Linear Temporal Logic

Mahesh Viswanathan

Fall 2018

Linear Temporal Logic (LTL) is a modal logic that was proposed by Amir Pnueli to specify properties of programs, and reason about them. Amir Pnueli got that Turing award for this work. The logic has connectives that allow one to express concepts such as *possibility*, *necessity*, and *existence*. It can express *temporal relationship* between events in an execution. The logic can be seen as describing a formal language of words of infinite length. We shall see that LTL expresses the same class of languages as the infinite word languages definable in first order logic. If LTL and first order logic on words are equi-expressive, then why do we a new language to express properties? LTL is a very different logic than first order logic. In particular, LTL does not have any variables or (explicit) quantification. This has made the logic popular among practitioners, as people find it easier to write properties in a logic without quantification. The second important property is that LTL admits efficient decision procedures. Recall that Meyer's lower bound proves that though MSO on words is decidable, it has a large complexity. We will show that LTL on the other hand admits decision procedures in PSPACE.

1 Syntax and Semantics

LTL is a modal logic that is built over a set of propositions. Let us fix a finite set of propositions $\mathcal{P} = \{p_1, p_2, \ldots p_n\}$. The formulas will be a sequence of symbols, where each symbol is one of the following:

- The symbol $\perp$ called *false*;
- An element $p_i \in \mathcal{P}$;
- The symbol $\rightarrow$ called *implication*;
- The symbol X called *next*;
- The symbol U called *until*;
- The symbols (and) called *parenthesis*.

The well formed formulas in the logic are defined inductively, as always. The formal definition is next.

Definition 1. A *(well formed) formula (wff)* over propositions $\mathcal{P} = \{p_1, p_2, \ldots p_n\}$ is inductively defined as follows

1. $\perp$ is a wff.
2. p_i is a wff, where $p_i \in \mathcal{P}$.
3. If φ and ψ are wffs then $(\varphi \rightarrow \psi)$ is a wff.
4. If φ is wff then $(X\varphi)$ is a wff.
5. If φ and ψ are wffs then $(\varphi U \psi)$ is a wff.

As always, the following abbreviations will be used: $\neg\varphi$ for the formula $\varphi \to \bot$, $\varphi \vee \psi$ for $(\neg\varphi) \to \psi$, $\varphi \wedge \psi$ for $\neg(\neg\varphi \vee \neg\psi)$, and $\top$ for $\neg\bot$.

The semantics of formulas (using propositions in $\mathcal{P}$) in LTL will be defined over words of infinite length over the alphabet $2^{\mathcal{P}}$. That is, a model for a formula φ will be a $\alpha : \mathbb{N} \to 2^{\mathcal{P}}$. We recall some notation that we have introduced in the past. For a word α, we have the following notation: $\alpha[i]$ denotes the symbol at position i; $\alpha[i, j]$ denotes the subword starting at position i and ends at position j (both included); $\alpha[*, i]$ is the subword $\alpha[0, i]$ or the prefix ending at position i; and finally, $\alpha[i, *]$ is the suffix starting at position at i.

Definition 2. Let us consider a formula φ over propositions $\mathcal{P} = \{p_1, \ldots p_n\}$ and a word $\alpha : \mathbb{N} \to 2^{\mathcal{P}}$. We define the relation $\alpha \models \varphi$ ("the word w satisfies the formula φ") inductively as follows.

- $\alpha \not\models \bot$ for all α
- $\alpha \models p_i$ iff $p_i \in \alpha[0]$
- $\alpha \models \varphi \to \psi$ iff either $\alpha \not\models \varphi$ or $\alpha \models \psi$
- $\alpha \models X\varphi$ iff $\alpha[1, *] \models \varphi$
- $\alpha \models \varphi U \psi$ iff for some j, $\alpha[j, *] \models \psi$ and for all $0 \leq i < j$, $\alpha[i, *] \models \varphi$.

Let us look at some examples to help understand what formulas in LTL can describe.

Example 3. We will consider a few example formulas over the propositions $\{p, q\}$. Models of such formulas are infinite words over the alphabet $\Sigma = 2^{\{p,q\}}$. We will denote each symbol in σ by $\binom{b_p}{b_q}$, where $b_p = 1$ iff $p \in A$ and $b_q = 1$ iff $q \in A$. We will also use # to denote any subset, i.e., as a "don't care". Let us look at the basic formulas in LTL.

