

Algoritmos de verificación para LTL y CTL*

Marcelo Arenas

Vamos a mostrar un algoritmo de verificación para LTL que usa autómatas sobre palabras infinitas.

- ▶ Vamos a utilizar algunas de las técnicas desarrolladas en el capítulo anterior.

Para empezar, tenemos que conectar los sistemas de transición con los autómatas.

Un sistema de transición sobre un alfabeto Σ es una estructura $\mathcal{M} = (E, (P_a)_{a \in \Sigma}, R)$, donde:

- ▶ E es un conjunto finito de estados;
- ▶ $P_a \subseteq E$ para cada $a \in \Sigma$;
- ▶ R es una relación binaria en E tal que para cada $e \in E$ existe $e' \in E$ tal que $(e, e') \in R$.

LTL sobre sistemas de transición: Para recordar

LTL sobre sistemas de transición:

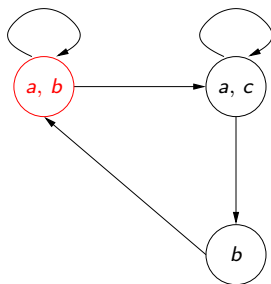
$$\psi, \psi' := a \ (a \in \Sigma) \mid \neg\psi \mid \psi \vee \psi' \mid \mathbf{X}\psi \mid \psi \mathbf{U}\psi'.$$

Problema de verificación: Dada una fórmula φ en LTL, un sistema de transición $\mathcal{M}$ y un estado e en $\mathcal{M}$: **Verificar si** $(\mathcal{M}, e) \models \mathbf{A}\varphi$.

- Para cada camino $\pi = e e_1 e_2 \cdots$ sobre $\mathcal{M}$ se tiene que $(\mathcal{M}, \pi) \models \varphi$.

LTL sobre sistemas de transición: Ejemplos

Sistema de transición sobre el alfabeto $\Sigma = \{a, b, c\}$:



¿Cuáles de las siguientes fórmulas son ciertas en el estado de color rojo?

$A(aU b)$, $A(aU c)$, $A(aU(b \wedge Xb))$.

Sistemas de transición como autómatas

Dado: Sistema de transición $\mathcal{M} = (E, (P_a)_{a \in \Sigma}, R)$ y $e \in E$:

- Notación: $P_{\mathcal{M}}(e) = \{a \in \Sigma \mid e \in P_a\}$.

En un autómata hay un símbolo en cada arco, mientras que en un sistema de transición pueden haber varios símbolos en un estado.

- Para resolver este problema usamos 2^Σ como el alfabeto del autómata.

Sistemas de transición como autómatas

Dado: Sistema de transición $\mathcal{M} = (E, (P_a)_{a \in \Sigma}, R)$ y $e \in E$:

- Notación: $P_{\mathcal{M}}(e) = \{a \in \Sigma \mid e \in P_a\}$.

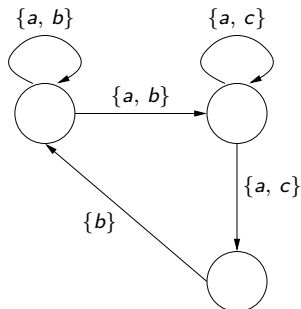
En un autómata hay un símbolo en cada arco, mientras que en un sistema de transición pueden haber varios símbolos en un estado.

- Para resolver este problema usamos 2^Σ como el alfabeto del autómata.

Definimos el autómata de Büchi $\mathcal{A}_{\mathcal{M}}$ que codifica a $\mathcal{M}$ como $(E, 2^\Sigma, E, \delta, E)$, donde: $\delta(e, P_{\mathcal{M}}(e)) = \{e' \mid (e, e') \in R\}$.

Sistemas de transición como autómatas: Ejemplo

Para el sistema de transición $\mathcal{M}$ mostrado anteriormente, se tiene que $\mathcal{A}_{\mathcal{M}}$ es:



Sistemas de transición como autómatas: Estados iniciales

Queremos verificar si una fórmula $\mathbf{A}\varphi$ es verdadera en un estado particular del sistema de transición.

