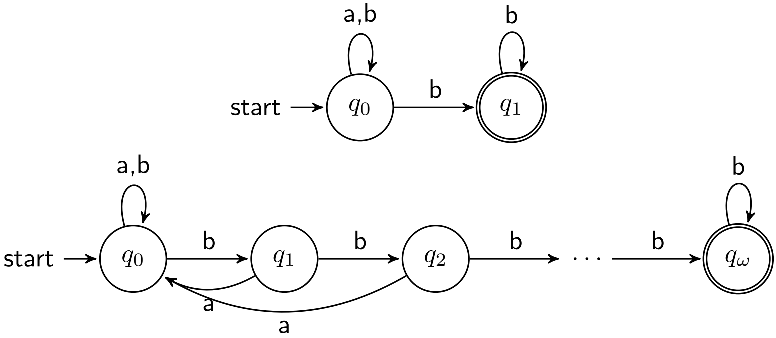
$$\delta^*(q_0, b^{n_1}ab^{n_2}) = q_2 \in F$$

Note that $q_1 \neq q_2$, Continuing this process, we obtain a sequence $n_1, n_2, n_3, \ldots \in \mathbb{N}^+$ and a sequence $q_1, q_2, q_3, \ldots$ of accepting states such that

$$\delta^*(q_0, b^{n_1}ab^{n_2}a \ldots b^{n_{i1}}ab^{n_i}) = q_i \in F, i \geq 1 \ldots$$

However, there are only finitely many states in $\mathcal{B}$, Contradiction.

NBA is more expressive than DBA



General Büchi automaton

A GNBA is a tuple $\mathcal{G} = (Q, \Sigma, \delta, Q_0, \mathcal{F})$ where Q, Σ, δ, Q_0 are defined as for NBA and $\mathcal{F}$ is a (possibly empty) subset of 2^Q. The elements of $\mathcal{F}$ are called accpectance sets. The accepted language $\mathcal{L}_\omega(\mathcal{G})$ consists of all infinite words in $(2^{AP})^\omega$ that have at least one infinite run $q_0 q_1 q_2 \ldots$ in $\mathcal{G}$ such that for each acceptance set $F \in \mathcal{F}$ there are infinitely many indices i with $q_i \in F$.

$Generalized\ Büchi : \mathcal{F} = \{F_1, \ldots, F_k\}, F_i \subseteq Q$, if for all $1 \leq i \leq k$, $inf(r) \cap F_i \neq \emptyset$, then r is accepted.

Büchi Theorem (over infinite words)

Büchi Theorem [Büchi,1962]

An ω language is Büchi recognizable iff it is MSO-definable and the transformation of Büchi automata into MSO formulae and conversely is effective.

From Büchi automata to MSO formulae

- $\phi_A := \exists \bar{R}(Part \wedge Init \wedge Trans \wedge Accept)$
  $State_q(x) := x \in R_q$
- $Part := \forall x(sing(x) \rightarrow \bigvee_{q \in Q} State_q(x))$
- $Init := \exists x(State_{q_I}(x) \wedge \forall y(sing(y) \rightarrow x \leq y))$
- $Trans := \forall x \forall y(sing(x) \wedge sing(y) \wedge S(x,y) \rightarrow \bigvee_{(q,a,q') \in \delta} (State_q(x) \wedge x \in P_a \wedge State_{q'}(y)))$
  $Inf(P) := \forall x(x \in P \rightarrow \exists y(y \in P \wedge x < y))$
  $InfOcc_q(P) := \exists Q(Q \subseteq P \wedge Q \subseteq R_q \wedge Inf(Q))$
  $Büchi(P) := \bigvee_{q \in F} InfOcc_q(P)$
- $Accept := \exists X(Büchi(X))$

From MSO formulae to Büchi automata

This direction is similar to the discussion in proof of Büchi Theorem over finite words. Use similar methods we can easily prove that Büchi automata are closed under Union and Homomorphism. It remains to discuss whether Büchi automata are closed under complement.

Complement [Klarlund, 1997]

Let $\mathcal{B}$ be the NBA $(Q, \Sigma, \delta, q_I, F)$, and let $w \in \Sigma^\omega$. The run graph G of $\mathcal{B}$ for w is a graph (V, E, C), with

(i) the set of vertices $V := \bigcup_{i\in\omega} S_i \times \{i\}$, where the S_i s are inductively defined by $S := \{q_{\mathrm{I}}\}$ and $S_{i+1} := \bigcup_{q\in S_i} \delta(q, w(i))$ for $i \geq 0$

(ii) the set of edges $E := \{((p, i), (q, i+1)) \in V \times V \mid q \in \delta(p, w(i))\}$, and

(iii) the set of marked vertices $C := \{(q, i) \in V \mid q \in F\}$

Corollary

Let $G = (V, E, C)$ be a run graph of $\mathcal{B}$, and let $w \in \Sigma^*$. Then, $w \notin \mathcal{L}(\mathcal{B})$ iff $Inf(\pi) \cap C = \emptyset$, for all paths π in G.