1. Formula Xp has models that are of the form $\#\binom{1}{0/1}\#^\omega$
2. Formula pUq has models of the form $(\binom{1}{0})^*\binom{0/1}{1}\#^\omega$
3. Formula $Fp = \top Up$ has models of the form $(\binom{0}{0/1})^*\binom{1}{0/1}\#^\omega$
4. Formula $Gp = \neg F \neg p$ has models of the form $(\binom{1}{0/1})^\omega$

Formulas 3 and 4 of Example 3 occur so commonly that they have a special notation. $F\varphi$ read as "eventually/finally φ" will denote the formula $\top U \varphi$. Unrolling the semantics given in Definition 2, we get

$$w \models F\varphi \text{ iff there exists } i,\ w[i, *] \models \varphi.$$

The second commonly used formula is $G\varphi$ which is read as "always/globally φ". This formula will denote φ", will denote the formula $\neg F \neg \varphi$. Using the semantics in Definition 2, we get

$$w \models G\varphi \text{ iff for every } i,\ w[i, *] \models \varphi.$$

Let us look at more examples, using these derived modalities.

Example 4. Once again, we consider formulas over $\{p, q\}$, and we represent elements of $2^{\{p,q\}}$ like we did in Example 3.

1. $G(p \to X(pUq))$ says "whenever p holds, then from the next moment onwards p holds until eventually q holds". The formula is satisfied in the model

$$\alpha = \binom{0}{0}\binom{1}{0}\binom{1}{0}\binom{1}{0}\binom{1}{1}\binom{0}{1}\#^\omega$$

   but it does not hold in $\binom{1}{0}\alpha$

2. If p expresses “process 1 is in critical section” and q expresses “process 2 is in critical section” mutual exclusion is expressed by the property $G(\neg p \vee \neg q)$.

Example 5. Consider the word

$$\alpha = \binom{1}{0}\binom{0}{0}\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{0}{1}\binom{0}{1}^{\omega}$$

where the first row is assignment to p and second row is assignment to q.

- $\alpha \models G(p \rightarrow Fq)$
- $\alpha \not\models G(q \rightarrow Fp)$
- $\alpha \models X(\neg q U p)$
- $\alpha \models \neg q U p$
- $\alpha \not\models pU(p \wedge q)$

2 Expressive Power of LTL

Let fix the set of propositions to be $\mathcal{P}$. Recall that models of LTL over $\mathcal{P}$ are infinite words over $\Sigma = 2^{\mathcal{P}}$. For a LTL formula φ, let $[\![\varphi]\!]$ denote the set of models of φ, i.e.,

$$[\![\varphi]\!] = \{\alpha \in \Sigma^{\omega} \mid \alpha \models \varphi\}.$$

We will show that for any LTL formula φ, $[\![\varphi]\!]$ is a ω-regular language. We will give a translation from an LTL formula φ to a Büchi automaton that recognizes $[\![\varphi]\!]$ in Section 3. In this section we will prove the regularity of $[\![\varphi]\!]$ by proving a stronger result, namely, that $[\![\varphi]\!]$ is definable in first order logic over words. Recall that first order logic for words over Σ are formulas over the signature $(<, S, \{Q_a\}_{a\in\Sigma})$ and for a sentence ψ, let $[\![\psi]\!]$ be the set of word (structures that satisfy ψ. Our main result about LTL expressivenes is the following.

Theorem 6. *For any LTL formula φ, there is a first order sentence (over words) ψ such that $[\![\varphi]\!] = [\![\psi]\!]$.*

Proof. We will establish the theorem by proving a stringer result — for every LTL formula φ, we will construct a first order formula $\mathsf{T}(\varphi)(x)$ with one free variable x, such that

$$\alpha \models \mathsf{T}(\varphi)(x) \text{ iff } w[i,*] \models \varphi.$$

The theorem then follows by taking ψ to be the sentence

$$\psi = \forall x(\mathsf{first}(x) \rightarrow T(\varphi)(x))$$

where $\mathsf{first}(x) = \neg(\exists y Syx)$.

The formula $\mathsf{T}(\varphi)$ will be constructed inductively on the structure of φ. For the base case, we have $\mathsf{T}(\bot)(x) = \bot$, and $\mathsf{T}(p)(x) = Q_p x$. The boolean case is also simple — $\mathsf{T}(\varphi_1 \rightarrow \varphi_2)(x) = \mathsf{T}(\varphi_1)(x) \rightarrow \mathsf{T}(\varphi_2)(x)$.

Let us now consider the different modal operators.

Next $\mathsf{T}(X\varphi_1)(x) = \exists y(Sxy \wedge \mathsf{T}(\varphi_1)(y))$

Until $\mathsf{T}(\varphi_1 U \varphi_2)(x) = \exists y((x \leq y) \wedge \mathsf{T}(\varphi_2)(y) \wedge \forall z(((x \leq z) \wedge (z < y)) \rightarrow \mathsf{T}(\varphi_1)(z)))[x \mapsto i]$, where $x < y$ is the formula $< xy$ and $x \leq y$ is $< xy \vee x = y$.

