Linear Time Temporal Logic and Model Checking

S. Sheerazuddin

National Institute of Technology, Calicut

Linear-time Temporal Logic : Syntax

Given a set of atomic propositions P the set of all well-formed LTL formulas over P is defined inductively as follows:

$$\alpha, \beta \in \Phi ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \alpha \wedge \beta \mid X\alpha \mid F\alpha \mid G\alpha \mid \alpha U \beta \mid \alpha R \beta$$

where the temporal modalities denote the following:
X – "next", F – "eventually", G – "always", U – "until" and R – "release".

The modalities F, G and R can be expressed in terms of U as follows:

$$F\ \alpha \equiv \mathit{True}\ U\ \alpha$$

$$G\ \alpha \equiv \neg F\ \neg\alpha$$

$$\alpha\ R\ \beta \equiv \neg(\neg\alpha\ U\ \neg\beta)$$

LTL semantics

- LTL formulas are interpreted over sequences of propositional assignments.
- Let P be the set of atomic propositions over which LTL formulas are defined. Then the potential models are $M : \mathbb{N}_0 \to 2^P$.
- That is $M = \nu_0\nu_1\nu_2\cdots$ where for each $i \in \mathbb{N}$, $\nu_i \subseteq P$.
- The satisfiability relation for LTL formulas is defined by induction on the structure as follows:

$$
\begin{aligned}
&M, i \models p \text{ iff } p \in M(i)\\
&M, i \models \neg\alpha \text{ iff } M, i \not\models \alpha\\
&M, i \models \alpha \vee \beta \text{ iff } M, i \models \alpha \text{ or } M, i \models \beta\\
&M, i \models X\alpha \text{ iff } M, i+1 \models \alpha\\
&M, i \models F\alpha \text{ iff } \exists j \geq i,\ M, j \models \alpha\\
&M, i \models G\alpha \text{ iff } \forall j \geq i,\ M, j \models \alpha\\
&M, i \models \alpha U \beta \text{ iff } \exists j \geq i,\ M, j \models \beta \text{ and}\\
&\forall k : i \leq k < j : M, k \models \alpha
\end{aligned}
$$

Büchi Automata

- Let Σ be a finite alphabet.
- An infinite word (or ω-word) over Σ is simply an infinite sequence $a_1 a_2 \cdots$ where each $a_i \in \Sigma$. We shall use Σ^ω to denote the set of all infinite words over the alphabet Σ.
- Let $A = (S, \Sigma, T, I, G)$ be a finite automaton, where G is the set of final (or good) states.
- There is a natural generalization of the notion of a run from finite to infinite words.
- A run over an infinite word $\sigma = a_1 a_2 \cdots$ is a sequence $\rho = s q_1 q_2 \cdots$ with $s \xrightarrow{a1} q1 \xrightarrow{a2} q2 \cdots$, where $s \in I$.
- But, when is such a run accepting?
- In any run ρ, some states of Q are visited only finite number of times and some others are visited infinitely often. Let us call these sets $fin(\rho)$ and $inf(\rho)$.

Büchi Automata

- Büchi's suggestion was to classify a run as accepting if it visits the set F infinitely often.
- Since there are only finitely many states in Q and F, this is equivalent to demanding that the run visit some fixed state in F infinitely often.
- In otherwords, if we regard the state space of a Büchi automaton as a graph, an accepting run traces an infinite path which starts at some state $s \in I$, reaches a good state $g \in F$ and, thereafter, keeps looping back to g infinitely often.
- Formally, a Büchi Automaton is a finite automaton $A = (S, \Sigma, T, I, G)$, and the language accepted by such an automaton is
  $L(A) = \{\sigma \in \Sigma^\omega \mid$ there is a run ρ over σ such that $inf(\rho) \cap G \neq \emptyset\}$.
- A langugage $L \subseteq \Sigma^\omega$ is said to be ω-regular if it is accepted by some Büchi automaton.

