Automata and Logic (CSC2428)
Lecture 7

Prof. Leonid Libkin
notes by Marcelo Arenas

March 3, 2005

1 Kamp's Theorem

Let Σ be a finite set of propositional letters. The syntax of Linear Temporal Logic is defined by the following grammar:

$$\varphi \;:=\; a,\, a \in \Sigma \;\mid\; \neg\varphi \;\mid\; \varphi \vee \varphi \;\mid\; X\varphi \;\mid\; Y\varphi \;\mid\; \varphi U \varphi \;\mid\; \varphi S \varphi.$$

The semantics of LTL is defined as follows. Given an infinite string $w \in \Sigma^\omega$, let a_i be the i-th letter of w. Then

$$\begin{array}{lcl}
(w,n) \models a & \Leftrightarrow & a_n = a \\
(w,n) \models \neg\varphi & \Leftrightarrow & \text{it is not the case that } (w,n) \models \varphi \\
(w,n) \models \varphi \vee \psi & \Leftrightarrow & (w,n) \models \varphi \text{ or } (w,n) \models \psi \\
(w,n) \models X\varphi & \Leftrightarrow & (w,n+1) \models \varphi \\
(w,n) \models Y\varphi & \Leftrightarrow & n > 0 \text{ and } (w,n-1) \models \varphi \\
(w,n) \models \varphi U \psi & \Leftrightarrow & \exists i > n\ (w,i) \models \psi \text{ and } \forall j \in [n,i)\ (w,j) \models \varphi \\
(w,n) \models \varphi S \psi & \Leftrightarrow & n > 0 \text{ and } \exists i < n\ (w,i) \models \psi \text{ and } \forall j \in (i,n]\ (w,j) \models \varphi
\end{array}$$

We also consider here the following abbreviations:

$$\begin{array}{rcl}
\top & = & \bigvee_{a \in \Sigma} a, \\
\bot & = & \neg\top, \\
F\varphi & = & \top U \varphi.
\end{array}$$

Given an ω-string of the form $a_0 a_1 \cdots$, we define a structure $\mathfrak{M}_w$ over the vocabulary $\{<, (P_a)_{a\in\Sigma}\}$ as follows. The domain of $\mathfrak{M}_w$ is $\mathbb{N}$, $<$ is interpreted as the usual order on the natural numbers and P_a ($a \in \Sigma$) is defined as $\{n \mid a_n = a\}$.

By representing ω-strings as structures we can also use FO logic to express properties of strings. Thus, it is a natural question whether FO over strings is more (or less) expressive than LTL.

Theorem 1 (Kamp, 1968) *Over ω-strings, LTL = FO. That is,*

(a) For every LTL formula φ, there exists an FO formula $\alpha_\varphi(x)$ such that, for every ω-string w and every natural number n: $(w, n) \models \varphi$ if and only if $\mathfrak{M}_w \models \alpha_\varphi(n)$.

(b) For every FO formula $\alpha(x)$ with one free variable, there exists an LTL formula φ_α such that for every ω-string w and every natural number n: $\mathfrak{M}_w \models \alpha(n)$ if and only if $(w, n) \models \varphi_\alpha$.

Moreover, for every FO sentence α, there exists an LTL formula φ_α such that for every ω-string w: $\mathfrak{M}_w \models \alpha$ if and only if $(w, 0) \models \varphi_\alpha$.

We note that this theorem also holds for finite strings.

The proof of the first part of Kamp's Theorem is straightforward, given the definition of the semantics of LTL. For the proof of the second part we need a syntactic property of LTL formulae called separation property. We show this property next.

Given a set Σ of propositional letters, we say that an LTL formula is a *future* formula if it is given by the following grammar:

$$\varphi \;:=\; a,\, a \in \Sigma \;\mid\; \neg\varphi \;\mid\; \varphi \vee \varphi \;\mid\; X\varphi \;\mid\; \varphi U \varphi,$$

and we say that an LTL formula is a *past* formula if it is given by the following grammar:

$$\psi \;:=\; a,\, a \in \Sigma \;\mid\; \neg\psi \;\mid\; \psi \vee \psi \;\mid\; Y\psi \;\mid\; \psi S \psi.$$

Moreover, we say that an LTL formula has the *separation property* if it is equivalent to a Boolean combination of past and future formulae.

