Summer School on Reliable Computing

Introduction to Model Checking

Orna Kupferman

1 Temporal Logic

The logic *LTL* is a linear temporal logic. Formulas of LTL are constructed from a set AP of atomic proposition using the usual Boolean operators and the temporal operators X ("next time") and U ("until"). Formally, given a set AP, an LTL formula is defined as follows:

- **true**, **false**, or p, for $p \in AP$.
- $\neg\psi_1$, $\psi_1 \vee \psi_2$, $X\psi_1$, or $\psi_1 U \psi_2$, where ψ_1 and ψ_2 are LTL formulas.

We define the semantics of LTL with respect to a *computation* $\pi = \sigma_0, \sigma_1, \sigma_2, \ldots$, where for every $j \geq 0$, σ_j is a subset of AP, denoting the set of atomic propositions that hold in the j's position of π. We denote the suffix $\sigma_j, \sigma_{j+1}, \ldots$ of π by π^j. We use $\pi \models \psi$ to indicate that an LTL formula ψ holds in the path π. The relation $\models$ is inductively defined as follows:

- For all π, we have that $\pi \models$ **true** and $\pi \not\models$ **false**.
- For an atomic proposition $p \in AP$, $\pi \models p$ iff $p \in \sigma_0$.
- $\pi \models \neg\psi_1$ iff $\pi \not\models \psi_1$.
- $\pi \models \psi_1 \vee \psi_2$ iff $\pi \models \psi_1$ or $\pi \models \psi_2$.
- $\pi \models X\psi_1$ iff $\pi^1 \models \psi_1$.
- $\pi \models \psi_1 U \psi_2$ iff there exists $k \geq 0$ such that $\pi^k \models \psi_2$ and $\pi^i \models \psi_1$ for all $0 \leq i < k$.

The logic *CTL** is a branching temporal logic. A path quantifier, E ("for some path") can prefix an assertion composed of an arbitrary combination of linear time operators. There are two types of formulas in CTL*: *state formulas*, whose satisfaction is related to a specific state, and *path formulas*, whose satisfaction is related to a specific path. Formally, let AP be a set of atomic proposition names. A CTL* state formula is either:

- **true**, **false**, or p, for $p \in AP$.

- $\neg\varphi_1$ or $\varphi_1 \vee \varphi_2$, where φ_1 and φ_2 are CTL* state formulas.
- $E\psi_1$, where ψ_1 is a CTL* path formula.

A CTL* path formula is either:

- A CTL* state formula.
- $\neg\psi_1$, $\psi_1 \vee \psi_2$, $X\psi_1$, or $\psi_1 U \psi_2$, where ψ_1 and ψ_2 are CTL* path formulas.

The logic CTL* consists of the set of state formulas generated by the above rules. We use the following abbreviations:

- $\wedge, \rightarrow$, and $\leftrightarrow$, interpreted in the usual way.
- $A\psi = \neg E \neg \psi$ ("in all paths").
- $F\psi = \mathbf{true} U \psi$ ("eventually", and the "F" comes from "Future").
- $G\psi = \neg F \neg \psi$ ("always", and the "G" comes from "Globally").

The logic *CTL* is a restricted subset of CTL*. In CTL, the temporal operators X, U, and their negations must be immediately preceded by a path quantifier. Formally, it is the subset of CTL* obtained by restricting the path formulas to be $X\varphi_1$, $\varphi_1 U \varphi_2$, or their negations, where φ_1 and φ_2 are CTL state formulas.

We define the semantics of CTL* with respect to a *Kripke structure* $K = \langle AP, W, R, w^0, L\rangle$, where AP is the set of atomic propositions, W is a set of states, $R \subseteq W \times W$ is a transition relation that must be total in its first element (i.e., for each $w \in W$ there exists at least one w' with $\langle w, w'\rangle \in R$), $w^0 \in W$ is an initial state, and $L : W \rightarrow 2^{AP}$ maps each state to a set of atomic propositions true in this state. A *path* in K is an infinite sequence of states $\pi = w_0, w_1, \ldots$, such that for all $i \geq 0$, we have $\langle w_i, w_{i+1}\rangle \in R$.