Structure of Büchi automata Run

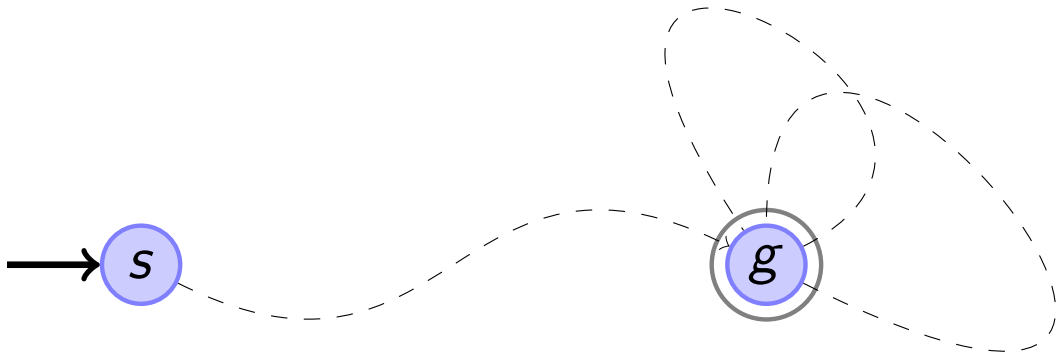

Figure: A typical accepting run of a Büchi automaton with $s \in I$ and $g \in G$.

Büchi Automata : Example

- Consider the alphabet $\Sigma = \{a, b\}$. Let $L \subseteq \Sigma^\omega$ consist of all infinite words α such that there are infinitely many occurrences of $a \in \alpha$. The Figure below shows a Büchi automaton recognizing L.
- In this automaton,all transitions labelled a lead into the good state and, conversely, all transitions coming into the good state are labelled a. From this, it follows that the automaton accepts an infinite word iff it has infinitely many occurrences of a.

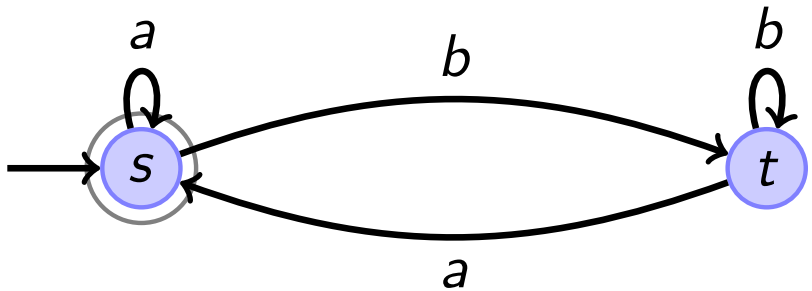

Figure: A Büchi automaton for L

Büchi Automata : Example

- The complement of L, which we denote $\overline{L}$, is the set of all infinite words σ such that σ has only finitely many occurrences of a. An automaton recognizing $\overline{L}$ is shown in Figure below..
- The automaton guesses a point in the input beyond which it will see no more a's – such a point must exist in any input with only a finite number of a's. Once it has made this guess, it can process only b's – there is no transition labelled a from the second state – so if it reads any more a's it gets stuck.

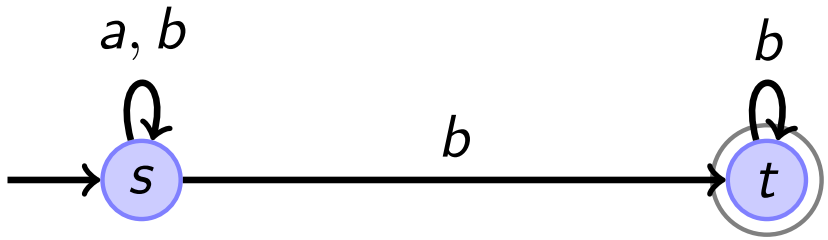

Figure: A Büchi automaton for $\overline{L}$

Deterministic Büchi vs. Non-deterministic Büchi

- In the example, notice that the automaton recognizing L is deterministic while the automaton for $\overline{L}$ is non-deterministic.
- It can be shown that the non-determinism in the second case is unavoidable – that is, there is no deterministic automaton recognizing $\overline{L}$.
- This means that Büchi automata are fundmentally different from their counterparts in finite inputs: we know that over finite words, deterministic automata are as powerful as non-deterministic automata.

Properties of Büchi Automata

It turns out that the class of Büchi-recognizable languages is closed under Boolean operations.

- Union
- Intersection
- Complementation

Language Emptiness of Büchi Automata

- In applications, we need to be able to check whether the language accepted by a Büchi automaton is empty.
- To do this, we recall our observation that any accepting run of a Büchi automaton must begin in an initial state, reach a final state g and then cycle back to g infinitely often.
- If we ignore the labels on the transitions, we can regard the state space of a Büchi automaton A as a directed graph $G_A = (V_A, E_A)$ where $V_A = S$ and $(s, s') \in E_A$ iff for some $a \in \Sigma, s \xrightarrow{a} s'$.
- Recall that a set of vertices X in a directed graph is a strongly connected component iff for every pair of vertices $v, v' \in X$, there is a path from v to v'.