Some definitions

- Let Q be a finite set. A sliced graph over Q is a graph $G = (V, E, C)$, where $V \subseteq Q \times \omega, C \subseteq V$, and for $(p, i), (q, j) \in V$, if $((p, i), (q, j)) \in E$ then $j = i + 1$.
  Note that a run graph is a sliced graph.
- The sliced graph $G = (V, E, C)$ is finitely marked if for all paths π in G, $Inf(\pi) \cap C = \emptyset$.
- The ith slice S_i is the set $\{q \in Q \mid (q, i) \in V\}$.
- The width of G, $||G||$ for short, is the limes superior of the sequence $(|S_i|)_{i \in \omega}$. In other words, the width of a sliced graph is the largest cardinality of the slices $S_0, S_1, \ldots$.

cont.

The **unmarked boundary** $U(G)$ is the set of vertices that do not have a nontrivially reachable vertex that is marked, i.e.

$$U(G) := \{v \in V \mid C \cap (R(v) \setminus \{v\}) = \emptyset\}$$

The **finite boundary** $B(G)$ is the set of vertices that have only finitely many reachable vertices, i.e.

$$B(G) := \{v \in V \mid R(v) \text{ is a finite set}\}$$

Lemma 1

Lemma 1

Let $G = (V, E, C)$ be a sliced graph that is finitely marked. If $V \neq \emptyset$ then $U(G) \neq \emptyset$

Proof.

Assume that $U(G) = \emptyset$. Let v_0 be some vertex in V. Note that $R(v_0) \setminus \{v_0\} \neq \emptyset$ because of the assumption $U(G) = \emptyset$. There is a finite path from v_0 to a vertex v_1 with $v_0 \neq v_1$ and $v_1 \in C$, since $v_0 \notin U(G)$. The vertex v_1 is not in $U(G)$, since it is assumed that $U(G)$ is empty. Repeating this argument we get an infinite sequence $v_0, v_1, v_2, \ldots$ of distinct vertices, where v_{i+1} is reachable from v_i, for $i \geq 0$. Furthermore, $v_i \in C$, for $i > 0$. This contradicts the assumption that G is finitely marked.

Lemma 2

Lemma 2

Let $G = (V, E, C)$ be a sliced graph. For every vertex $v \in V \setminus B(G)$, there exists an infinite path in $G \setminus B(G)$ starting with v.

Proof.

If $R(v) \setminus B(G)$ is infinite then, by König's Lemma, there exists an infinite path in $G \setminus B(G)$ starting with v, since $R(v) \setminus B(G)$ is infinite and $G \setminus B(G)$ is finitely branching. It remains to show that $R(v) \setminus B(G)$ is infinite. So, for a contradiction assume that $R(v) \setminus B(G)$ is finite. Let $B := \{u \in B(G) \mid$ there exists a $u' \in R(v) \setminus B(G)$ with $(u', u) \in E\}$. The set B is finite since $R(v) \setminus B(G)$ is finite and G is finitely branching. Since $B \subseteq B(G)$, we have that $R(u)$ is finite, for all $u \in B$. We have the following equality: $R(v) = (R(v) \setminus B(G)) \cup \bigcup_{u \in B} R(u)$.

In particular, $R(v)$ is a finite union of finite sets. This is not possible since $R(v)$ is infinite, for all $v \in V \setminus B(G)$.

Slices

Let $G = (V, E, C)$ be a sliced graph. We define a sequence of sliced graphs $G_0, G_1, \dots$ and a sequence of sets of vertices $V_0, V_1, \dots$ as follows: $G_0 := G, V_0 := B(G)$, and for $i \geq 0$:

$$G_{2i+1} := G_{2i} \setminus V_{2i} \qquad V_{2i+1} := U(G_{2i+1})$$

$$G_{2i+2} := G_{2i+1} \setminus V_{2i+1} \qquad V_{2i+2} := B(G_{2i+1})$$

Lemma 3

Lemma 3

Let $G = (V, E, C)$ be a sliced graph that is finitely marked with $||G_{2i+1}|| > 0$, for some $i \geq 0$. Then $||G_{2i+2}|| < ||G_{2i+1}||$.

Proof.

Since $||G_{2i+1}|| > 0$ the set of vertices of G_{2i+1} is not empty. From Lemma 1 it follows that there is a vertex $v_0 \in U(G_{2i+1})$. From the definition of $G_{2i+1} = G_{2i} \setminus V_{2i}$ it follows that $v_0 \in V \setminus B(G)$ if $i = 0$, and $v_0 \in V' \setminus B(G_{2i-1})$ if $i > 0$, where V' is the set of vertices of G_{2i}. From Lemma 2 we can conclude that there exists an infinite path $v_0 v_1 v_2 \ldots$ in G_{2i+1}. Obviously, $v_j \in U(G_{2i+1})$, for all $j \geq 0$. Let $v_j = (q_j, k_j)$. It holds $||G_{2i+2}|| < ||G_{2i+1}||$ since each slice of G_{2i+2} with index k_j does not contain q_j.