Theorem 2 *All LTL formulae have the separation property.*

We say that two LTL formulae φ and ψ are *initially equivalent* if for every ω-string w, we have that $(w, 0) \models \varphi$ if and only if $(w, 0) \models \psi$. Since every LTL formula of the form either $Y\varphi$ or $\varphi S \psi$ is false in $(w, 0)$, by Theorem 2 we conclude that

Corollary 1 *Every LTL formula is initially equivalent to a future formula.*

We use this corollary in the proof of the second part of Kamp's Theorem. In this proof, we also use Ehrenfeucht-Fraïssé games (see Lecture 3 for a description of these games).

Given structures $\mathfrak{A}$ and $\mathfrak{B}$, we say that $\mathfrak{A} \equiv_n \mathfrak{B}$ if the duplicator has a winning strategy in the n-round Ehrenfeucht-Fraïssé game on structures $\mathfrak{A}$ and $\mathfrak{B}$.

Theorem 3 *$\mathfrak{A} \equiv_n \mathfrak{B}$ if and only if $\mathfrak{A}$ and $\mathfrak{B}$ agree on all FO sentences of quantifier rank at most n. Moreover, $\equiv_n$ has finitely many equivalence classes and each is definable by a FO sentence of quantifier rank n.*

Now we are ready to prove the second part of Kamp's Theorem. In this proof we only consider sentences. The proof for FO formulae with one free variable is similar.

Proof of Theorem 1 (b): The proof is by induction on the quantifier rank of sentence α. We assume that the vocabulary also contains a constant $\underline{min}$ interpreted as 0, so that there exist sentences of quantifier rank 0.

The basis case holds trivially: $P_a(\underline{min})$ ($a \in \Sigma$) and $\underline{min} < \underline{min}$ are the only atomic sentences and they are equivalent to LTL formulae a and $\perp$, respectively.

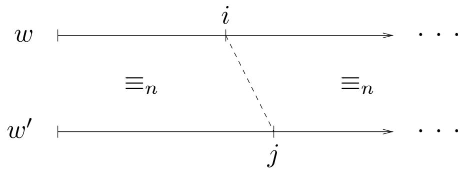
Assume that the property holds for sentence of quantifier rank n, and assume that the quantifier rank of α is $n+1$.

Since α is a sentence, it is a Boolean combination of sentences of the form $\exists x\beta(x)$, where $\beta(x)$ is a FO formula of quantifier rank n. Thus, given that LTL is closed under Boolean combinations, we only need to prove that $\exists x\beta(x)$ is equivalent to an LTL formula.

To prove that the property also holds for $\exists x\beta(x)$, we need to introduce some terminology and prove a claim. Given an ω-string $w = a_0a_1\cdots$ and a natural number i, we denote by $\mathfrak{M}_{a_0\cdots a_i}$ the finite structure representing string $a_0\cdots a_i$ and by $\mathfrak{M}^i_w$ the infinite structure representing ω-string $a_ia_{i+1}\cdots$.

Claim 1 *Let $w = a_0a_1\cdots$, $w' = a'_0a'_1\cdots$ be ω-strings and i, j natural numbers. If $\mathfrak{M}_{a_0\cdots a_i} \equiv_n \mathfrak{M}_{a'_0\cdots a'_j}$ and $\mathfrak{M}^i_w \equiv_n \mathfrak{M}^j_{w'}$, then $(\mathfrak{M}_w, i) \equiv (\mathfrak{M}_{w'}, j)$,*

The assertion of this claim is depicted in the following figure.