Language Emptiness of Büchi Automata

- Clearly, $L(A)$ is non-empty iff there is a strongly connected component X in G_A such that X contains a vertex g from G and X is reachable from one of the initial states.
- Therefore, the emptiness problem for Büchi automata is decidable.
- Computing the maximal strongly connected components of a directed graph can be done in time linear in the size of the graph, where the size of a graph $G = (V, E)$ is, as usual, given by $|V| + |E|$.
- Checking reachability can also be done in linear time.
- So, if A has n states, checking that $L(A) \neq \emptyset$ can be done in time $O(n^2)$.

- As we saw in the early slides, a model for an LTL formula α is a function $M : \mathbb{N}_0 \longrightarrow 2^P$.
- Note that to check whether α is satisfiable, it suffices to look at models defined over $Voc(\alpha) =$ set of atomic formulas occurring in α.
- In other words, we can restrict our attention to models of the form $P_0 P_1 \cdots$ where each P_i is a subset of $Voc(\alpha)$.
- Since $Voc(\alpha)$ is finite, we can treat each model as an infinite word over the finite alphabet $\Sigma = 2^{Voc(\alpha)}$.
- The result we shall establish is that the set of all infinite words over $2^{Voc(\alpha)}$ which are models for α – i.e., the set

$$Mod(\alpha) = \{M = P_0 P_1 \cdots \mid M, 0 \models \alpha\}$$

  actually constitutes a B üchi recognizable language. That is, $Mod(\alpha) = L(A)$ for some Büchi automaton A.
- We shall also demonstrate how to explicit construct a B üchi automaton A_α over the alphabet $2^{Voc(\alpha)}$ such that $L(A_\alpha) = Mod(\alpha)$.

Formula Automaton for LTL

First, construct the (Fischer-Ladner) closure of α, $CL(\alpha)$, as the smallest set containing α and satisfying the following conditions:

For every $F\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $G\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $X\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $\beta \in CL(\alpha)$, $\neg\beta \in CL(\alpha)$.

For every $\neg\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $F\beta \in CL(\alpha)$, $XF\beta \in CL(\alpha)$.

For every $G\beta \in CL(\alpha)$, $XG\beta \in CL(\alpha)$.

For every $\beta \vee \gamma \in CL(\alpha)$, $\beta, \gamma \in CL(\alpha)$.

For every $\beta \wedge \gamma \in CL(\alpha)$, $\beta, \gamma \in CL(\alpha)$.

Formula Automaton for LTL

Now, define the set of states S_α as follows. $B \subseteq CL(\alpha)$ is a state in S_α if following conditions hold:

For every $\beta \in CL(\alpha)$, either $\beta \in B$ or $\neg\beta \in B$ but never both.

For every $(\beta \vee \gamma) \in CL(\alpha)$, $(\beta \vee \gamma) \in B$ iff either $\beta \in B$ or $\gamma \in B$.

For every $(\beta \wedge \gamma) \in CL(\alpha)$, $(\beta \wedge \gamma) \in B$ iff both $\beta \in B$ and $\gamma \in B$.

For every $F\beta \in CL(\alpha)$, $F\beta \in B$ iff $XF\beta \in B$ or $\beta \in B$

For every $G\beta \in CL(\alpha)$, $G\beta \in B$ iff $XG\beta \in B$ and $\beta \in B$

Formula Automaton for LTL

The initial states of the formula automaton are defined as follows:

$$I_\alpha = \{B \in S_\alpha \mid \alpha \in B\}.$$

The good states of the formula automaton are defined as follows:

$$G_\alpha = \{B \in S_\alpha \mid \text{ for every } F\beta \in CL(\alpha), F\beta \notin B \text{ or both } F\beta, \beta \in B\}.$$

Formula Automaton for LTL

The transition relation $T_\alpha = S_\alpha \times \Sigma_\alpha \times S_\alpha$ is defined as follows.

Let $P' \in \Sigma_\alpha$.

For any $B_1, B_2 \in S_\alpha$, $(B_1, P', B_2) \in T_\alpha$ (denoted by $B_1 \xrightarrow{P'} B_2$) if $P' \cap B_1$ and the following condition holds:

for every $X\beta \in CL(\alpha), X\beta \in B_1$ iff $\beta \in B_2$.