Lemma 4

Lemma 4

Let $G = (V, E, C)$ be a sliced graph that is finitely marked and let $n = ||G||$. Then G_{2n+1} is the empty graph.

Proof.

Note that $n \leq |Q|$ assuming $V \subseteq Q \times \omega$ for some finite set Q. Assume that G_{2n+1} is not the empty graph. It holds $||G_{2n+1}|| > 0$, since $G_{2n+1} = G_{2n} \setminus B(G_{2n-1})$. From the lemma above it follows that $n > ||G_1|| > ||G_3|| > \ldots > ||G_{2n+1}||$. This contradicts $||G_{2n+1}|| > 0$.

Progress measure

A **progress measure** of size $m \in \omega$ for a sliced graph $G = (V, E, C)$ is a function $\mu : V \to \{1, \dots, 2m+1\}$ satisfying the following three conditions:

(i) $\mu(u) \geq \mu(v)$, for all $(u, v) \in E$,

(ii) if $\mu(u) = \mu(v)$ and $(u, v) \in E$ then $\mu(u)$ is odd or $v \notin C$, and

(iii) there is no infinite path $v_0 v_1 v_2 \dots \in V^\omega$ where $\mu(v_0)$ is odd and $\mu(v_0) = \mu(v_1) = \mu(v_2) = \dots$.

Progress measure and Automata

Theorem

Let $\mathcal{B} = (Q, \Sigma, \delta, q_I, F)$ be a NBA and let $w \in \Sigma^\omega$. Then, $\mathcal{B}$ rejects w iff there exists a progress measure of size $|Q|$ for the run graph $G = (V, E, C)$ of $\mathcal{B}$ for w.

Proof.

($\Rightarrow$:) Note that the run graph G is finitely marked by Corollary. Let $\mu : V \to \{1, \ldots, 2|Q|+1\}$ be the function defined by $\mu(v) := i + 1$, where i is the uniquely determined index with $v \in V_i$ and $v \notin V_{i+1}$. From Lemma 4 it follows that $1 \leq i \leq 2|Q|$ and thus μ is well-defined. It remains to show that μ is a progress measure.
First, we show that there is no infinite path $v_0 v_1 \ldots$ with $\mu(v_0) = \mu(v_1) = \ldots$ where $\mu(v_0)$ is odd. Assume that $\mu(v_0) = 2i + 1$ for $v_0 \in V$. Then $v_0 \in V_{2i}$. By definition of V_{2i}, the vertices in V_{2i} have only finitely many reachable states in G if $i = 0$ and G_{2i-1} if $i > 0$. Thus, every path $v_0 v_1 \ldots$ with $2i + 1 = \mu(v_0) = \mu(v_1) = \ldots$ must be finite.

Cont.

Proof.

Second, for $(u, v) \in E$, it holds $\mu(u) \geq \mu(v)$. This follows from the fact that (i) $u \in U(G')$ implies $v \in U(G')$, and (ii) $u \in B(G')$ implies $v \in B(G')$, for every sliced graph $G' = (V', E', C')$ with $(u, v) \in V$.
Third, we show by contradiction that if $\mu(u) = \mu(v)$ then $\mu(u)$ is odd or $v' \in C$, for $(u, v) \in E$. Assume that $\mu(u)$ is even and $v \in C$. Since $\mu(u)$ is even, we have that $u \in V_{2i+1} = U(G_{2i+1})$, for some $0 \leq i \leq |Q|$. Since $v \in C$, it holds $u \notin U(G_{2i+1})$. Contradiction!
$(\Leftarrow:)$ Let $\mu : V \to \{1, \ldots, 2|Q| + 1\}$ be a progress measure for G. Let π be an infinite path in G. Since μ is monotonicly decreasing, there exists a $k \geq 0$ with $\mu(\pi(k)) = \mu(\pi(k+1)) = \ldots$. By the definition of a progress measure, $\mu(\pi(k))$ must be even and $\mu(\pi(k)), \mu(\pi(k+1)), \ldots \notin C$. Thus, the corresponding run of π is not accepting. Since π was chosen arbitrarily there is no accepting run of $\mathcal{B}$ on w by Corollary.

Construction

Let $\mathcal{B} = (Q, \Sigma, \delta, q_I, F)$ be a NBA. We can construct a NBA $\mathcal{B}'$ with $2^{O(|Q|+|Q|log|Q|)}$ states such that $\mathcal{B}$ accepts $w \in \Sigma^\omega$ iff there exists a progress measure of size $|Q|$ for the run graph G of $\mathcal{B}$ for w. Let Ψ be the set of partial functions from Q to $\{1, \ldots, 2m+1\}$. Note that the cardinality of Ψ is $|Q|^{O(|Q|)} = 2^{O(|Q|log|Q|)}$. Moreover, let $f_I \in \Psi$ be the partial function, where $f_I(q_I) := 2|Q|+1$ and $f_I(q)$ is undefined for $q \neq q_I$. Let $\mathcal{B}'$ be the NBA $(\Psi \times P(Q), \Sigma, \delta', (f_I, \emptyset), \Psi \times \emptyset)$ with $(f', P') \in \delta'((f, P), a)$ iff the following conditions are satisfied:

(1) $q' \in dom(f)$ iff there exists $q \in dom(f)$ such that $q' \in \delta(q, a)$.

