

Introducción a las Lógicas Temporales

IIC3800

Pablo Barceló

Model checking

Model checking: Técnica que permite verificar propiedades de sistemas concurrentes en forma **automática**. Especialmente utilizado en el diseño de circuitos digitales y protocolos de comunicación.

Realiza una búsqueda exhaustiva en el conjunto de estados para ver si una especificación se cumple o no.

El model checking consta de tres partes: **modelamiento**, **especificación**, y **verificación**.

Lógicas temporales: Permiten especificar propiedades dinámicas de un sistema sin introducir el tiempo explícitamente.

La idea es describir secuencias de transiciones entre estados en un sistema que evoluciona en el tiempo.

Ejemplo: “Eventualmente” ocurrirá P , “nunca” ocurrirá P , etc.

Estos “eventualmente”, “nunca”, etc., son los operadores de los lógicas.

Breve historia de las lógicas temporales:

- ▶ Inicialmente propuestas en los 50s para investigaciones filosóficas.
- ▶ Pnueli (1977) fue el primero en proponerlas para verificación. El sistema se especificaba como un sistema de axiomas, y se probaban propiedades a partir de estos axiomas.
- ▶ Clarke y Emerson (1981) automatizaron este proceso diseñando algoritmos de verificación eficientes para ciertas lógicas temporales.
- ▶ Luego comenzó a estudiarse la complejidad de verificación de estas lógicas (model-checking, parametrized complexity, etc).

Sistemas de transición

Las lógicas temporales permiten hablar sobre las transiciones entre estados (es decir, de las **secuencias de computación** realizadas por el sistema). Los sistemas deben ser modelados entonces como **sistemas de transición**.

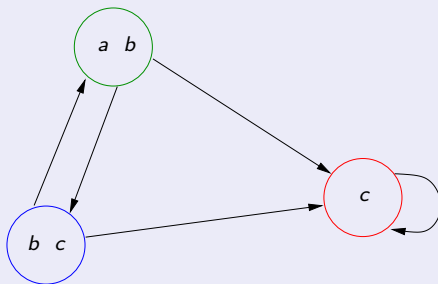
Cuando hablemos de lógicas temporales:

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

- ▶ E es un conjunto **finito** de elementos (estados),
- ▶ cada P_a es un predicado en E , i.e. $P_a \subseteq E$, y
- ▶ R es una relación binaria en E (llamada **relación de transición**), tal que para cada $e \in E$ existe $e' \in E$ tal que $R(e, e')$.

Ejemplo de sistema de transición

Ejemplo: $\mathcal{M}_E$ es un sistema de transición sobre alfabeto $\{a, b, c\}$.



Intuición sobre sistemas de transición

El alfabeto representa el conjunto de descripciones atómicas que son relevantes para el sistema. Por ejemplo:

- La impresora no tiene papel, la alarma se ha encendido, etc.

Estas **no** hacen referencia a la dinámica del sistema.

Cada estado e representa una descripción **instantánea** del comportamiento del sistema. Si $e \in P_a$ esto quiere decir que la descripción atómica a es cierta en ese instante.

Una **transición** $R(e, e')$ representa un cambio en el estado del sistema. Una **computación** de un sistema es una secuencia infinita de sus estados, donde cada estado se obtiene desde el anterior por medio de una transición.

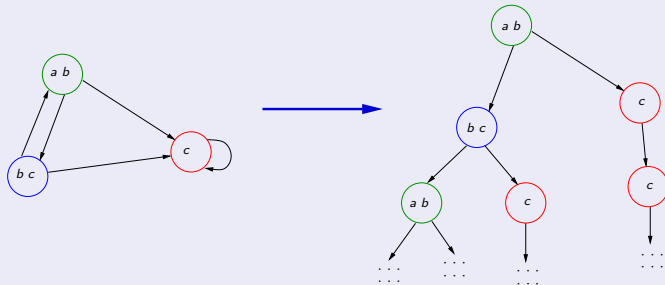
La lógica temporal CTL*

La primera lógica temporal que veremos es **CTL***. Esta especifica propiedades del **árbol de computación** de un sistema de transición.

Este árbol (infinito) se obtiene designando a un estado del sistema como la raíz (estado inicial), y muestra todas las posibles computaciones desde ese estado.