Formula Automata : Example

Formula Automaton for $\alpha = p$

$$CL(\alpha) = \{p, \neg p\}$$
$$S = \{s = \{p\}, t = \{\neg p\}\}$$
$$I = \{s\}$$
$$G = \{s, t\}$$

Figure: Formula automaton of p

Formula Automata : Example

Formula Automaton for $\alpha = p \vee q$

$$CL(\alpha) = \{p \vee q, \neg(p \vee q), p, q, \neg p, \neg q\}$$

$$S = \{s1 = \{p \vee q, p, q\}, s2 = \{p \vee q, p, \neg q\}, s3 = \{p \vee q, \neg p, q\}, s4 = \{\neg(p \vee q), \neg p, \neg q\}\}$$

$$I = \{s1, s2, s3\}$$

$$G = \{s1, s2, s3, s4\}$$

Figure: Formula automaton of $p \vee q$

Formula Automata : Example

Formula Automaton for $\alpha = p \wedge q$

$$CL(\alpha) = \{p \wedge q, \neg(p \wedge q), p, q, \neg p, \neg q\}$$

$$S = \{s1 = \{p \wedge q, p, q\}, s2 = \{\neg(p \wedge q), p, \neg q\}, s3 = \{\neg(p \wedge q), \neg p, q\}, s4 = \{\neg(p \wedge q), \neg p, \neg q\}\}$$

$$I = \{s1\}$$

$$G = \{s1, s2, s3, s4\}$$

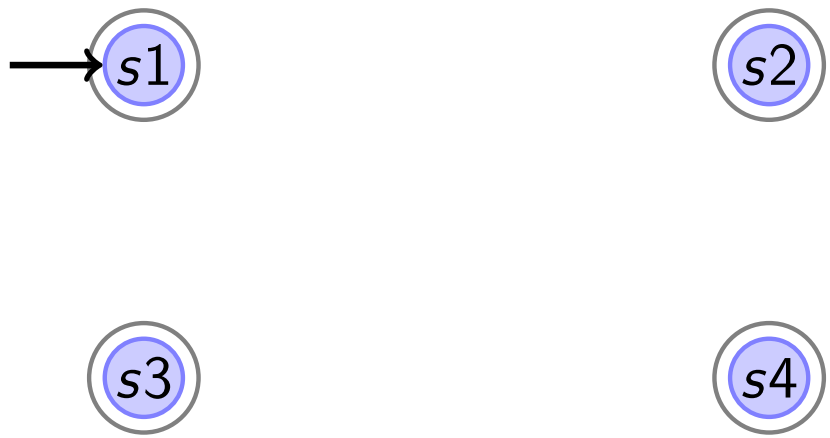

Figure: Formula automaton of $p \wedge q$

Formula Automata : Example

Formula Automaton for $\alpha = Xp$

$CL(\alpha) = \{Xp, \neg Xp, p, \neg p\}$

$S = \{s1 = \{Xp, p\}, s2 = \{Xp, \neg p\}, s3 = \{\neg Xp, p\}, s4 = \{\neg Xp, \neg p\}\}$