- ▶ Usamos el estado inicial del autómata para indicar cuál es el estado desde el cual se quiere hacer la verificación.

Dado: Sistema de transición $\mathcal{M} = (E, (P_a)_{a \in \Sigma}, R)$ y estado $e_0 \in E$.

Sistemas de transición como autómatas: Estados iniciales

Queremos verificar si una fórmula $\mathbf{A}\varphi$ es verdadera en un estado particular del sistema de transición.

- Usamos el estado inicial del autómata para indicar cuál es el estado desde el cual se quiere hacer la verificación.

Dado: Sistema de transición $\mathcal{M} = (E, (P_a)_{a \in \Sigma}, R)$ y estado $e_0 \in E$.

$\mathcal{A}_{\mathcal{M}, e_0}$: Autómata de Büchi $(E, 2^\Sigma, \{e_0\}, \delta, E)$ tal que $\delta(e, P_{\mathcal{M}}(e)) = \{e' \mid (e, e') \in R\}$.

Vardi & Wolper para sistemas de transición

En la verificación de fórmulas LTL es fundamental la construcción de Vardi & Wolper.

Un pequeño problema: Dado un alfabeto Σ , ahora consideramos autómata con alfabeto 2^Σ .

- ▶ Tenemos que hacer algunas modificaciones a la construcción de Vardi & Wolper.

Dado: Fórmula φ en LTL sobre un alfabeto Σ .

Notación: Dado un camino $\pi = e_0 e_1 e_2 \dots$ sobre un sistema de transición $\mathcal{M}$, $P_{\mathcal{M}}(\pi) = P_{\mathcal{M}}(e_0)P_{\mathcal{M}}(e_1)P_{\mathcal{M}}(e_2)\dots$.

Ahora queremos construir un autómata de Büchi $\mathcal{A}_{\varphi}$ tal que para todo sistema de transición $\mathcal{M}$ y camino π sobre $\mathcal{M}$:

$$(\mathcal{M}, \pi) \models \varphi \quad \text{si y sólo si} \quad P_{\mathcal{M}}(\pi) \in L_{\omega}(\mathcal{A}_{\varphi}).$$

Vardi & Wolper para sistemas de transición: Cambios

Dado: Fórmula φ en LTL sobre un alfabeto Σ .

Primero tenemos que cambiar la definición de las funciones de etiquetado:

- Una función $\xi : \mathbb{N} \rightarrow 2^{\text{cl}(\varphi)}$ es una función de etiquetado para φ sobre un camino $\pi = \sigma_0\sigma_1\sigma_2\cdots$, donde $\sigma_i \subseteq \Sigma$ para cada $i \geq 0$, si:

Vardi & Wolper para sistemas de transición: Cambios

Dado: Fórmula φ en LTL sobre un alfabeto Σ .

Primero tenemos que cambiar la definición de las funciones de etiquetado:

- ▶ Una función $\xi : \mathbb{N} \rightarrow 2^{\text{cl}(\varphi)}$ es una función de etiquetado para φ sobre un camino $\pi = \sigma_0\sigma_1\sigma_2\cdots$, donde $\sigma_i \subseteq \Sigma$ para cada $i \geq 0$, si:
 - ▶ Para cada $a \in \Sigma$: Si $a \in \xi(i)$, entonces $a \in \sigma_i$, y si $\neg a \in \xi(i)$, entonces $a \notin \sigma_i$.

Vardi & Wolper para sistemas de transición: Cambios

Dado: Fórmula φ en LTL sobre un alfabeto Σ .

Primero tenemos que cambiar la definición de las funciones de etiquetado:

- ▶ Una función $\xi : \mathbb{N} \rightarrow 2^{\text{cl}(\varphi)}$ es una función de etiquetado para φ sobre un camino $\pi = \sigma_0 \sigma_1 \sigma_2 \cdots$, donde $\sigma_i \subseteq \Sigma$ para cada $i \geq 0$, si:
 - ▶ Para cada $a \in \Sigma$: Si $a \in \xi(i)$, entonces $a \in \sigma_i$, y si $\neg a \in \xi(i)$, entonces $a \notin \sigma_i$.
 - ▶ Se cumplen las reglas 2-7 para la definición original de funciones de etiquetado.