La lógica temporal CTL*

Por ejemplo, el árbol de transición para $\mathcal{M}_E$ designando al estado verde es el siguiente:



La lógica temporal CTL*

CTL* está compuesta por **operadores temporales** y **cuantificación sobre caminos**.

La cuantificación sobre caminos está dada por los cuantificadores **A** (para todo camino) y **E** (existe un camino).

Estos cuantificadores son utilizados en un estado particular del sistema de transición, y evaluadas sobre el árbol de computación relacionado con ese estado.

La lógica temporal CTL*

Los operadores temporales sirven para describir propiedades de un camino en el árbol de computación.

Usamos 4 operadores básicos:

- ▶ **X** (próximo estado): unario.
- ▶ **F** (en el futuro): unario.
- ▶ **G** (siempre en el futuro): unario.
- ▶ **U** (algo ocurre hasta que otra cosa ocurre): binario.

La lógica consiste en fórmulas de **estado** (que se evalúan en un estado del sistema) y de **camino** (que se evalúan en una rama del árbol).

La sintaxis de las fórmulas de estado sobre alfabeto Σ es como sigue:

- ▶ Si $a \in \Sigma$, entonces a es una fórmula de estado.
- ▶ Si ϕ, ϕ' son fórmulas de estado, entonces $\neg\phi$, $\phi \wedge \phi'$ y $\phi \vee \phi'$ son fórmulas de estado.
- ▶ Si ψ es una fórmula de **camino**, entonces **E** ψ y **A** ψ son fórmulas de estado.

La sintáxis de las fórmulas de camino sobre alfabeto Σ es como sigue:

- ▶ Toda fórmula de **estado** es también una fórmula de camino.
- ▶ Si ψ, ψ' son fórmulas de camino, entonces $\neg\psi$, $\psi \wedge \psi'$ y $\psi \vee \psi'$ son fórmulas de camino.
- ▶ Si ψ, ψ' son fórmulas de camino, entonces **X** ψ , **F** ψ , **G** ψ , y ψ **U** ψ' son fórmulas de camino.

Ejercicio: ¿Cómo expresaría las siguientes propiedades en CTL*?

- ▶ En cada camino la proposición a ocurre infinitas veces.

$$\mathbf{A} \, (\mathbf{GF}a).$$

- ▶ Es posible acceder a un estado donde la proposición b es cierta, y desde el cual existe un camino en el que la proposición a ocurre finitas veces.

$$\mathbf{EF} \, (b \wedge \mathbf{EFG} \neg a).$$

Para definir la semántica de CTL* debemos definir primero qué es un camino en un sistema de transición.

Sea $\mathcal{M}$ un sistema de transición. Un **camino** π sobre $\mathcal{M}$ es una secuencia $e_0 e_1 \cdots$ tal que para cada $i \geq 0$, $(e_i, e_{i+1}) \in R$.

En otras palabras, el camino $e_0 e_1 \cdots$ es una rama del árbol de computación de $\mathcal{M}$ con raíz e_0 .

Denotamos por π^i el sufijo de π que empieza en e_i .

Dado sistema de transición $\mathcal{M}$ y estado e :

- ▶ $(\mathcal{M}, e) \models a$ si y sólo si $e \in P_a$.
- ▶ $(\mathcal{M}, e) \models \neg\phi$ si y sólo si no es el caso que $(\mathcal{M}, e) \models \phi$.
- ▶ $(\mathcal{M}, e) \models \phi \vee \phi'$ si y sólo si $(\mathcal{M}, e) \models \phi$ o $(\mathcal{M}, e) \models \phi'$.
- ▶ $(\mathcal{M}, e) \models \phi \wedge \phi'$ si y sólo si $(\mathcal{M}, e) \models \phi$ y $(\mathcal{M}, e) \models \phi'$.
- ▶ $(\mathcal{M}, e) \models \mathbf{E}\psi$ si y sólo existe un camino π de la forma $ee_1 \cdots$ tal que $(\mathcal{M}, \pi) \models \psi$.
- ▶ $(\mathcal{M}, e) \models \mathbf{A}\psi$ si y sólo para todo camino π de la forma $ee_1 \cdots$ se tiene que $(\mathcal{M}, \pi) \models \psi$.