We use $w \models \varphi$ to indicate that a state formula φ holds at state w (assuming an agreed Kripke structure K). The relation $\models$ is inductively defined as follows (the relation $\pi \models \psi$ for a path formula ψ is the same as for ψ in LTL, adding the case where ψ is a state formula):

- For all w, we have that $w \models \mathbf{true}$ and $w \not\models \mathbf{false}$.
- For an atomic proposition $p \in AP$, $w \models p$ iff $p \in L(w)$.
- $w \models \neg\varphi_1$ iff $w \not\models \varphi_1$.
- $w \models \varphi_1 \vee \varphi_2$ iff $w \models \varphi_1$ or $w \models \varphi_2$.
- $w \models E\psi_1$ iff there exists a path $\pi = w_0, w_1, \ldots$ such that $w_0 = w$ and $\pi \models \psi_1$.

- $\pi \models \varphi$ for a state formula φ iff $w_0 \models \varphi$.

For a Kripke structure K, and a CTL* formula φ, we say that K satisfies φ, denoted $K \models \varphi$, if $w^0 \models \varphi$. We often interpret LTL formulas over Kripke structures too, and we say that K satisfies an LTL formula ψ if all the paths of K that strat at w^0 satisfy ψ.

We say that two formulas φ_1 and φ_2 are *equivalent* ($\varphi_1 \sim \varphi_2$) if for every Kripke structure K, we have $K \models \varphi_1$ iff $K \models \varphi_2$.

The *model-checking problem* is to determine, given a temporal logic formula ψ and a Kripke structure K, whether K satisfy ψ.

1.1 Some facts

1. The model-checking problem $K \models \varphi$ can be solved in time $O(|K| \cdot |\varphi|)$ for φ in CTL.

2. The model-checking problem $K \models \varphi$ can be solved in time $|K| \cdot 2^{O(|\varphi|)}$ and is PSPACE-complete for φ in LTL or CTL*.

2 Büchi word automata

Let Σ be a finite alphabet. A *Büchi word automaton* is $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$, where Σ is the input alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $Q_0 \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of accepting states. Since $\mathcal{A}$ may have several initial states and since the transition function may specify many possible transitions for each state and letter, $\mathcal{A}$ may be *nondeterministic*. If $|Q_0| = 1$ and δ is such that for every $q \in Q$ and $\sigma \in \Sigma$, we have that $|\delta(q, \sigma)| \leq 1$, then $\mathcal{A}$ is a *deterministic* automaton. We use NBW and DBW to denote nondeterministic and deterministic Büchi automata, respectively.

Given an input word $w = \sigma_0 \cdot \sigma_1 \cdots$ in Σ^ω, a *run* of $\mathcal{A}$ on σ is a function $r : \mathbb{N} \to Q$ where $r(0) \in Q_0$ and for every $i \geq 0$, we have $r(i+1) \in \delta(r(i), \sigma_i)$; i.e., the run starts in one of the initial states and obeys the transition function. Note that a nondeterministic automaton can have many runs on w. In contrast, a deterministic automaton has a single run on w. For a run r, let $\mathit{inf}(r)$ denote the set of states that r visits infinitely often. That is,

$$\mathit{inf}(r) = \{q \in Q \ : r(i) = q \text{ for infinitely many } i \geq 0\}.$$

As Q is finite, it is guaranteed that $\mathit{inf}(r) \neq \emptyset$. The run r is *accepting* iff $\mathit{inf}(r) \cap F \neq \emptyset$. That is, iff there exists a state in F that r visits infinitely often. A run that is not accepting is *rejecting*. An automaton $\mathcal{A}$ accepts an input word w iff there exists an accepting run of $\mathcal{A}$ on w.

A *generalized Büchi automaton* is $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$, where Σ, Q, δ, and Q_0 are as in Büchi automata, and the acceptance condition $F \subseteq 2^Q$ consists of sets $F_i \subseteq Q$. A run r of $\mathcal{A}$ with $F = \{F_1, F_2, \ldots, F_k\}$ is accepting iff $inf(r) \cap F_i \neq \emptyset$ for all $1 \leq i \leq k$. That is, r is accepting if every set in F is visited infinitely often.

2.1 Some facts

1. NBW are strictly more expressive than DBW.
2. NBW are closed under intersection.
3. The nonemptiness problem for NBW is NLOGSPACE-complete.