$I = \{s1, s2\}$

$G = \{s1, s2, s3, s4\}$

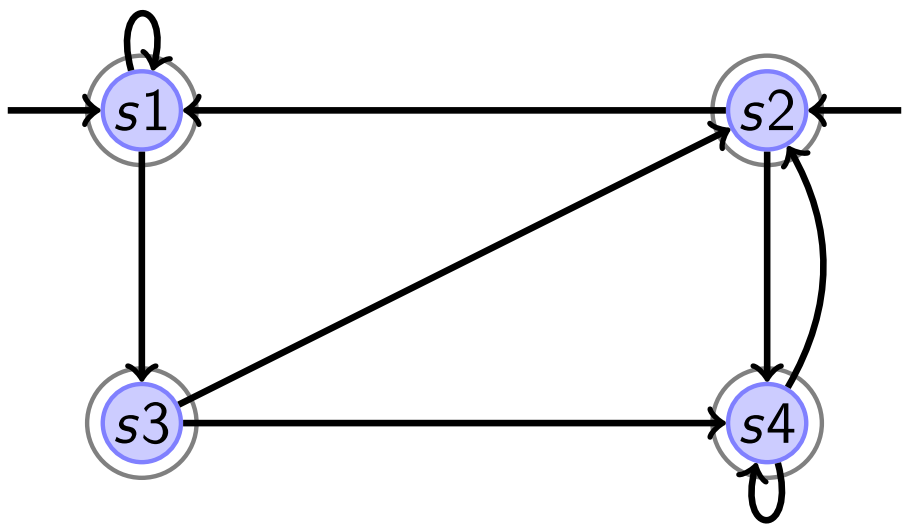

Figure: Formula automaton of Xp

Formula Automata : Example

Formula Automaton for Fp

$$CL(\alpha) = \{Fp, \neg Fp, XFp, \neg XFp, p, \neg p\}$$
$$S = \{s1 = \{Fp, XFp, p\}, s2 = \{Fp, XFp, \neg p\}, s3 = \{Fp, \neg XFp, p\}, s4 = \{\neg Fp, \neg XFp, \neg p\}\}$$
$$I = \{s1, s2, s3\}$$
$$G = \{s1, s3, s4\}$$

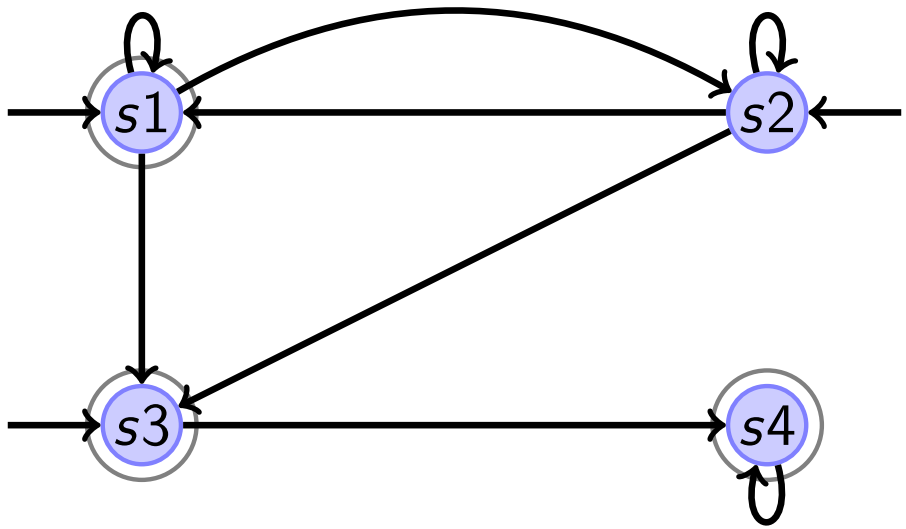

Figure: Formula automaton of Fp

Formula Automata : Example

Formula Automaton for Gp

$$CL(\alpha) = \{Gp, \neg Gp, XGp, \neg XGp, p, \neg p\}$$
$$S = \{s1 = \{\neg Gp, \neg XGp, \neg p\}, s2 = \{\neg Gp, \neg XGp, p\}, s3 = \{\neg Gp, XGp, \neg p\}, s4 = \{Gp, XGp, p\}\}$$
$$I = \{s4\}$$
$$G = \{s1, s3, s4\}$$

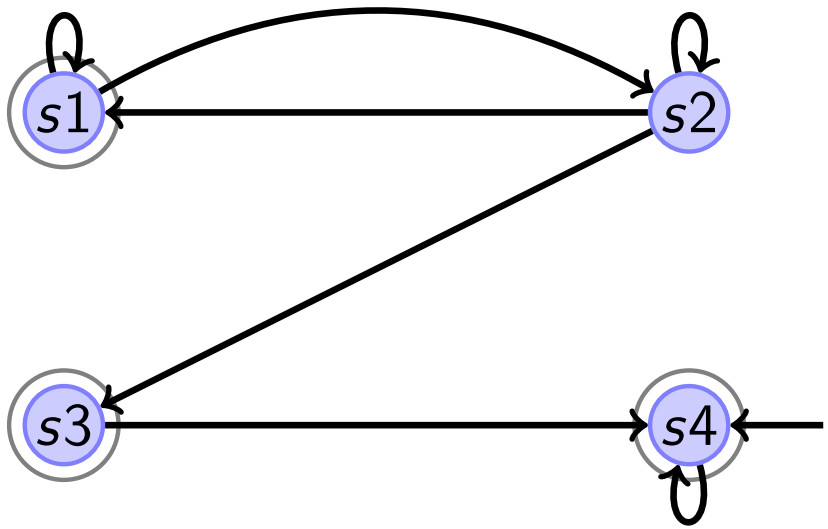

Figure: Formula automaton of Gp

Formula Automata : Exercises

Compute the formula automata for

1. XFp
2. XGp
3. FXp
4. GFp
5. XXp
6. FFp
7. GGp
8. FGp
9. GFp
10. $Fp \vee Fq$
11. $Gp \wedge Gq$
12. $F(p \rightarrow Gq)$
13. $G(p \rightarrow Fq)$
14. $F(p \rightarrow Xq)$
15. $G(p \rightarrow Xq)$