Semántica de CTL*: Fórmulas de camino

Dado sistema de transición $\mathcal{M}$ y camino π .

- ▶ Para fórmula de estado ϕ , $(\mathcal{M}, \pi) \models \phi$ si y sólo si e es el primer estado de π y $(\mathcal{M}, e) \models \phi$.
- ▶ $(\mathcal{M}, \pi) \models \mathbf{X}\psi$ si y sólo si $(\mathcal{M}, \pi^1) \models \psi$.
- ▶ $(\mathcal{M}, \pi) \models \mathbf{F}\psi$ si y sólo si existe $j \geq 0$ tal que $(\mathcal{M}, \pi^j) \models \psi$.
- ▶ $(\mathcal{M}, \pi) \models \mathbf{G}\psi$ si y sólo si para todo $j \geq 0$ se tiene que $(\mathcal{M}, \pi^j) \models \psi$.
- ▶ $(\mathcal{M}, \pi) \models \psi \mathbf{U} \psi'$ si y sólo si existe $j \geq 0$ tal que $(\mathcal{M}, \pi^j) \models \psi'$ y para todo $0 \leq k < j$ se tiene que $(\mathcal{M}, \pi^k) \models \psi$.
- ▶ Combinaciones Booleanas de fórmulas de camino evaluadas de forma estándar.

Ejercicio: Evalúe las siguientes consultas en $(\mathcal{M}_E, e)$, donde e es el nodo verde.

- ▶ **EXc , $E(XGc)$, y $A((a \vee b)Uc)$.**
- ▶ **EFc , $E(XG\neg b \wedge GFa)$, y $E\neg F(\neg a \wedge \neg b \wedge c)$.**

Propiedades de CTL*

Ejercicio: Demuestre que los operadores $\forall, \neg, \mathbf{X}, \mathbf{U}, \mathbf{E}$ bastan para expresar cada fórmula en CTL*.

Otros operadores interesantes pueden ser además definidos:

- ¿Qué significa $\neg(\neg\psi \mathbf{U} \neg\psi')$ (esto se escribe comúnmente como $(\psi \mathbf{R} \psi')$)?

¿Qué significa $\mathbf{XF}\psi$? ¿Qué significa $\mathbf{FG}\psi$?

¿Cómo podría definirse un operador $\mathbf{U}_s$ tal que $(\mathcal{M}, \pi) \models \psi \mathbf{U}_s \psi'$ si y sólo si existe $j > 0$ tal que $(\mathcal{M}, \pi^j) \models \psi'$ y para todo $0 \leq k < j$, $(\mathcal{M}, \pi^k) \models \psi$?

Pólizas de seguridad (Safety properties)

Un tipo interesante de propiedades expresables en CTL* son las **pólizas de seguridad**. Estas dicen que “algo malo nunca ocurrirá” en el sistema:

- ▶ La planta nuclear nunca explotará, no ocurrirá una revolución, etc.

Sintácticamente son las propiedades expresadas por fórmulas de la forma **AG β** , donde β es una combinación Booleana de proposiciones atómicas.

Son comúnmente las propiedades más fáciles de verificar en un sistema de verificación.

Pólizas de seguridad (Safety properties)

Ejercicio: Sean $\mathbf{AG}\beta$ y $\mathbf{AG}\beta'$ dos pólizas de seguridad. Demuestre que existe póliza de seguridad $\mathbf{AG}\gamma$ tal que para cada sistema de transición $\mathcal{M}$ y estado e ,

$$(\mathcal{M}, e) \models \mathbf{AG}\gamma \Leftrightarrow (\mathcal{M}, e) \models \mathbf{AG}\beta \wedge \mathbf{AG}\beta'.$$

Ejercicio: ¿Por qué la propiedad $\mathbf{AG}(p \rightarrow \mathbf{AF}q)$ no puede ser expresada como una póliza de seguridad?

Pólizas de seguridad (Safety properties)

Desearíamos que otro tipo de propiedades también fueran pólizas de seguridad. Por ejemplo, “no ocurrirá una revolución a menos que el gobierno sea corrupto”, representado por:

$$\mathbf{A}(\neg(\neg\text{corrupto} \mathbf{U} (\neg\text{corrupto} \wedge \text{revolucion}))).$$

Luego mostraremos como capturar esta clase de propiedades.