Given two ω-strings w and w' and natural numbers i, j, if the duplicator has winning strategies in the n-round game on $\mathfrak{M}_{a_0\cdots a_i}$ and $\mathfrak{M}_{a'_0\cdots a'_j}$, and in the n-round game on $\mathfrak{M}^i_w$ and $\mathfrak{M}^j_{w'}$, then she has a winning strategy in the n-round game on $(\mathfrak{M}_w, i)$ and $(\mathfrak{M}_{w'}, j)$ which is the composition of the two previous strategies. More precisely, if the spoiler picks i in w (j in w'), then the duplicator responds with j in w' (i in w). If the spoiler picks $k < i$ in w ($k < j$ in w'), then the duplicator responds by using her winning strategy in the n-round game on $\mathfrak{M}_{a_0\cdots a_i}$ and $\mathfrak{M}_{a'_0\cdots a'_j}$. Finally, if the spoiler picks $k > i$ in w ($k > j$ in w'), then the duplicator responds by using her winning strategy in the n-round game on $\mathfrak{M}^i_w$ and $\mathfrak{M}^j_{w'}$.

Let $\tau_1, \ldots \tau_M$ be pairwise non-equivalent sentences of quantifier rank n enumerating the equivalence classes of $\equiv_n$ (such formulae exist by Theorem 3). Then by Claim 1 there exists pairs (τ_{j_1}, τ_{k_1}), $\ldots$, (τ_{j_m}, τ_{k_m}) such that for every ω-string $w = a_0a_1\cdots$ and natural number i, $\mathfrak{M}_w \models \beta(i)$ if and only if there exists $l \in [1, m]$ such that $\mathfrak{M}_{a_0\cdots a_i} \models \tau_{j_l}$ and $\mathfrak{M}^i_w \models \tau_{k_l}$. By induction hypothesis, there exists an LTL formula σ_{k_l} equivalent to τ_{k_l}, for every $l \in [1, m]$. Furthermore, by induction hypothesis, there exists an LTL formula γ_{j_l} such that $\mathfrak{M}_{a_0\cdots a_i} \models \tau_{j_l}$ if and only if the reversed string $a_i\cdots a_0$ satisfies

γ_{j_l} $((a_i \cdots a_0, 0) \models \gamma_{j_l})$, for every $l \in [1, m]$. By Corollary 1, there exists a future LTL formula γ'_{j_l} which is initially equivalent to γ_{j_l}, for every $l \in [1, m]$. Thus, by replacing X by Y and U by S in each formula γ'_{j_l} we obtain past formulae $\sigma_{j_1}, \ldots, \sigma_{j_m}$ that together with LTL formulae $\sigma_{k_1}, \ldots, \sigma_{k_m}$ shown above defined an LTL formula equivalent to $\exists x \beta(x)$. More precisely, for every ω-string w:

$$\mathfrak{M}_w \models \exists x \beta(x) \quad \text{if and only if} \quad (w, 0) \models \bigvee_{l=1}^{m} F(\sigma_{j_l} \wedge \sigma_{k_l}).$$

This conclude the proof of Kamp's Theorem. □

2 Vardi and Wolper's Theorem

By Kamp's Theorem we know that LTL = FO over ω-strings. Thus, given that FO $\subseteq$ MSO and that for every MSO sentence φ there exists a Büchi automaton $\mathcal{A}_\varphi$ such that for every ω-string w, $\mathfrak{M}_w \models \varphi$ if and only if $w \in L(\mathcal{A}_\varphi)$, we can verify whether a program P satisfies a specification given by an LTL formula φ as follows. Assuming that P is a finite automaton, we construct Büchi automaton $\mathcal{A}_{\neg\varphi}$, and then we construct an automaton accepting the language $L(P) \cap L(\mathcal{A}_{\neg\varphi})$. We use this automaton to verify whether $L(P) \cap L(\mathcal{A}_{\neg\varphi}) = \emptyset$.

The previous approach has a problem. In general, the size of Büchi automaton $\mathcal{A}_\varphi$ for an MSO sentence φ is non-elementary on $\|\varphi\|$. In the following theorem, we show that it is possible to obtain smaller Büchi automata for LTL formulae by translating them directly without going through the general transformation for MSO sentences.