(2) $f'(q') \leq f(q)$, for $q' \in \delta(q, a)$. Moreover, if $q' \in F$ and $f(q)$ is even then $f'(q') < f(q)$.

(3) If $P = \emptyset$ then $q \in P'$ iff $f'(q)$ is odd, for $q \in \mathrm{dom}(f')$

(4) If $P \neq \emptyset$ then $q' \in P'$ iff there exists $q \in P$ such that $q' \in \delta(q, a)$ and $f(q) = f^{'}(q')$ is odd.

Cont.

The number of the states of $\mathcal{B}'$ is

$$|\Psi \times P(Q)| = 2^{O(|Q|log|Q|)} \cdot 2^{|Q|} = 2^{O(|Q|+|Q|log|Q|)}.$$

($\Rightarrow$) Let r be an accepting run of $\mathcal{B}'$ on w with $r(k) = (f_k, P_k)$, for $k \in \omega$ and let $G = (V, E, C)$ be the run graph of $\mathcal{B}$ for w. Let $\mu : V \to \{1, \ldots, 2|Q|+1\}$ with $\mu(q,k) := f(q)$, for $r(k) = (f, P)$.
It remains to show that μ is a progress measure for G. Because of condition (1) it holds for all $k \in \omega$ that $((q,k),(q',k+1)) \in E$ iff $q \in dom(f_k)$, $q' \in dom(f_{k+1})$, and $q' \in \delta(q, w(k))$. This can be easily shown by induction over k. Let $(v, v') \in E$. Because of condition (2), $\mu(v) \leq \mu(v')$, and if $v' \in C$ then $\mu(v) < \mu(v')$. Note that $P_k = \emptyset$, for infinitely many $k \in \omega$, since r is accepting. Hence, the conditions (3) and (4) ensure that there is no infinite path $v_0 v_1 \ldots$. in G, where $\mu(v_0) = \mu(v_1) = \ldots$ and $\mu(v_0)$ is odd.

$\Rightarrow$

Let $\mu : V \to \{1, \ldots, 2|Q| + 1\}$ be a progress measure for the run graph $G = (V, E, C)$ of $\mathcal{B}$ for w. Note that $w \notin \mathcal{L}(\mathcal{B})$ by Theorem. Let $f_k : Q \to \{1, \ldots, 2|Q| + 1\}$ be the partial function where $f_k(q) := \mu(q, k)$, for $q \in S_k$, and otherwise f_k is undefined. Let r be the infinite word, with $r(0) := (f_I, \emptyset)$and for $k \geq 0, r(k+1) := (f_{k+1}, P_{k+1})$ with

$$P_{k+1} := \{q \in Q \mid f_{k+1}(q) \text{ is odd}\},$$

for $P_k = \emptyset$, and

$$P_{k+1} := \{q \in Q \mid f_k(p) = f_{k+1}(q) \text{ is odd and } ((p, k), (q, k+1)) \in E\}$$

otherwise.

$\Leftarrow$

By induction over k it is straightforward to show that r is a run of $\mathcal{B}'$ on w. It remains to show that r is accepting, i.e., there are infinitely many $k \in \omega$ such that $P_k = \emptyset$. Assume that there is an $n \in \omega$ such that $P_n = \emptyset$ and $P_{n+1}, P_{n+2}, \ldots \neq \emptyset$. Note that if $q \in P_k$ with $k > n$ then there exists a $p \in P_{n+1}$ such that the vertex (q, k) is reachable from a vertex $(p, n+1)$ in G. Thus, there is an infinite path $v_0 v_1 \ldots$ with $v_i = (q_i, k_i)$ for $i \geq 0$, and there is an infinite sequence of indices $i_0 < i_1 < \ldots$ such that $q_{i_j} \in P_{k_{i_j}}$ for all $j \geq 0$. Since μ is a progress measure, it is $\mu(v_{i_{j'}}) \leq \mu(v_{i_j})$ for $j' \geq j$. Thus, there exists a $k > n$ such that $\mu(v_k)$ is odd and $\mu(v_k) = \mu(v_{k+1}) = \ldots$ This contradicts the assumption that μ is a progress measure.