3 From LTL to Büchi automata

Given an LTL formula ψ, we construct a generalized Büchi word automaton $\mathcal{A}_\psi$ such that $\mathcal{A}_\psi$ accepts exactly all the computations that satisfy ψ. The construction was first suggested by Vardi and Wolper in 1986.

For an LTL formula ψ, the *closure* of ψ, denoted $cl(\psi)$, is the set of ψ's subformulas and their negation ($\neg\neg\psi$ is identified with ψ). Formally, $cl(\psi)$ is the smallest set of formulas that satisfy the following.

- $\psi \in cl(\psi)$.
- If $\psi_1 \in cl(\psi)$ then $\neg\psi_1 \in cl(\psi)$.
- If $\neg\psi_1 \in cl(\psi)$ then $\psi_1 \in cl(\psi)$.
- If $\psi_1 \wedge \psi_2 \in cl(\psi)$ then $\psi_1 \in cl(\psi)$ and $\psi_2 \in cl(\psi)$.
- If $X\psi_1 \in cl(\psi)$ then $\psi_1 \in cl(\psi)$.
- If $\psi_1 U \psi_2 \in cl(\psi)$ then $\psi_1 \in cl(\psi)$ and $\psi_2 \in cl(\psi)$.

For example, $cl(p \wedge ((Xp)Uq))$ is

$\{p \wedge ((Xp)Uq), \neg(p \wedge ((Xp)Uq)), p, \neg p, (Xp)Uq, \neg((Xp)Uq), Xp, \neg Xp, q, \neg q\}$.

Given an LTL formula ψ over AP, we define $\mathcal{A}_\psi = \langle 2^{AP}, Q, \delta, Q_0, F \rangle$, where

- We say that a set $S \subseteq cl(\psi)$ is *good in* $cl(\psi)$ if S is a maximal set of formulas in $cl(\psi)$ that does not have propositional inconsistency. Thus, S satisfies the following conditions.

1. For all $\psi_1 \in cl(\psi)$, we have $\psi_1 \in S$ iff $\neg\psi_1 \notin S$, and
2. For all $\psi_1 \wedge \psi_2 \in cl(\psi)$, we have $\psi_1 \wedge \psi_2 \in S$ iff $\psi_1 \in S$ and $\psi_2 \in S$.

The state space $Q \subseteq 2^{cl(\psi)}$ is the set of all the good sets in $cl(\psi)$.

- Let S and S' be two good sets in $cl(\psi)$, and let $\sigma \subseteq AP$ be a letter. Then $S' \in \delta(S, \sigma)$ if the following hold.

  1. $\sigma = S \cap AP$,
  2. For all $X\psi_1 \in cl(\psi)$, we have $X\psi_1 \in S$ iff $\psi_1 \in S'$, and
  3. For all $\psi_1 U \psi_2 \in cl(\psi)$, we have $\psi_1 U \psi_2 \in S$ iff either $\psi_2 \in S$ or both $\psi_1 \in S$ and $\psi_1 U \psi_2 \in S'$.

  Note that the last condition also means that for all $\neg(\psi_1 U \psi_2) \in cl(\psi)$, we have that $\neg(\psi_1 U \psi_2) \in S$ iff $\neg\psi_2 \in S$ and either $\neg\psi_1 \in S$ or $\neg(\psi_1 U \psi_2) \in S'$.

- $Q_0 \subseteq Q$ is the set of all states $S \in Q$ for which $\psi \in S$.

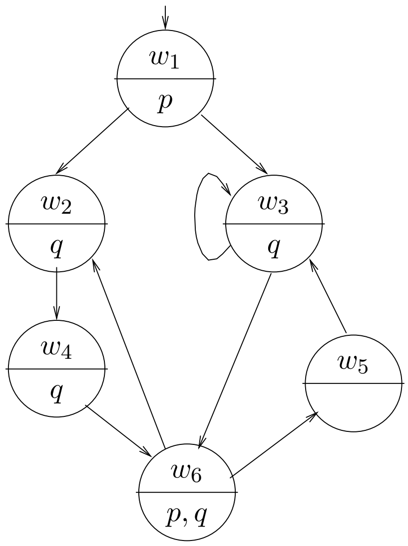
- Every formula $\psi_1 U \psi_2$ contributes to F the set

$$F_{\psi_1 U \psi_2} = \{S \in Q : \psi_2 \in S \text{ or } \neg(\psi_1 U \psi_2) \in S\}.$$

4 Exerecises

Question 1

Consider the Kripke structure $K = \langle \{p, q\}, \{w_1, w_2, \ldots, w_6\}, R, \{w_1\}, L \rangle$ below.