□

From Büchi’s theorem that says ω-regular languages are MSO definable languages, and from Theorem 6, we have the following immediate corollary.

Corollary 7. *For any LTL formula φ, $[\![\varphi]\!]$ is a ω-regular language.*

The proof of corollary as a consequence of of Büchi's theorem and Theorem 6 results in an automaton of very large size — a tower of exponentials whose height depends on the number of quantifier alternations. We will give a different direct translation from LTL to Büchi automata that will result in an automaton that is only exponentially sized. This efficient translation is one of the reasons for the popularity of LTL over first order logic, as it means that decision procedures for classical problems like satisfiability and validity are more efficient than those for first order logic. We postpone the direct construction of the Büchi automaton to Section 3.

The converse of Theorem 6 also holds. Thus, LTL has exactly the same expressive power as first order logic over words. Proving this converse direction is, however, a difficult result, and we skip its proof.

Theorem 8 (Kamp)**.** *For every first order sentence φ over words, there is an LTL formula ψ such that $[\![\varphi]\!] = [\![\psi]\!]$.*

2.1 MSO and LTL

Theorems 6 and 8 imply that first order logic over words and LTL have the same expressive power. On the other hand, Büchi's theorem says that ω-regularity and MSO have the same expressive power. Syntactically, first order logic is a sublogic of MSO. But is it *semantically* weaker than MSO over words? In other words, are there ω-regular/MSO-definable languages that cannot be expressed in first order logic/LTL? The answer turns out to be true. We will establish this by observing a combinatorial characterization of first order/LTL definable languages.

Definition 9. $A \subseteq \Sigma^\omega$ is said to *non-counting* if there is a number n_0 such that for every $n \geq n_0$ and for every $u, v \in \Sigma^*$ and $\alpha \in \Sigma^\omega$,

$$uv^n\alpha \in A \text{ if and only if } uv^{n+1}\alpha \in A.$$

$A \subseteq \Sigma^\omega$ is said to be *counting* if it is not non-counting. In other words,

$$\begin{array}{ll} \forall n_0 \exists n \geq n_0 \exists u, v \in \Sigma^* \exists \alpha \in \Sigma^\omega. & (uv^n\alpha \in A \wedge uv^{n+1}\alpha \notin A) \vee \\ & (uv^n\alpha \notin A \wedge uv^{n+1}\alpha \in A) \end{array}$$

Let us look at examples of counting and non-counting languages.

Example 10. Let us look at examples of non-counting languages. The simplest example is $A_1 = \{a, b\}^\omega$. To prove that A_1 is non-counting, we can take n_0 to be 0. Notice that for any u, v, α and $n \geq 0$, we have $uv^n\alpha \in A_1$ and $uv^{n+1}\alpha \in A_1$, establishing that A_1 is non-counting.

Consider now the language $A_2 = a^*b\{a, b\}^\omega$. Notice, first that we cannot take $n_0 = 0$. This is because, if $u = \varepsilon$, $v = b$ and $\alpha = a^\omega$ then $uv^0\alpha \notin A_1$ but $uv\alpha \in A_2$. Let us take $n_0 = 1$. Consider any u, v, α and n such that $uv^n\alpha \in A_2$. Then either $u \notin a^*$ or $v \notin a^*$ or $\alpha \neq a^\omega$; in each of these cases, we have $uv^{n+1}\alpha$ is also in A_2. On the other hand, suppose $uv^{n+1}\alpha \in A_2$. We need to show that $uv^n\alpha \in A_2$; this is the direction that failed for $n_0 = 0$. Since $uv^{n+1}\alpha \in A_2$, we have (again) either $u \notin a^*$ or $v \notin a^*$ or $\alpha \neq a^\omega$. In each case, because $n \geq 1$, we have $uv^n\alpha \in A_2$.

Let $A_3 = b(aa^*bb)^\omega$. The choices of $n_0 = 0$ and $n_0 = 1$ do not work — take $u = \varepsilon$, $v = b$, and $\alpha = (abb)^\omega$; we have $uv^0\alpha \notin A_3$, $uv\alpha \in A_3$, and $uv^2\alpha \notin A_3$. Observe that $n_0 = 2$ also does not work because we can take $u = ba$, $v = b$, and $\alpha = (abb)^\omega$. In this case we have $uv^2\alpha \in A_3$ but $uv^3\alpha \notin A_3$. However, taking $n_0 = 3$ does prove that A_3 is non-counting.