Formula Automaton for LTL formula α

First, construct the (Fischer-Ladner) closure of α, $CL(\alpha)$, as the smallest set containing α and satisfying the following conditions:

For every $F\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $G\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $X\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $\beta \in CL(\alpha)$, $\neg\beta \in CL(\alpha)$.

For every $\neg\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $F\beta \in CL(\alpha)$, $XF\beta \in CL(\alpha)$.

For every $G\beta \in CL(\alpha)$, $XG\beta \in CL(\alpha)$.

For every $\beta \vee \gamma \in CL(\alpha)$, $\beta, \gamma \in CL(\alpha)$.

For every $\beta \wedge \gamma \in CL(\alpha)$, $\beta, \gamma \in CL(\alpha)$.

Formula Automaton for LTL formula α

Now, define the set of states S_α as follows. $B \subseteq CL(\alpha)$ is a state in S_α if following conditions hold:

For every $\beta \in CL(\alpha)$, either $\beta \in B$ or $\neg\beta \in B$ but never both.

For every $(\beta \vee \gamma) \in CL(\alpha)$, $(\beta \vee \gamma) \in B$ iff either $\beta \in B$ or $\gamma \in B$.

For every $(\beta \wedge \gamma) \in CL(\alpha)$, $(\beta \wedge \gamma) \in B$ iff both $\beta \in B$ and $\gamma \in B$.

For every $F\beta \in CL(\alpha)$, $F\beta \in B$ iff $XF\beta \in B$ or $\beta \in B$

For every $G\beta \in CL(\alpha)$, $G\beta \in B$ iff $XG\beta \in B$ and $\beta \in B$

Formula Automaton for LTL formula α

The initial states of the formula automaton are defined as follows:

$$I_\alpha = \{B \in S_\alpha \mid \alpha \in B\}.$$

The good states of the formula automaton are defined as follows:

$$G_\alpha = \{B \in S_\alpha \mid \text{ for every } F\beta \in CL(\alpha), F\beta \notin B \text{ or both } F\beta, \beta \in B\}.$$

Formula Automaton for LTL formula α

The transition relation $T_\alpha = S_\alpha \times \Sigma_\alpha \times S_\alpha$ is defined as follows.

Let $P' \in \Sigma_\alpha$.

For any $B_1, B_2 \in S_\alpha$, $(B_1, P', B_2) \in T_\alpha$ (denoted by $B_1 \xrightarrow{P'} B_2$) if $P' \cap B_1$ and the following condition holds:

for every $X\beta \in CL(\alpha), X\beta \in B_1$ iff $\beta \in B_2$.

Correctness of Formula Automaton Construction

In order to prove the correctness of formula automaton (A_α) construction for a given LTL formula α, we have to prove the following proposition:

Theorem

$$L(A_\alpha) = Mod(\alpha)$$

- The proof has two parts.
- In the first, given an accepting run ρ of A_α, we construct a model M of α ($M, 0 \models \alpha$).
- Conversely, given a model M of α ($M, 0 \models \alpha$) we construct an accepting run ρ of A_α.
- Thus, we show that the two sets $L(A_\alpha)$ and $Mod(\alpha)$ are same.

Proof of Correctness

$\Rightarrow$:) Let $\rho = B_0 \xrightarrow{P_0} B_1 \xrightarrow{P_1} \cdots \xrightarrow{P_{i-1}} B_i \xrightarrow{P_i} \cdots$ be an accepting run of A_α.

- B_0 is an initial state, i.e, $\alpha \in B_0$.
- For every $i \in \mathbb{N}_0$, $P_i = B_i \cap P$ and for every $X\beta \in CL(\alpha)$, $X\beta \in B_i$ iff $\beta \in B_{i+1}$.
- $\exists^\infty i \in \mathbb{N}_0$, B_i is a good state.

We have to show that $M = P_0 P_1 P_2 \cdots$ is a model of α. That is, $M, 0 \models \alpha$.