Check the emptiness of NBA

Lemma

Let $\mathcal{B} = (Q, \Sigma, \delta, Q_0, F)$ be an NBA. Then, the following two statements are equivalent:
(a) $\mathcal{L}(\mathcal{B}) \neq \emptyset$,
(b) There exists a reachable accept state q that belongs to a cycle in $\mathcal{B}$.

$$\exists q_0 \in Q_0\ \exists q \in F\ \exists w \in \Sigma^*\ \exists v \in \Sigma^+\ \ q \in \delta^*(q_0, w) \cap \delta^*(q, v)$$

By the above lemma, the emptiness problem for NBA can be solved by means of graph algorithms that explore all reachable states and check whether they belong to a cycle.
Since the strongly connected components of a (finite) directed graph can be computed in time linear in the number of states and edges, the time complexity of this algorithm for the emptiness check of NBA $\mathcal{B}$ is linear in the size of $\mathcal{B}$.

Decidability of MSO

MSO is decidable

By Büchi Theorem we may effectively construct an automaton $\mathcal{A}$ such that $\mathcal{A}$ accepts $\mathfrak{A}$ iff $\mathfrak{A} \models \neg\phi$. The question whether or not $\mathfrak{A} \models \phi$ always holds can be reduced to the question whether or not the language of $\mathcal{A}$ is empty. And emptiness of all these languages is decidable.

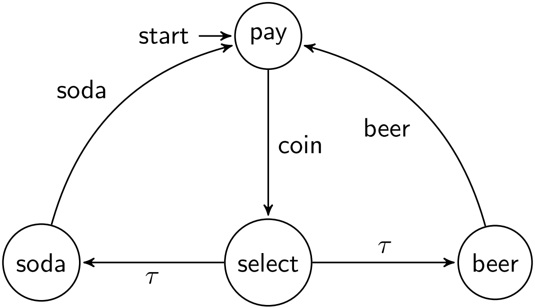
Outline

Transition System

A transition system TS is a tuple $(S, Act, \rightarrow, I, AP, L)$ where

- S is a set of states
- Act is a set of actions,
- $\rightarrow \subseteq S \times Act \times S$ is a transition relation,
- $I \subseteq S$ is a set of initial states,
- AP is a set of atomic propositions,
- $L : S \rightarrow 2^{AP}$ is a labeling function.
- TS is called finite if S, Act, and AP are finite.

An Example for TS



Path and Trace in TS

A finite path fragment π of TS is a finite state sequence $s_0 s_1 \ldots s_n$ such that $s_i \in Post(s_{i-1})$ for all $0 < i \leq n$, where $n \geq 0$.
An infinite path fragment π is an infinite state sequence $s_0 s_1 s_2 \ldots$ such that $s_i \in Post(s_{i-1})$for all $i > 0$. For $j \geq 0$, let $\pi[j] = s_j$ denote the jth state of π. The jth suffix of π, notation $\pi[j \ldots]$, is defined as $\pi[j \ldots] = s_j s_{j+1} \ldots$
A maximal path fragment is either a finite path fragment that ends in a terminal state, or an infinite path fragment.
A path fragment is called initial if it starts in an initial state.

A path of transition system TS is an initial, maximal path fragment.

Traces in TS

Traces are sequences of the form $\mathcal{L}(s_0)\mathcal{L}(s_1)\mathcal{L}(s_2)\ldots$ that register the (set of) atomic propositions that are valid along the execution. The traces of a transition system are thus words over the alphabet 2^{AP}, the sequence of sets of atomic propositions that are valid in the states of the path. A trace of state s is the trace of an infinite path fragment π with $first(\pi) = s$. Accordingly, a finite trace of s is the trace of a finite path fragment that starts in s. Let $Traces(s)$ denote the set of traces of s, and $Traces(TS)$ the set of traces of the initial states of transition system TS:

$$Traces(s) = trace(Paths(s)),\ Traces(TS) = \bigcup_{s \in I} Traces(s)$$

Semantics of LTL over Paths and States

Let $TS = (S, Act, \to, I, AP, L)$ be a transition system without terminal states, and let ϕ be an LTL-formula over AP.
For infinite path fragment π of TS, the satisfaction relation is defined by

$$\pi \vDash \phi \text{ iff } trace(\pi) \vDash \phi.$$

For state $s \in S$, the satisfaction relation $\vDash$ is defined by

$$s \vDash \phi \text{ iff } (\forall \pi \in Paths(s)\ \pi \vDash \phi).$$

TS satisfies ϕ, denoted $TS \vDash \phi$, if $Traces(TS) \subseteq Words(\phi)$.

LTL(Syntax)

LTL formulae over the set AP of atomic proposition are formed according to the following grammar:

$$\phi ::= true \mid a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc\phi \mid \phi_1 U \phi_2$$

where $a \in AP$.