Example 11. Let us look at examples of counting languages. Consider $B_1 = (aa)^*b^\omega$. Given any n_0, take $n = n_0$. Take $u = \varepsilon$, $v = a$ and $\alpha = b^\omega$. If n_0 is even then we have $uv^n\alpha \in B_1$ and $uv^{n+1}\alpha \notin B_1$. On the other hand, if n_0 is odd, we have $uv^n\alpha \notin B_1$ but $uv^{n+1}\alpha \in B_1$.

Let $B_2 = b(aa^*(bb)^*)^\omega$; notice the similarity and difference with A_3 of Example 10. Given any n_0, take $n = n_0$, with $u = ba$, $v = b$, and $\alpha = (abb)^\omega$. If n_0 is even then $uv^n\alpha \in B_2$ but $uv^{n+1}\alpha \notin B_2$. On the other hand, if n_0 is odd, $uv^n\alpha \notin B_2$ but $uv^{n+1}\alpha \in B_2$.

LTL can only define non-counting languages. Thus, LTL is expressively limited.

Theorem 12. *For any LTL formula φ, $[\![\varphi]\!]$ is non-counting.*

Proof. We will define the number $n(\varphi)$, which "witnesses" the non-counting property of $[\![\varphi]\!]$, inductively on the structure of φ. Let us start with the base cases. For $\varphi = \bot$, we can take $n(\varphi) = 0$; this works trivially because $[\![\bot]\!] = \emptyset$. On the other hand, for a proposition p, take $n(p) = 1$. We leave the proof that $n(p) = 1$ witnesses the non-counting of $\varphi = p$ to the reader.

Consider $\varphi = \psi_1 \to \psi_2$. Induction hypothesis guarantees constants $n(\psi_1)$ and $n(\psi_2)$ that satisfy the property that for any $n \geq n(\psi_i)$ $(i \in \{1,2\})$ and every u, v, α

$$uv^n\alpha \models \psi_i \text{ iff } uv^{n+1}\alpha \models \psi_i.$$

Therefore, for any $n \geq \max(n(\psi_1), n(\psi_2))$ and u, v, α we have

$$uv^n\alpha \models \psi_i \text{ iff } uv^{n+1}\alpha \models \psi_i.$$

This means that for any $n \geq \max(n(\psi_1), n(\psi_2))$ and u, v, α we have

$$uv^n\alpha \models \varphi \text{ iff } uv^{n+1}\alpha \models \varphi.$$

Thus, we take $n(\varphi) = \max(n(\psi_1), n(\psi_2))$.

Next, let us consider $\varphi = X\psi$. Again, we have inductively defined $n(\psi)$ such that for any $n \geq n(\psi)$ and u, v, α

$$uv^n\alpha \models \psi \text{ iff } uv^{n+1}\alpha \models \psi$$

Let us take $n(\varphi) = n(\psi) + 1$. We need to show that $n(\varphi)$ witnesses the non-counting of $[\![\varphi]\!]$. Consider arbitrary u, v, α and $n \geq n(\varphi)$. There are two cases to consider.

- $u \neq \varepsilon$. In this case, let $u = au'$, where $a \in \Sigma$ and $u' \in \Sigma^*$. We have the following argument.

$$\begin{aligned} au'v^n\alpha \models \varphi \quad & \text{iff} \quad u'v^n\alpha \models \psi \\ & \text{iff} \quad u'v^{n+1}\alpha \models \psi \\ & \text{iff} \quad au'v^{n+1}\alpha \models \varphi \end{aligned}$$

- Suppose $u = \varepsilon$. If $v = \varepsilon$ then we trivially have $uv^n\alpha \models \varphi$ iff $uv^{n+1}\alpha \models \varphi$ because $uv^n\alpha = uv^{n+1}\alpha$. So without loss of generality, let us assume that $v = av'$ with $a \in \Sigma$ and $v' \in \Sigma^*$. In this case we have,

$$\begin{aligned} (av')^n\alpha \models \varphi \quad & \text{iff} \quad (av')(av')^{n-1}\alpha \models \varphi \\ & \text{iff} \quad v'(av')^{n-1}\alpha \models \psi \\ & \text{iff} \quad v'(av')^{n}\alpha \models \psi \\ & \text{iff} \quad (av')(av')^{n}\alpha \models \varphi \\ & \text{iff} \quad (av')^{n+1}\alpha \models \varphi \end{aligned}$$

Finally, consider $\varphi = \psi_1 U \psi_2$. By induction hypothesis we have, for $i \in \{1,2\}$, for any $n \geq n(\psi_1)$ and u, v, α

$$uv^n\alpha \models \psi_i \text{ iff } uv^{n+1}\alpha \models \psi_i.$$

We will show that $n(\varphi) = \max(n(\psi_1), n(\psi_2)) + 1$ demonstrates the non-counting of $[\![\varphi]\!]$. Consider any $n \geq n(\varphi)$, and consider u, v, α such that $uv^n\alpha \models \varphi$. We need to show that $uv^{n+1}\alpha \models \varphi$. Since $uv^n\alpha \models \varphi$, there is j such that $uv^n\alpha[j,*] \models \psi_2$ and for all $i < j$, $uv^n\alpha[i,*] \models \psi_1$. We will consider different cases based on the value of j.