Given a finite set of propositional letters Σ, from now on we consider LTL formulae given by the grammar:

$$\varphi \;\; := \;\; a,\, a \in \Sigma \;\mid\; \neg\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \varphi \vee \varphi \;\mid\; X\varphi \;\mid\; \varphi U \varphi,$$

ans whose semantics is defined for ω-strings over 2^Σ, that is, for ω-strings S of the form $S_0 S_1 S_2 \cdots$, where each $S_i \subseteq \Sigma$. The only difference with the original semantics is the definition of the semantics for the base case: $(S, i) \models a$ if and only if $a \in S_i$. In particular, we say that $S \models a$ if and only if $a \in S_0$.

Theorem 4 (Vardi, Wolper) *For every LTL formula φ, there is a Büchi automaton A_φ such that A_φ accepts exactly the ω-strings satisfying φ and $\|\mathcal{A}_\varphi\|$ is $2^{O(\|\varphi\|)}$.*

Corollary 2 *LTL model checking can be done in time $\|P\| \cdot 2^{O(\|\varphi\|)}$.*

Proof of Theorem 4: We first show how to push the negation to the propositional letters. To do this, we introduce a new temporal connective R (release). The semantics of this connective is as follows:

$$(w, n) \models \varphi R \psi \quad \Leftrightarrow \quad \forall i > n \; (w, i) \models \neg\psi \rightarrow \exists j \in [n, i) \; (w, j) \models \varphi.$$

¿From now on, we assume that LTL also contains temporal connective R. Then we use the following rules to push the negations to the propositional letters:

$$
\begin{array}{lcl}
\neg(\neg\varphi) & \longleftrightarrow & \varphi, \\
\neg(\varphi \wedge \psi) & \longleftrightarrow & \neg\varphi \vee \neg\psi, \\
\neg(\varphi \vee \psi) & \longleftrightarrow & \neg\varphi \wedge \neg\psi, \\
\neg(X\varphi) & \longleftrightarrow & X\neg\varphi \\
\neg(\varphi U \psi) & \longleftrightarrow & (\neg\varphi)R(\neg\psi), \\
\neg(\varphi R \psi) & \longleftrightarrow & (\neg\varphi)U(\neg\psi).
\end{array}
$$

Second, we inductively define the *closure* of an LTL formula φ, denoted by $C(\varphi)$, as follows.

$$
\begin{array}{lcl}
C(a) & = & \{a\}, \\
C(\neg a) & = & \{\neg a\}, \\
C(\varphi \wedge \psi) & = & C(\varphi) \cup C(\psi) \cup \{\varphi \wedge \psi\}, \\
C(\varphi \vee \psi) & = & C(\varphi) \cup C(\psi) \cup \{\varphi \vee \psi\}, \\
C(X\varphi) & = & C(\varphi) \cup \{X\varphi\}, \\
C(\varphi S \psi) & = & C(\varphi) \cup C(\psi) \cup \{\varphi S \psi\}, \\
C(\varphi R \psi) & = & C(\varphi) \cup C(\psi) \cup \{\varphi R \psi\}.
\end{array}
$$

We note that $|C(\varphi)|$ is $O(\|\varphi\|)$.

Third, given an LTL formula φ and an ω-string S, we define a labelling sequence $\lambda_{\varphi,S} : \mathbb{N} \to 2^{C(\varphi)}$ as a mapping satisfying all the conditions below:

1. $a \in \lambda_{\varphi,S}(i)$ if and only if $a \in S_i$.
2. $\neg a \in \lambda_{\varphi,S}(i)$ if and only if $a \notin S_i$.
3. If $\alpha \wedge \beta \in \lambda_{\varphi,S}(i)$, then both α and β are in $\lambda_{\varphi,S}(i)$.
4. If $\alpha \vee \beta \in \lambda_{\varphi,S}(i)$, then α or β is in $\lambda_{\varphi,S}(i)$.
5. If $X\alpha \in \lambda_{\varphi,S}(i)$, then α is in $\lambda_{\varphi,S}(i+1)$.
6. If $\alpha U \beta \in \lambda_{\varphi,S}(i)$, then α is in $\lambda_{\varphi,S}(i)$, and β is in $\lambda_{\varphi,S}(i+1)$ or $\alpha U \beta$ is in $\lambda_{\varphi,S}(i+1)$.
7. If $\alpha R \beta \in \lambda_{\varphi,S}(i)$, then α is in $\lambda_{\varphi,S}(i)$, or both β and $\alpha R \beta$ are in $\lambda_{\varphi,S}(i+1)$.
8. If $\alpha U \beta \in \lambda_{\varphi,S}(i)$, then there exists $j > i$ such that β is in $\lambda_{\varphi,S}(j)$.

In the last part of the proof we use the following lemma.

Lemma 1 *For every LTL formula φ and ω-string S, $S \models \varphi$ if and only if there is a labelling sequence $\lambda_{\varphi,S}$ such that $\varphi \in \lambda_{\varphi,S}(0)$.*

To prove the ($\Rightarrow$) direction, we simply create a labelling function such that for every subformula α of φ, $\alpha \in \lambda_{\varphi,S}(i)$ if and only if $(S,i) \models \alpha$. To prove the other direction, we show by induction on LTL formulae that if $\alpha \in \lambda_{\varphi,S}(i)$, then $(S,i) \models \alpha$.

Finally, given an LTL formula φ, we define automaton $\mathcal{A}_\varphi = (Q, 2^\Sigma, Q_0, \delta, \mathcal{F})$ as follows. Instead of having a set of final states, this automaton has a family $\mathcal{F} = \{F_1, \ldots, F_m\}$ of sets of final states (each $F_i \subseteq Q$), and its acceptance condition is that there exists a run ρ such that for every $i \in [1, m]$, $\mathit{Inf}(\rho) \cap F_i \neq \emptyset$, for every $i \in [1, m]$. It is straightforward to construct a usual Büchi automaton equivalent to $\mathcal{A}_\varphi$.

The set of states Q of $\mathcal{A}_\varphi$ is a subset of $2^{C(\varphi)}$ such that $A \in Q$ if and only if A satisfies all the conditions below.

1. For each $a \in \Sigma$, exactly one of a, $\neg a$ is in A.
2. If $\alpha \wedge \beta \in A$, then both α and β are in A.
3. If $\alpha \vee \beta \in A$, then α or β is in A.

Transition function $\delta : Q \times 2^\Sigma \to 2^Q$ is defined as follows. Given A, $A' \in Q$ and $Z \in 2^\Sigma$ such that $\{a \in \Sigma \mid a \in A\} = Z$, we have that $A' \in \delta(A, Z)$ if A, A' satisfy the following conditions:

1. If $X\alpha \in A$, then α is in A'.
2. If $\alpha U \beta \in A$, then α is in A, and β is in A' or $\alpha U \beta$ is in A'.
3. If $\alpha R \beta \in A$, then α is in A, or both β and $\alpha R \beta$ are in A'.

The set of initial states is given by all the states $A \in Q$ such that $\varphi \in A$. Finally, we use $\mathcal{F}$ to enforce the last condition of the definition of labelling sequences. That is, we want to avoid "bad" sequences of states $A_0 A_1 A_2 \cdots$ such that for some subformula $\alpha U \beta$ of φ, $\alpha U \beta$ is in A_i and for every $j > i$, $\beta \notin A_j$. We avoid these sequences by going infinitely often through a state that either does not have $\alpha U \beta$ in it or has β in it. Thus, for each $\alpha U \beta \in C(\varphi)$, we define $F_{\alpha \cup \beta} = \{A \in Q \mid \alpha U \beta \notin A \text{ or } \beta \in A\}$, and we define $\mathcal{F} = \{F_{\alpha U \beta} \mid \alpha U \beta \in C(\varphi)\}$. This concludes the proof of the theorem. $\square$