Vardi & Wolper para sistemas de transición: Cambios

Hay que hacer los siguientes cambios en la definición del autómata de Büchi generalizado $\mathcal{A}_\varphi = (Q, 2^\Sigma, Q_0, \delta, \mathcal{G})$.

1. Q está formado por todos los subconjuntos X de $\text{cl}(\varphi)$ tales que:
 - ▶ para todo $a \in \Sigma$: Si $a \in X$, entonces $\neg a \notin X$;
 - ▶ si $\psi \vee \theta \in X$, entonces $\psi \in X$ o $\theta \in X$;
 - ▶ si $\psi \wedge \theta \in X$, entonces $\psi \in X$ y $\theta \in X$.

3. Dados $X, Y \in Q$ y $\sigma \subseteq \Sigma$, se tiene que $Y \in \delta(X, \sigma)$ si:

- ▶ si $a \in X$, entonces $a \in \sigma$;
- ▶ si $\neg a \in X$, entonces $a \notin \sigma$;
- ▶ si $\mathbf{X}\psi \in X$, entonces $\psi \in Y$;
- ▶ si $\psi \mathbf{U}\theta \in X$, entonces $\theta \in X$ o $(\psi \in X \text{ y } \psi \mathbf{U}\theta \in Y)$;
- ▶ si $\psi \mathbf{R}\theta \in X$, entonces $\theta \in X$ y $(\psi \in X \text{ o } \psi \mathbf{R}\theta \in Y)$.

Algoritmo de verificación para LTL

¡Finalmente tenemos todos los ingredientes para hacer verificación de fórmulas LTL!

Dado: Sistema de transición $\mathcal{M}$, estado e en $\mathcal{M}$ y fórmula φ en LTL.

► Por resolver: $\mathcal{L}(\mathcal{M}, e) \models \mathbf{A}\varphi$?

Algoritmo de verificación para LTL

El siguiente algoritmo verifica si $(\mathcal{M}, e) \models \mathbf{A}\varphi$.

Algoritmo de verificación para LTL

El siguiente algoritmo verifica si $(\mathcal{M}, e) \models \mathbf{A}\varphi$.

1. Construya el autómata $\mathcal{A}_{\mathcal{M},e}$;

Algoritmo de verificación para LTL

El siguiente algoritmo verifica si $(\mathcal{M}, e) \models \mathbf{A}\varphi$.

1. Construya el autómata $\mathcal{A}_{\mathcal{M},e}$;
2. Construya el autómata $\mathcal{A}_{\neg\varphi}$;

Algoritmo de verificación para LTL

El siguiente algoritmo verifica si $(\mathcal{M}, e) \models \mathbf{A}\varphi$.

1. Construya el autómata $\mathcal{A}_{\mathcal{M},e}$;
2. Construya el autómata $\mathcal{A}_{\neg\varphi}$;
3. Construya el autómata $\mathcal{A}_{\mathcal{M},e} \times \mathcal{A}_{\neg\varphi}$ que acepta el lenguaje $L_w(\mathcal{A}_{\mathcal{M},e}) \cap L_w(\mathcal{A}_{\neg\varphi})$;

Algoritmo de verificación para LTL

El siguiente algoritmo verifica si $(\mathcal{M}, e) \models \mathbf{A}\varphi$.

1. Construya el autómata $\mathcal{A}_{\mathcal{M},e}$;
2. Construya el autómata $\mathcal{A}_{\neg\varphi}$;
3. Construya el autómata $\mathcal{A}_{\mathcal{M},e} \times \mathcal{A}_{\neg\varphi}$ que acepta el lenguaje $L_w(\mathcal{A}_{\mathcal{M},e}) \cap L_w(\mathcal{A}_{\neg\varphi})$;
4. Verifique si $L_w(\mathcal{A}_{\mathcal{M},e} \times \mathcal{A}_{\neg\varphi}) = \emptyset$
 - Si es así retorne **sí**, y en caso contrario retorne **no**.