- Suppose $j < |u|$. For any $i < |u|$, we have $uv^n\alpha[i,*] = u[i,*]v^n\alpha$ and $uv^{n+1}\alpha[i,*] = u[i,*]v^{n+1}\alpha$. Since $n > \max(n(\psi_1), n(\psi_2))$, we can conclude that $uv^{n+1}\alpha \models \varphi$.

- Suppose $j \geq |u|$. Take $j' = j + |v|$. Will show $uv^{n+1}\alpha[j',*] \models \psi_2$ and for all $i' < j'$, $uv^{n+1}\alpha[i',*] \models \psi_1$. First observe that $uv^n\alpha[j,*] = uv^{n+1}\alpha[j',*]$. Therefore, $uv^{n+1}\alpha[j',*] \models \psi_2$. To prove that for all $i' < j'$, $uv^{n+1}\alpha[i',*] \models \psi_1$, we consider 3 cases based on the value of i'.
    - Suppose $i' < |u|$. In this case, $uv^{n+1}\alpha[i',*] = (u[i',*])v^{n+1}\alpha \models \psi_1$ because $uv^n\alpha[i',*] = (u[i',*])v^n\alpha \models \psi_1$ and $n > n(\psi_1)$.
    - Suppose $|u| \leq i' < |u| + |v|$. Observe that $uv^{n+1}\alpha[i',*] = (v[i' - |u|,*])v^n\alpha \models \psi_1$ because $uv^n\alpha[i',*] = (v[i' - |u|,*])v^{n-1}\alpha \models \psi_1$ and $n - 1 \geq n(\psi_1)$.
    - Finally, consider $i' \geq |u| + |v|$. Observe that $i = i' - |v| < j' - |v| = j$ and $uv^{n+1}\alpha[i',*] = uv^n\alpha[i,*]$. Thus, $uv^{n+1}\alpha[i',*] \models \psi_1$.

  Let us now prove that if $uv^{n+1}\alpha \models \varphi$ then $uv^n\alpha \models \varphi$. There is a j such that $uv^{n+1}\alpha[j,*] \models \psi_2$ and for all $i < j$, $uv^{n+1}\alpha[i,*] \models \psi_1$. We consider different cases based on the value of j.
    - Consider the case when $j < |u| + |v|$. For $k \in \{1, 2\}$ and $i < |u| + |v|$, we have $uv^{n+1}\alpha[i,*] \models \psi_k$ iff $uv[i,*]v^n\alpha \models \psi_k$ iff $uv[i,*]v^{n-1}\alpha \models \psi_k$ because $n - 1 \geq n(\psi_k)$ iff $uv^n\alpha[i,*] \models \psi_k$. Therefore, we have $uv^n\alpha \models \varphi$ in this case.
    - Consider the case when $j \geq |u| + |v|$. Observe that $uv^{n+1}\alpha[j,*] = uv^n\alpha[j - |v|,*]$. Taking $j' = j - |v|$, we have $uv^{n+1}\alpha[j,*] \models \psi_2$ iff $uv^n\alpha[j',*] \models \psi_2$. For any i' such that $|u| + |v| \leq i' < j' = j - |v|$, we have $i' + |v| < j$ and $uv^{n+1}\alpha[i' + |v|,*] = uv^n\alpha[i',*]$, and so $uv^n\alpha[i',*] \models \psi_1$ in this case. For $i' < |u| + |v|$, we have $uv^{n+1}\alpha[i',*] \models \psi_1$ iff $uv[i',*]v^n\alpha \models \psi_1$ iff $uv[i',*]v^{n-1}\alpha \models \psi_1$ because $n - 1 \geq n(\psi_1)$ iff $uv^n\alpha[i',*] \models \psi_1$.

□

Theorem 12 demonstrates the expressive weakness of LTL (and first order logic). Observe that the examples in Example 11 are all ω-regular languages. By Theorem 12, none of them are expressible in LTL. Hence, we can conclude that there are ω-regular languages that cannot be expressed in LTL.