Consider the following CTL* formulas.

1. AXq.
2. $AXAXq$.
3. $AGFq$.
4. $EGF((\neg p) \wedge \neg q)$.
5. $AX(AqUp)$.
6. $AX(EqUp)$.
7. $AF(p \wedge q)$.
8. $(AFp) \wedge AFq$.
9. $EFG(p \vee (Xp) \vee XXp)$.

1. For each formula above φ, answer the following questions:

   (a) Is φ a CTL formula?

   (b) Does K satisfy φ? If the answer is "no", describe a counter example.

2. For a state w_i and a subset $S \subseteq \{p, q\}$, let $K_{S,i}$ be the Kripke structure obtained from K by changing $L(w_i)$ to S. We say that $L(w_i)$ *is not important* if for all $S \subseteq \{p, q\}$, the Kripke structure $K_{S,i}$ agrees with K about the satisfaction of all the formulas in question 1. Similarly, $L(w_i)$ *can be changed* if there is $S \subseteq \{p, q\}$ such that $K_{S,i}$ agrees with K about the satisfaction of all the formulas in question 1. Fot each state w_i, decide whether $L(w_i)$ is not important or can be changed, and justify your decision.

Question 2

For each pair $\varphi_1; \varphi_2$ of formulas below, decide which of the following hold (note that possibly both a and b hold, or none of them):

a. $\varphi_1 \rightarrow \varphi_2$.

b. $\varphi_1 \leftarrow \varphi_2$.

When the an implication does not hold, describe a counter example (when a does not hold, describe a model for φ_1 that does not satisfy φ_2, and when b does not hold, describe a model for φ_2 that does not satisfy φ_1).

$$
\begin{array}{lllll}
\mathbf{1}. & AGp & ; & \neg EF \neg p \\
\mathbf{2}. & EG(p \vee q) & ; & EGp \vee EGq \\
\mathbf{3}. & AG(p \wedge q) & ; & AGp \wedge AGq \\
\mathbf{4}. & AqUp & ; & p \wedge EqUp \\
\mathbf{5}. & A(pU(qUr)) & ; & A((pUq)Ur) \\
\mathbf{6}. & E(pU(qUr)) & ; & E((pUq)Ur) \\
\mathbf{7}. & p \wedge EXq & ; & EpUq \wedge EqUp
\end{array}
$$

Question 3

We define the operator *strict until*, denoted by SU, as follows.

$$pSUq = X(pUq).$$

1. Show that X, U, and G, can can be specified with SU (and boolean operators). Thus, describe LTL formulas whose only temporal operator is SU, equivalent to Xp, pUq, and Gp.

2. Describe CTL formulas equivalent to the CTL* formulas $E(X(pUq))$ and $E((Xp)Uq)$, and prove the equivalence.

Question 4

The question refers to the language

$$L = \{w : w \text{ has infinitely many } a\text{'s and infinitely many } b\text{'s}\}$$

over the alphabet $\{a, b\}$.

1. Describe two different (non isomorphic) DBWs with three states for L.
2. Prove that there is no DBW with two states for L.

Question 5

Prove or give a counter example:

1. Every NBW $\mathcal{A}$ has an equivalent NBW $\mathcal{A}'$ with a single initial state.
2. Every NBW $\mathcal{A}$ has an equivalent NBW $\mathcal{A}'$ with a single accepting state.

Question 6

Consider a nondeterministic word automaton $\mathcal{A}$. Let $\mathcal{L}_*(\mathcal{A})$ be the language of $\mathcal{A}$ when regarded as an automaton on finite words, and let $\mathcal{L}_\omega(\mathcal{A})$ be the language of $\mathcal{A}$ when regarded as a Büchi automaton. Prove or give a counter example:

1. $\mathcal{L}_\omega(\mathcal{A}) = lim(\mathcal{L}_*(\mathcal{A}))$.
2. $\mathcal{L}_\omega(\mathcal{A}) = lim(\mathcal{L}_*(\mathcal{A}))$ iff there is some DBW that recognizes $\mathcal{L}_\omega(\mathcal{A})$.