LTL(Semantics)

$\sigma \models true$		
$\sigma \models a$	iff	$a \in A$
$\sigma \models \phi_1 \vee \phi_2$	iff	$\sigma \models \phi_1$ and $\sigma \models \phi_2$
$\sigma \models \neg\phi$	iff	$\sigma \not\models \phi$
$\sigma \models \bigcirc\phi$	iff	$\sigma[1\ldots] \models A_1A_2A_3 \models \phi$
$\sigma \models \phi_1 U \phi_2$	iff	$\exists j \geq 0\ \sigma[j\ldots] \models \phi_2$ and $\sigma[i\ldots] \models \phi_1,$ for all $0 \leq i < j$
$\sigma \models \Diamond\phi$	iff	$\exists j \geq 0\ \sigma[j\ldots] \models \phi$
$\sigma \models \Box\phi$	iff	$\forall j \geq 0\ \sigma[j\ldots] \models \phi$
$\sigma \models \Box\Diamond\phi$	iff	$\exists^\infty \sigma[j\ldots] \models \phi$
$\sigma \models \Box\Diamond\phi$	iff	$\forall^\infty \sigma[j\ldots] \models \phi$

Equivalence of LTL Formulae

LTL formulae ϕ_1, ϕ_2 are equivalent, denoted $\phi_1 \equiv \phi_2$, if $Words(\phi_1) = Words(\phi_2)$

$$\phi U \psi \equiv \psi \vee (\phi \wedge \bigcirc(\phi U \psi))$$

$$\neg(\phi U \psi) \equiv ((\phi \wedge \neg\psi) U (\neg\phi \wedge \neg\psi)) \vee \Box(\phi \wedge \neg\psi)$$

$$\phi_1 R \phi_2 \stackrel{\text{def}}{=} \neg(\neg\phi_1 U \neg\phi_2)$$

Positive Normal Form

For $a \in AP$, LTL formulae in positive normal form (PNF) are given by

$$\phi := true \mid false \mid a \mid \neg a \mid \phi_1 \wedge \phi_2 \mid \bigcirc\phi \mid \phi_1 U \phi_2 \mid \phi_1 R \phi_2$$

For any LTL formula ϕ there exists an equivalent LTL formula ϕ in PNF with $|\phi| = O(|\phi|)$

The algorithm of Model Checking

This approach is based on the fact that each LTL formula ϕ can be represented by a NBA. The basic idea is to try to disprove $TS \vDash \phi$ by "looking" for a path π in TS with $\pi \nvDash \phi$. If such a path is found, a prefix of π is returned as error trace. If no such path is encountered, it is concluded that $TS \vDash \phi$

$$\begin{array}{lll} TS \vDash \phi & \text{iff} & \text{Traces }(TS) \subseteq \text{ Words }(\phi) \\ & \text{iff} & \text{Traces }(TS) \cap \left(\left(2^{AP}\right)^{\omega} \setminus \text{ Words }(\phi)\right) = \varnothing \\ & \text{iff} & \text{Traces }(TS) \cap \text{ Words }(\neg\phi) = \varnothing \end{array}$$

Closure of ϕ

Closure of ϕ

The closure of LTL formula ϕ is the set closure(ϕ) consisting of all subformulae ψ of ϕ and their negation $\neg\psi$ (where ψ and $\neg\neg\psi$ are identified).

Elementary Sets of Formulae

$B \subseteq closure(\phi)$ is elementary if it is consistent with respect to propositional logic, maximal, and locally consistent with respect to the until operator.

Elementary Sets of Formulae

The requirements for local consistency result from the expansion law

$$\phi_1 U \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \bigcirc(\phi_1 U \phi_2))$$

Due to the required maximality and propositional logic consistency, we have

$$\psi \in B \text{ if and only if } \neg\psi \notin B$$

for all elementary sets B and subformulae ψ of ϕ. Further, due to maximality and local consistency, we have

$$\phi_1, \phi_2 \notin B \text{ implies } \phi_1 U \phi_2 \notin B$$

Hence, if $\phi_1, \phi_2 \notin B$ then $\{\neg\phi_1, \neg\phi_2, \neg(\phi_1 U \phi_2)\} \subseteq B$ (assuming that $\phi_1 U \phi_2$ is a subformula of ϕ).

GNBA for LTL Formulae [Vardi,Wolper, 1986

For any LTL formula ϕ (over AP) there exists a GNBA $\mathcal{G}_\phi$ over the alphabet 2^{AP} such that

(1) $Words(\phi) = \mathcal{L}(\mathcal{G}_\phi)$.

(2) $\mathcal{G}_\phi$ can be constructed in time and space $2^{O(|\phi|)}$.