3 Translating LTL to Büchi Automata

One of the most important discoveries, that led to both the popluarity of LTL and Büchi automata, was the efficient translation of LTL formulas to Büchi automata by Sistla, Vardi, and Wolper. While Theorems 6 and 8 establish the regularity of $[\![\varphi]\!]$ and thereby prove the decidability of the satisfiability and validity problems for LTL, the resulting algorithm is inefficient. In this section, we will show that for a LTL formula of size n, we can construct a Büchi automaton M_φ of size $O(2^n)$, thereby giving us PSPACE decision procedures for LTL. We will translate LTL into *generalized Büchi automata*, not (classical Büchi automata. Generalized Büchi automata recognize the intersection of finite many Büchi automata, all of whom share the same transition structure. Formally, it is given as follows.

Definition 13. A *generalized Büchi automaton* is $M = (Q, \Sigma, \delta, q_0, \mathcal{F})$, where Q, Σ, δ and q_0 are as for Büchi automata, and $\mathcal{F} \subseteq 2^Q$. Let $\mathcal{F} = \{F_1, F_2, \ldots F_m\}$. A run ρ of M is *accepting* if for each i, some state of F_i appears infinitely often in ρ. The language recognized by M is the collection of all input strings on which M has some accepting run.

Another way to define the language recognized by M is as follows. Let M_i be the Büchi automaton $(Q, \Sigma, \delta, q_0, F_i)$. Then the language of M is defined as follows.

$$\mathbf{L}_{\exists\mathsf{B}}(M) = \bigcap_{i=1}^{m} \mathbf{L}_{\exists\mathsf{B}}(M_i)$$

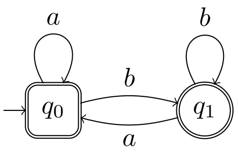

Figure 1: Generalized Büchi automaton recognizing strings with infinitely many as and bs

Example 14. Consider the generalized Büchi automaton $M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{\{q_0\}, \{q_1\}\})$, where $\delta(q, a) = q_0$ and $\delta(q, b) = q_1$, where $q \in \{q_0, q_1\}$. The automaton is shown in Figure 1; the states q_0 and q_1 belonging to different sets within $\mathcal{F}$ are shown in different shapes. The language recognized by M is the collection of strings over $\{a, b\}$ that have infinite many as and infinitely many bs.

Our construction of a generalized Büchi automaton for an LTL formula φ relies on identifying a collection formulas whose truth is necessary to track in order to discover whether φ holds on an infinite string. This collection of formulas is called the *Fischer-Ladner closure* of a formula.

Definition 15. The *Fischer-Ladner closure*, $\mathsf{cl}(\varphi)$, of a formula φ is the smallest set such that

- $\varphi \in \mathsf{cl}(\varphi)$,
- $\mathsf{cl}(\varphi)$ is closed under subformulas,
- If $\psi \in \mathsf{cl}(\varphi)$ then $\neg\psi \in \mathsf{cl}(\varphi)$, where we identify $\neg\neg\psi$ with ψ,
- If $\neg X\psi \in \mathsf{cl}(\varphi)$ then $X\neg\psi \in \mathsf{cl}(\varphi)$, and
- If $\psi_1 U \psi_2 \in \mathsf{cl}(\varphi)$ then $X(\psi_1 U \psi_2) \in \mathsf{cl}(\varphi)$.

It is easy to see that $|\mathsf{cl}(\varphi)| \leq 2n$.

Example 16. Consider $\psi = (\neg h)Uc$ and $\varphi = \neg\psi$. Then

$$\mathsf{cl}(\varphi) = \{\varphi, \psi, X\psi, \neg X\psi, X\neg\psi, \neg h, h, c, \neg c\}$$

Our automata states will track the truth of formulas in $\mathsf{cl}(\varphi)$, i.e., each state will correspond to the set of formulas in $\mathsf{cl}(\varphi)$ that need to hold from then on. This collection of formulas will clearly be consistent (i.e., without contradictions) and complete. Such consistent and complete subsets of $\mathsf{cl}(\varphi)$ will be called *atoms*, and they will form the states of our automaton.

Definition 17. An *atom* is a maximally consistent subset of $\mathsf{cl}(\varphi)$, i.e., $A \subseteq \mathsf{cl}(\varphi)$ is an atom iff

- $\psi \in A$ iff $\neg\psi \notin A$
- $\psi_1 \vee \psi_2 \in A$ iff $\psi_1 \in A$ or $\psi_2 \in A$
- $\psi_1 U \psi_2 \in A$ iff $\psi_2 \in A$ or ($\psi_1 \in A$ and $X(\psi_1 U \psi_2) \in A$).