Lemma

$$\forall i \in \mathbb{N}_0, \forall \beta \in CL(\alpha),\ \beta \in B_i \text{ iff } M, i \models \beta.$$

Assuming the lemma, as $\alpha \in B_0$, the Lemma gives us $M, 0 \models \alpha$ and we are done.

Proof of Truth Lemma

By induction on the structure of β.

$\beta = p$:) $p \in B_i$ iff $p \in P_i$ — by the definition of P_i

iff $M, i \models p$ — by the definition of $\models$

$\beta = \neg\gamma$:) $\neg\gamma \in B_i$ iff $\gamma \notin B_i$ — by the definition of B's

iff $M, i \not\models \gamma$ — by Inductive Hypothesis

iff $M, i \models \neg\gamma$ — by the definition of $\models$

$\beta = \gamma \vee \delta$:) $\gamma \vee \delta \in B_i$ iff $\gamma \in B_i$ or $\delta \in B_i$ — by the definition of B's

iff $M, i \models \gamma$ or $M, i \models \delta$ — by Inductive Hypothesis

iff $M, i \models \gamma \vee \delta$ — by the definition of $\models$

$\beta = \gamma \wedge \delta$:) $\gamma \wedge \delta \in B_i$ iff $\gamma \in B_i$ and $\delta \in B_i$ — by the definition of B's

iff $M, i \models \gamma$ and $M, i \models \delta$ — by Inductive Hypothesis

iff $M, i \models \gamma \wedge \delta$ — by the definition of $\models$

Proof of Truth Lemma

$\beta = X\gamma$:) $X\gamma \in B_i$ iff $\gamma \in B_{i+1}$ by the definition of $\rightarrow$

iff $M, i+1 \models \gamma$ by Inductive Hypothesis

iff $M, i \models X\gamma$ by the definition of $\models$

$\beta = G\gamma$:) Given $G\gamma \in B_i$, we can argue that $\forall j \geq i$, $\gamma \in B_j$. By inductive hypothesis, $\forall j \geq i$, $M, j \models \gamma$. By definition of $\models$, $M, i \models G\gamma$.

Conversely, let $M, i \models G\gamma$. By the definition of $\models$, $\forall j \geq i$, $M, j \models \gamma$, $M, j \models G\gamma$ and $M, j \models XG\gamma$. By inductive hypothesis, $\forall j \geq i$, $\gamma \in B_j$ and $XG\gamma \in B_j$. By the definition of B's, $\forall j \geq i$, $G\gamma \in B_j$. In particular, $G\gamma \in B_i$ and we are done.

$\beta = F\gamma$:) Given $F\gamma \in B_i$, we can argue that $\exists j \geq i$, $\gamma \in B_j$. Otherwise ρ may not be an accepting run. Let k be the smallest such j. That is, $\gamma \in B_k$. By inductive hypothesis, $M, k \models \gamma$. By definition of $\models$, $M, i \models F\gamma$.

Conversely, let $M, i \models F\gamma$. By definition of $\models$, $\exists j \geq i$ $M, j \models \gamma$. Let k be the smallest such j. By inductive hypothesis, $\gamma \in B_k$. Now, using the definitions of B's and $\rightarrow$, we can argue that $\forall k' : i \leq k' \leq k$, $F\gamma \in B_{k'}$. So, in particular $F\gamma \in B_i$ and we are done.

Proof of Correctness

$\Leftarrow$:) Let $M = P_0 P_1 P_2 \cdots$ be a model of α. We constuct a run of A_α accepting M as follows. Let $\rho = B_0 B_1 B_2 \cdots$ where for any $i \in \mathbb{N}_0$,

$$B_i = \{\beta \in CL(\alpha) \mid M, i \models \beta\}.$$

We need to verify that ρ is a valid and accepting run of A_α over M.

- For every $i \in \mathbb{N}_0$, B_i is a state of A_α.
- B_0 is an initial state of A_α, i.e, $\alpha \in B_0$.
- For every $i \in \mathbb{N}_0$, $P_i = B_i \cap P$ and for every $X\beta \in CL(\alpha)$, $X\beta \in B_i$ iff $\beta \in B_{i+1}$.
- $\exists^\infty i \in \mathbb{N}_0$, B_i is a good state.