Let ϕ be an LTL formula over AP. Let $\mathcal{G}_\phi = (Q, 2^{AP}, \delta, Q_0, \mathcal{F})$ where

- Q is the set of all elementary sets of formulae $B \subseteq$ closure (ϕ)
- $Q_0 = \{B \in Q \mid \phi \in B\}$
- $\mathcal{F} = \{F_{\phi_1 U \phi_2} \mid \phi_1 U \phi_2 \in \text{closure}(\phi)\}$ where

$$F_{\phi_1 U \phi_2} = \{B \in Q \mid \phi_1 U \phi_2 \notin B \text{ or } \phi_2 \in B\}$$

The transition relation $\delta : Q \times 2^{AP} \to 2^Q$ is given by:

- If $A \neq B \cap AP$, then $\delta(B, A) = \varnothing$
- If $A = B \cap AP$, then $\delta(B, A)$ is the set of all elementary sets of formulae B' satisfying
  (i) for every $\bigcirc \psi \in$ closure (ϕ) : $\bigcirc\psi \in B \Leftrightarrow \psi \in B'$, and
  (ii) for every $\phi_1 U \phi_2 \in$ closure (ϕ) :
  $\phi_1 U \phi_2 \in B \Leftrightarrow (\phi_2 \in B \vee (\phi_1 \in B \wedge \phi_1 U \phi_2 \in B'))$

$\mathcal{L}(\mathcal{G}_\phi) = Words(\phi)$

$\supseteq$: Let $\sigma = A_0 A_1 A_2 \ldots \in$ Words (ϕ). Then, $\sigma \in \left(2^{AP}\right)^\omega$ and $\sigma \models \phi$. The elementary set B_i of formulae is defined as follows:

$$B_i = \{\psi \in \text{closure}(\phi) \,|A_i A_{i+1} \ldots \vDash \psi\}$$

Obviously, B_i is an elementary set of formulae, i.e., $B_i \in Q$. We now prove that $B_0 B_1 B_2 \ldots$ is an accepting run for σ. Observe that $B_{i+1} \in \delta(B_i, A_i)$ for all $i \geqslant 0$, since for all i :

- $A_i = B_i \cap AP$
- for $\bigcirc \psi \in$ closure(ϕ)

$$\bigcirc\psi \in B_i$$

iff

$$A_i A_{i+1} \ldots \models \bigcirc\psi \text{ iff}$$
$$A_{i+1} A_{i+2} \ldots \models \psi \text{ iff}$$
$$\psi \in B_{i+1}$$

$\supseteq$

- for $\phi_1 \cup \phi_2 \in$ closure (ϕ):

$$\phi_1 \cup \phi_2 \in B_i$$

iff

$$A_i A_{i+1} \ldots \models \phi_1 U \phi_2$$

iff

$$A_i A_{i+1} \ldots \models \phi_2 \text{ or}$$
$$A_i A_{i+1} \ldots \models \phi_1 \text{ and } A_{i+1} A_{i+2} \ldots \models \phi_1 U \phi_2$$

iff

$$\phi_2 \in B_i \text{ or } (\phi_1 \in B_i \text{ and } \phi_1 U \phi_2 \in B_{i+1})$$

This shows that $B_0 B_1 B_2 \ldots$ is a run of $\mathcal{G}_\phi$.

It remains to prove that this run is accepting, i.e., for each subformula $\phi_{1,j} U \phi_{2,j}$ in closure(ϕ), $B_i \in F_j$ for infinitely many i.

Assume there are finitely many i such that $B_i \in F_j$. We have:

$$B_i \notin F_j = F_{\phi_{1,j} U \phi_{2,j}} \Rightarrow \phi_{1,j} U \phi_{2,j} \in B_i \text{ and } \phi_{2,j} \notin B_i$$

As $B_i = \{\psi \in \text{closure}(\phi) \mid A_i A_{i+1} \ldots \vDash \psi\}$, it follows that if $B_i \notin F_j$, then:

$$A_i A_{i+1} \ldots \vDash \phi_{1,j} U \phi_{2,j} \text{ and } A_i A_{i+1} \ldots \vDash \phi_{2,j}$$

Thus, $A_k A_{k+1} \vDash \phi_{2,j}$ for some $k > i$. By definition of the formula sets B_i, it then follows that $\phi_{2,j} \in B_k$, and by definition of $F_j, B_k \in F_j$. Thus, $B_i \in F_j$ for finitely many i, then $B_k \in F_j$ for infinitely many k. Contradiction.

$\subseteq$

Let $\sigma = A_0 A_1 A_2 \ldots \in \mathcal{L}(\mathcal{G}_\phi)$, i.e., there is an accepting run $B_0 B_1 B_2 \ldots$ for σ in G_ϕ. Since $\delta(B, A) = \emptyset$ for all pairs (B, A) with $A \neq B \cap AP$, it follows that $A_i = B_i \cap AP$ for $i \geq 0$. Thus,
$\sigma = (B_0 \cap AP)(B_1 \cap AP)(B_2 \cap AP) \ldots$
We prove the following more general proposition:

$$\psi \in B_0 \Leftrightarrow A_0 A_1 A_2 \ldots \models \psi$$

by structural induction on the structure of ψ.
Base case: The statement for $\psi = \mathit{true}$ or $\psi = a$ with $a \in AP$ follows directly from the definition of closure.
Induction step: Based on the induction hypothesis that the claim holds for $\psi', \phi_1, \phi_2 \in \mathit{closure}(\phi)$, it is proven that for the formulae