The collection of all atoms for φ will be denoted by $\mathsf{atom}(\varphi)$.

Example 18. Recall for $\psi = (\neg h)Uc$ and $\varphi = \neg\psi$,

$$\mathsf{cl}(\varphi) = \{\varphi, \psi, X\psi, \neg X\psi, X\neg\psi, \neg h, h, c, \neg c\}$$

The atoms are

$$\begin{array}{ll} A_1 = \{c, h, \psi, \neg X\psi, X\neg\psi\} & A_2 = \{c, h, \psi, X\psi, \neg X\neg\psi\} \\ A_3 = \{\neg c, h, \varphi, \neg X\psi, X\neg\psi\} & A_4 = \{\neg c, h, \varphi, X\psi, \neg X\neg\psi\} \\ A_5 = \{c, \neg h, \psi, \neg X\psi, X\neg\psi\} & A_6 = \{c, \neg h, \psi, X\psi, \neg X\neg\psi\} \\ A_7 = \{\neg c, \neg h, \varphi, \neg X\psi, X\neg\psi\} & A_8 = \{\neg c, \neg h, \psi, X\psi, \neg X\neg\psi\} \end{array}$$

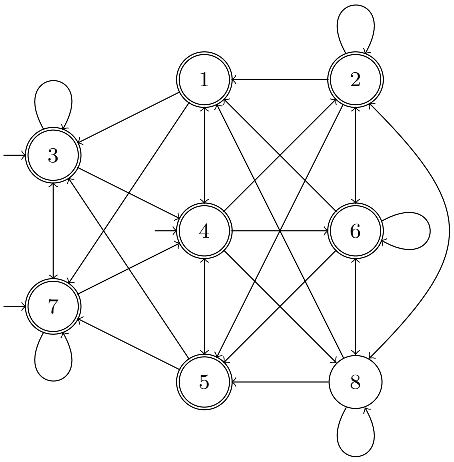

Figure 2: Generalized Büchi automaton for formula φ from Example 20. State i corresponds to atom A_i. Label of edge leaving a state q is the set of propositions that are true in q.

We are now ready to define the generalized Büchi automaton for a given LTL formula.

Definition 19. Consider a LTL formula φ over propositions $\mathcal{P}$, and let $\Sigma = 2^{\mathcal{P}}$. The (generalized) Büchi automata corresponding to φ is $M_\varphi = (Q, \Sigma, \delta, Q_0, \mathcal{F})$ where

- $Q = \mathsf{atom}(\varphi)$,
- $Q_0 = \{A \in \mathsf{atom}(\varphi) \mid \varphi \in A\}$
- $\delta(A, P) = \emptyset$ if $P \neq (A \cap \mathcal{P})$ and $\delta(A, P) = \{B \in Q \mid \forall X\psi \in \mathsf{cl}(\varphi).\ X\psi \in A \text{ iff } \psi \in B\}$ otherwise,
- Let $\{\alpha_1 U\beta_1, \ldots \alpha_k U\beta_k\}$ be all the until formulas in $\mathsf{cl}(\varphi)$. Then $\mathcal{F} = \{F_1, \ldots F_k\}$ such that $F_i = \{A \in \mathrm{atom}(\varphi) \mid \alpha_i U\beta_i \notin A \text{ or } \beta_i \in A\}$

Before proving the correctness of this construction, let us look at an example.

Example 20. Recall for $\psi = (\neg h)Uc$ and $\varphi = \neg\psi$, $\mathsf{cl}(\varphi) = \{\varphi, \psi, X\psi, \neg X\psi, X\neg\psi, \neg h, h, c, \neg c\}$. The atoms are

$$\begin{array}{ll}
A_1 = \{c, h, \psi, \neg X\psi, X\neg\psi\} & A_2 = \{c, h, \psi, X\psi, \neg X\neg\psi\} \\
A_3 = \{\neg c, h, \varphi, \neg X\psi, X\neg\psi\} & A_4 = \{\neg c, h, \varphi, X\psi, \neg X\neg\psi\} \\
A_5 = \{c, \neg h, \psi, \neg X\psi, X\neg\psi\} & A_6 = \{c, \neg h, \psi, X\psi, \neg X\neg\psi\} \\
A_7 = \{\neg c, \neg h, \varphi, \neg X\psi, X\neg\psi\} & A_8 = \{\neg c, \neg h, \psi, X\psi, \neg X\neg\psi\}
\end{array}$$

ψ is the only until formula in $\mathsf{cl}(\varphi)$.

The automaton resulting from the construction described in Definition 19 is shown in Figure 2. State i corresponds to atom A_i. Labels on transitions are not shown; they are assumed to be the set of propositions that are true in the source state of the transition.