CTL: Una restricción de CTL*

CTL es la restricción de CTL* que satisface lo siguiente:

- ▶ Cada fórmula de camino en CTL debe estar precedida por un cuantificador de caminos **A** o **E**.

Esta lógica se estudia porque combina buenas propiedades de expresividad y complejidad.

En otras palabras, las fórmulas en CTL están dadas por la siguiente gramática:

$$\phi, \phi' := a \mid \neg\phi \mid \phi \vee \phi' \mid \mathbf{EX}\phi \mid \mathbf{EG}\phi \mid \mathbf{E}(\phi\mathbf{U}\phi')$$

CTL: Una restricción de CTL*

Defina $\alpha \mathbf{W} \beta$ como $\neg(\neg\beta \mathbf{U}(\neg\alpha \wedge \neg\beta))$. Esta fórmula es equivalente a $\mathbf{G}\alpha \vee (\alpha \mathbf{U}\beta)$.

Ejercicio: Demuestre que las siguientes fórmulas pueden ser expresadas en CTL:

- ▶ $\mathbf{AX}\phi$, $\mathbf{AG}\phi$, $\mathbf{A}(\phi \mathbf{U}\phi')$, $\mathbf{A}(\phi \mathbf{R}\phi')$, y $\mathbf{A}(\phi \mathbf{W}\phi')$.
- ▶ $\mathbf{EF}\phi$, $\mathbf{E}(\phi \mathbf{R}\phi')$, y $\mathbf{E}(\phi \mathbf{W}\phi')$.

Una lógica lineal: LTL

LTL es el conjunto de fórmulas de camino de CTL^* que se contruyen sólo a partir de las proposiciones atómicas.

Es decir, LTL es la clase de fórmulas dadas por la siguiente gramtica:

$$\psi, \psi' := a \mid \neg\psi \mid \psi \vee \psi' \mid \mathbf{X}\psi \mid \psi \mathbf{U} \psi'$$

Fórmulas en LTL se evalúan de la manera natural sobre secuencias $e_0e_1 \cdots$ de elementos en Σ .

Para evaluar fórmula ψ en LTL en un estado e de un sistema de transición $\mathcal{M}$:

$$(\mathcal{M}, e) \models \psi \text{ si y sólo si } (\mathcal{M}, e) \models \mathbf{A}\psi.$$

Pólizas de seguridad fuertes

Ahora podemos definir una clase más general de pólizas de seguridad.

Una póliza de seguridad **fuerte** es una fórmula en LTL de la forma β , donde β satisface lo siguiente:

- Si $\pi = e_0 e_1 \cdots$ es una secuencia de elementos en Σ tal que $\pi \not\models \beta$, entonces existe $i \geq 0$ tal que para cualquier secuencia π' de elementos en Σ de la forma $e_0 e_1 \cdots e_i e'_{i+1} e'_{i+2} \cdots$, se tiene que $\pi' \not\models \beta$.

En otras palabras, una póliza de seguridad fuerte es una fórmula en CTL* cuyo contraejemplo es siempre una secuencia **finita** de estados.

Ejercicio: Demuestre que toda póliza de seguridad es una póliza de seguridad fuerte.

Teorema

*Toda fórmula β en LTL construida usando sólo las proposiciones atómicas y los operadores $\vee$, **X**, **G**, y **W** es una póliza de seguridad fuerte.*

Ejercicio: Demuestre el teorema.

Otro tipo de propiedades expresables en LTL son las **pólizas de vitalidad**. Estas son el complemento de las pólizas de seguridad: Dicen que “algo bueno ocurrirá eventualmente”.

- ▶ **Algún día el tirano morirá, el sistema eventualmente termina, etc.**

Por tanto, estas pólizas sólo pueden ser falsificadas por secuencias **infinitas** de estados.

Ejercicio: Defina formalmente qué es una póliza de vitalidad.

La fórmula $\alpha \equiv (a \wedge \mathbf{XGF}b)$ es una fórmula en LTL que no es una póliza de seguridad fuerte, ni es una póliza de vitalidad.