Modified Formula Automaton for LTL formula α

As usual, construct the (Fischer-Ladner) closure of α, $CL(\alpha)$, as the smallest set containing α and satisfying the following conditions:

For every $F\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $G\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $X\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $\beta \in CL(\alpha)$, $\neg\beta \in CL(\alpha)$.

For every $\neg\beta \in CL(\alpha)$, $\beta \in CL(\alpha)$.

For every $F\beta \in CL(\alpha)$, $XF\beta \in CL(\alpha)$.

For every $G\beta \in CL(\alpha)$, $XG\beta \in CL(\alpha)$.

For every $\beta \vee \gamma \in CL(\alpha)$, $\beta, \gamma \in CL(\alpha)$.

For every $\beta \wedge \gamma \in CL(\alpha)$, $\beta, \gamma \in CL(\alpha)$.

Modified Formula Automaton for LTL formula α

Now, define the set of atoms AT_α as follows. $B \subseteq CL(\alpha)$ is an atom in AT_α if following conditions hold:

For every $\beta \in CL(\alpha)$, either $\beta \in B$ or $\neg\beta \in B$ but never both.

For every $(\beta \vee \gamma) \in CL(\alpha)$, $(\beta \vee \gamma) \in B$ iff either $\beta \in B$ or $\gamma \in B$.

For every $(\beta \wedge \gamma) \in CL(\alpha)$, $(\beta \wedge \gamma) \in B$ iff both $\beta \in B$ and $\gamma \in B$.

For every $F\beta \in CL(\alpha)$, $F\beta \in B$ iff $XF\beta \in B$ or $\beta \in B$

For every $G\beta \in CL(\alpha)$, $G\beta \in B$ iff $XG\beta \in B$ and $\beta \in B$

Modified Formula Automaton for LTL formula α

- Define the set of F-Requirements $FR_\alpha = \{F\beta \mid F\beta \in CL(\alpha)\}$.
- Define the set of states of the formula automaton as follows:

$$S_\alpha = \{(B, u) \mid B \in AT_\alpha \text{ and } u \subseteq FR_\alpha\}$$

- The initial states of the formula automaton are defined as follows:

$$I_\alpha = \{(B, u) \in S_\alpha \mid \alpha \in B \text{ and } u = \emptyset\}.$$

- The good states of the formula automaton are defined as follows:

$$G_\alpha = \{(B, u) \in S_\alpha \mid u = \emptyset\}.$$

Modified Formula Automaton for LTL formula α

The transition relation $T_\alpha = S_\alpha \times S_\alpha$ is defined as follows.

For any $(B_1, u_1), (B_2, u_2) \in S_\alpha$, $(B_1, u_1) \xrightarrow{P'} (B_2, u_2)$ if $P' = P \cap B_1$ and the following conditions holds:

- for every $X\beta \in CL(\alpha), X\beta \in B_1$ iff $\beta \in B_2$.
- If $u_1 = \emptyset$ then $u_2 = \{F\beta \in B_2 \mid \beta \notin B_2\}$ else if $u_1 \neq \emptyset$ then $u_2 = \{F\beta \in u_1 \mid \beta \notin B_2\}$

Correctness of Modified Formula Automata Construction

We have to argue the $\beta = F\gamma$ case in Truth Lemma afresh. Also, we have to verify the correctness of automaton constructed in the $\Leftarrow$ part of the proof. Rest of the proof remains unchanged.

LTL with Past Modalities – LTL(Past)

Let us add two more temporal modalities to our logic: H and J with the following semantics:

$$M, i \models H\alpha \text{ iff } \exists j \leq i,\ M, j \models \alpha$$

$$M, i \models J\alpha \text{ iff } \forall j \leq i,\ M, j \models \alpha$$

H is the analogue of F and J is the analogue of G in the past.

How do we modify the formula automaton to reason with H and J?

Model Checking against LTL Specs

- Given a system description $A = (S, I, T, V, G)$ and an LTL formula α, Does every accepting run of A conform to α, denoted by $A \models^{?} \alpha$.
- Let $V : S \rightarrow 2^P$ be the assignment of propositions to states of A.
- A can be model checked against α as follows:

  Construct formula automaton A_α for α such that $L(A_\alpha)$ contains all satisfying models of α and nothing else.
  Now, $A \models \alpha$ if for every accepting run ρ, $\rho, 0 \models \alpha$.
  That is, $L(A) \subseteq L(A_\alpha)$.
  Which is equivalent to checking whether
  $L(A) \cap L(A_{\neg\alpha}) = \emptyset$.