Theorem 21. *For formula φ over propositions $\mathcal{P}$, let M_φ be the generalized Büchi automaton constructed in Definition 19. For any $\alpha \in \Sigma^\omega$ (where $\Sigma = 2^{\mathcal{P}}$), $\alpha \in \mathbf{L}_{\exists\mathsf{B}}(M_\varphi)$ if and only if $\alpha \models \varphi$.*

Proof. We begin by first proving the easy direction of the theorem, namely, showing that if $\alpha \models \varphi$ then M_φ has an accepting run. Let $A_i = \{\psi \in \mathsf{cl}(\varphi) \mid \alpha[i, *] \models \psi\}$. Observe that A_i is an atom and $A_0 A_1 \cdots$ is an accepting run of M_φ.

We now prove the other direction. Let $\rho = A_0 A_1 \cdots$ be an accepting run of M_φ on $\alpha = P_0 P_1 \cdots$. We will prove the following stronger statement

$$\alpha[i,*] \models \psi \text{ iff } \psi \in A_i$$

for any $\psi \in \mathsf{cl}(\varphi)$ by induction on the structure of ψ. Let us consider each of the simpler cases, except until.

- *Case $\psi = p$:* $\alpha[i,*] \models p$ iff $p \in P_i$ iff $P_i = A_i \cap \mathcal{P}$ iff $p \in A_i$.
- *Case $\psi = \neg\psi'$:* $\alpha[i,*] \models \psi$ iff $\alpha[i,*] \not\models \psi'$ iff $\psi' \notin A_i$ (by induction hypothesis) iff $\psi = \neg\psi' \in A_i$ (by definition of atom).
- *Case $\psi = \psi_1 \vee \psi_2$:* $\alpha[i,*] \models \psi$ iff $\alpha[i,*] \models \psi_1$ or $\alpha[i,*] \models \psi_2$ iff $\psi_1 \in A_i$ or $\psi_2 \in A_i$ (by induction hypothesis) iff $\psi = \psi_1 \vee \psi_2 \in A_i$ (by definition of atom).
- *Case $\psi = X\psi'$:* $\alpha[i,*] \models \psi$ iff $\alpha[i+1,*] \models \psi'$ iff $\psi' \in A_{i+1}$ (by induction hypothesis) iff $\psi = X\psi' \in A_i$ (because $A_i \xrightarrow{P_i} A_{i+1}$).

We now consider the difficult case of until. Let $\psi = \psi_1 U \psi_2$. Suppose $\alpha[i,*] \models \psi$. Then, there is $k \geq i$ such that $\alpha[k,*] \models \psi_2$ and for all j, $i \leq j < k$, $\alpha[j,*] \models \psi_1$. We will show $\psi \in A_i$ by induction on $k - i$.

- *Case $k - i = 0$:* Since $\alpha[i,*] \models \psi_2$, $\psi_2 \in A_i$. Hence, (by definition of atom) $\psi = \psi_1 U \psi_2 \in A_i$.
- *Case $k - i = \ell + 1$:* This means, $\alpha[i,*] \models \psi_1$ and $\alpha[i+1,*] \models \psi_1 U \psi_2$. By induction hypothesis, $\psi_1 \in A_i$ and $\psi_1 U \psi_2 \in A_{i+1}$. Hence, $X(\psi_1 U \psi_2 \in A_i$ (by definition of transition), and so $\psi_1 U \psi_2 \in A_i$ (by definition of atoms).

We now show that if $\psi \in A_i$ then $\alpha[i,*] \models \psi$. Let ψ be the mth until formula in $\mathsf{cl}(\varphi)$. Since $A_0 A_1 \cdots$ is an accepting run of $\mathcal{A}$, there is $k \geq i$ such that $A_k \in F_m$. We will prove by induction on $k - i$ that $\alpha[i,*] \models \psi$.

- *Case $k - i = 0$:* Since $\psi \in A_i$ and $A_i \in F_m$, it must be the case that $\psi_2 \in A_i$. Then by induction hypothesis, $\alpha[i,*] \models \psi_2$ and so $\alpha[i,*] \models \psi_1 U \psi_2$.
- *Case $k - i = \ell + 1$:* Since $A_i \notin F_m$, (from definition of atoms) $\psi_1 \in A_i$ and $X(\psi_1 U \psi_2) \in A_i$. By induction hypothesis $\alpha[i,*] \models \psi_1$. By the definition of the transition relation, $\psi_1 U \psi_2 \in A_{i+1}$. By induction hypothesis, $\alpha[i+1,*]) \models \psi$. Thus, $\alpha[i,*] \models \psi$.

□