$$\psi = \bigcirc \psi', \psi = \neg \psi', \psi = \phi_1 \wedge \phi_2 \text{ and } \psi = \phi_1 U \phi_2$$

the claim also holds. We only discuss $\psi = \phi_1 U \phi_2$. Let $A_0 A_1 A_2 \ldots \in (2^{AP})^\omega$ and $B_0 B_1 B_2 \ldots \in Q^\omega$ satisfying the constraints (i) and (ii). It is now shown that:

$$\psi \in B_0 \text{ iff } A_0 A_1 A_2 \ldots \models \psi$$

$\Leftarrow$

Assume $A_0 A_1 A_2 \ldots \vDash \psi$ where $\psi = \phi_1 U \phi_2$. Then, there exists $j \geq 0$ such that

$$A_j A_{j+1} \ldots \vDash \phi_2 \text{ and } A_i A_{i+1} \ldots \vDash \phi_1 \text{ for } 0 \leq i < j$$

From the induction hypothesis (applied to ϕ_1 and ϕ_2) it follows that

$$\phi_2 \in B_j \text{ and } \phi_1 \in B_i \text{ for } 0 \leq i < j.$$

By induction on j we obtain: $\phi_1 U \phi_2 \in B_j, B_{j-1} \ldots, B_0$.

$\Rightarrow$

Assume $\phi_1 U \phi_2$ ∈B$_0$. Since B_0 is elementary, $\phi_1 \in B_0$ or $\phi_2 \in B_0$. Distinguish between $\phi_2 \in B_0$ and $\phi_2 \notin B_0$. If $\phi_2 \in B_0$, it follows from the induction hypothesis $A_0 A_1 \ldots \vDash \phi_2$, and thus $A_0 A_1 \ldots \vDash \phi_1 U \phi_2$. This remains the case $\phi_2 \in B_0$. Then $\phi_1 \in B_0$ and $\phi_1 U \phi_2 \in B_0$. Assume $\phi_2 \in B_j$ for all $j \geq 0$. From the definition of the transition relation δ, we obtain using an inductive argument (successively applied to $\phi_1 \in B_j, \phi_2 \in B_j$ and $\phi_1 U \phi_2 \in B_j$ for $j \geq 0$):

$$\phi_1 \in B_j \text{ and } \phi_1 U \phi_2 \in B_j \text{ for all } j \geq 0.$$

As $B_0 B_1 B_2 \ldots$ satisfies constraint (ii), it follows that

$$B_j \in F_{\phi_1 U \phi_2} \text{ for infinitely many } j \geq 0$$

On the other hand, we have

$$\phi_2 \notin B_j \text{ and } \phi_1 U \phi_2 \in B_j \text{ iff } B_j \notin F_{\phi_1 U \phi_2} \text{ for all } j$$

Contradiction! Thus, $\phi_2 \in B_j$ for some $j \geq 0$. Without loss of generality, assume $\phi_2 \notin B_0, \ldots, B_{j1}$, i.e., let j be the smallest index such that $\phi_2 \in B_j$. The induction hypothesis for $0 \leq i < j$ yields

$$\phi_1 \in B_i \text{ and } \phi_1 U \phi_2 \in B_i \text{ for all } 0 \leq i < j$$

From the induction hypothesis applied to ϕ_1 and ϕ_2 it follows that

$$A_j A_{j+1} \ldots \vDash \phi_2 \text{ and } A_i A_{i+1} \ldots \vDash \phi_1 \text{ for } 0 \leq i < j$$

We conclude that $A_0 A_1 A_2 \ldots \vDash \phi_1 U \phi_2$.

Size of GNBA

States in the GNBA $\mathcal{G}_\phi$ are elementary sets of formulae in $closure(\phi)$. Let $subf(\phi)$ denote the set of all subformulae of ϕ. The number of states in $\mathcal{G}_\phi$ is bounded by $2|subf(\phi)|$, the number of possible formula sets. As $|subf(\phi)| \leq |\phi|$, the number of states in the GNBA $\mathcal{G}_\phi$ is bounded by $2^{O(|\phi|)}$.

Conclusion

- Büchi Theorem for checking the decidabilty of MSO
- LTL model-checking problem can be reduced into an emptiness problem for NBA

References I

Christel Baier and Joost-Pieter Katoen.
Principles of model checking.
The MIT Press, Cambridge, Mass, 2008.
OCLC: ocn171152628.

Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors.
Handbook of Model Checking.
Springer International Publishing, Cham, 2018.

Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors.
Automata, logics, and infinite games.
Number 2500 in Lecture notes in computer science. Springer, Berlin ; New York, 2002.

Leonid Libkin.
Elements of finite model theory.
Texts in theoretical computer science. Springer, Berlin, 2010.
OCLC: 837781579.

References II

Shmuel Safra.
Complexity of automata on infinite objects.
PhD Thesis, Citeseer, 1989.

Wolfgang Thomas.
Languages, Automata, and Logic.
In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, pages 389–455. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

Thanks for your attention!