

Validación y Verificación de Software

Lógica Temporal

Métodos vistos hasta ahora para representar propiedades

Los métodos vistos hasta ahora están orientados a la representación del comportamiento y no a las propiedades mismas.

En particular:

FSP provee una sintaxis para representar el comportamiento de los procesos y

las expresiones ω -regulares proveen un lenguaje para representar el conjunto de trazas de interés.

Métodos vistos hasta ahora para representar propiedades

Ejemplo: para representar la propiedad:

“Siempre que llovió, paró”

utilizando una expresión ω -regular, uno debe pensar en qué forma tienen las trazas que cumplen esa propiedad:

$$(\Sigma^* \text{ llueve } \Sigma^* (\neg \text{ llueve}))^\omega$$

Ya hemos experimentado razonamientos semejantes al escribir propiedades en FSP. (¿Cómo se representarían estas propiedades en FSP?)

Limitaciones de las lógicas usuales

Lógica proposicional: Es incapaz de reflejar el cambio de valor de verdad de las proposiciones según transcurra el tiempo.

Ejemplo: “Siempre que llovió, paro”

1er intento llueve \rightarrow \neg llueve

Esta propiedad sólo es verdadera si no llueve :-(

Limitaciones de las lógicas usuales

2^{do} intento `primero_llueve` \rightarrow `luego_para`

Dos inconvenientes:

Disocia los conceptos de llover y dejar de llover.

¿Cómo diferenciamos que esto ocurre siempre o que esto ocurre sólo una vez?

En otras palabras, la lógica proposicional carece de suficiente expresividad.

Limitaciones de las lógicas usuales

Lógica de primer orden: Contrariamente, la lógica de primer orden es demasiado expresiva.

Por un lado presenta un lenguaje demasiado complejo para representar propiedades temporales. La propiedad “Siempre que llovió, paró” se podría representar como:

$$\forall t \in \text{Tiempo} : \text{llueve}(t) \rightarrow \exists t' \in \text{Tiempo} : (t \leq t') \wedge \neg \text{llueve}(t')$$

donde **Tiempo** pueden ser, por ejemplo, los enteros no negativos o los reales no negativos.

Por otro lado, a diferencia de las lógicas proposicionales, la lógica de primer orden es indecidible y por consiguiente no permite en general el cálculo automático de si una fórmula de primer orden se satisface en un sistema dado.

Debemos encontrar algo intermedio.

Lógicas modales

En principio, la lógica modal estudia la formalización del razonamiento que involucra aserciones con modalidades, tales como “necesariamente” o “posiblemente”.

Uno de los enfoques principales consiste en complementar los operadores “clásicos” con los operadores modales:

- que expresa “necesidad”, y

- ◇ que expresa “posibilidad”.

Lógicas modales

Estos operadores modales tienen otras interpretaciones muy útiles, por ej:

Lógica deóntica:

\Box significa “obligatoriamente”

\Diamond significa “está permitido”

Lógica temporal:

\Box significa “siempre en el futuro”

\Diamond significa “en el algún momento en el futuro”

Lógica Temporal

Las lógicas temporales son variantes de la lógica modal que conciernen al razonamiento sobre la **relación temporal de eventos**.

Existen muchos tipos de lógicas temporales. Sus diferencias radican en el modelo temporal, i.e., en como cada una puede observar el paso del tiempo. Por ejemplo:

- el tiempo transcurrido entre eventos es observable

- el tiempo transcurrido entre eventos no es observable, sólo el orden temporal de los eventos.

- los instantes de tiempo son numerables

Lógica Temporal

los instantes de tiempo constituyen un conjunto denso.

el transcurso temporal esta organizado linealmente (como una sola ejecución)

el transcurso temporal se ramifica (puede observar todas -o alguna de- las posibles ejecuciones a partir de cualquier instante)

Algunas Lógicas Temporales

Las siguientes lógicas temporales son comúnmente utilizadas para especificar las propiedades a verificar por las herramientas de model checking:

LTL tiene como modelo de tiempo un conjunto numerable de instantes organizados linealmente pero no puede observar el tiempo transcurrido.

CTL tiene como modelo de tiempo un conjunto numerable de instantes organizados de forma ramificada y tampoco puede observar el tiempo transcurrido.

TCTL tiene como modelo de tiempo un conjunto numerable o denso de instantes organizados de forma ramificada y puede observar (contar) el tiempo transcurrido entre dos eventos.

Algunas Lógicas Temporales

Las siguientes lógicas temporales son comúnmente utilizadas para especificar las propiedades a verificar por las herramientas de model checking:

Nos concentraremos en el uso de LTL, que es la lógica utilizada por SPIN.

LTL tiene como modelo de tiempo un conjunto numerable de instantes organizados linealmente pero no puede observar el tiempo transcurrido.

CTL tiene como modelo de tiempo un conjunto numerable de instantes organizados de forma ramificada y tampoco puede observar el tiempo transcurrido.

TCTL tiene como modelo de tiempo un conjunto numerable o denso de instantes organizados de forma ramificada y puede observar (contar) el tiempo transcurrido entre dos eventos.

LTL (Lógica Temporal Lineal)

Las modalidades de LTL:

Los siguientes son algunos de los operadores modales que admite la lógica LTL.

$\Box \phi$: “siempre en el futuro sucede ϕ ”

$\Diamond \phi$: “en algún instante futuro sucede ϕ ”

$\bigcirc \phi$: “en el siguiente instante sucede ϕ ”

$\phi \mathbf{U} \psi$: “ ϕ sucede hasta que suceda ψ ”

$\phi \mathbf{WU} \psi$: “ ϕ sucede siempre o hasta que suceda ψ ”

LTL (Lógica Temporal Lineal)

Las modalidades de LTL:

Orden Temporal
de los eventos

Los siguientes son algunos de los operadores modales que admite la lógica LTL.

$\Box \phi$: “siempre en el futuro sucede ϕ ”

$\Diamond \phi$: “en algún instante futuro sucede ϕ ”

$\bigcirc \phi$: “en el siguiente instante sucede ϕ ”

$\phi \mathbf{U} \psi$: “ ϕ sucede hasta que suceda ψ ”

$\phi \mathbf{WU} \psi$: “ ϕ sucede siempre o hasta que suceda ψ ”

Sintaxis de LTL

Sea \mathcal{PA} el conjunto de todas las proposiciones atómicas. Luego

cualquier proposición atómica $p \in \mathcal{PA}$ es una fórmula LTL.

Si ϕ y ψ son fórmulas LTL,

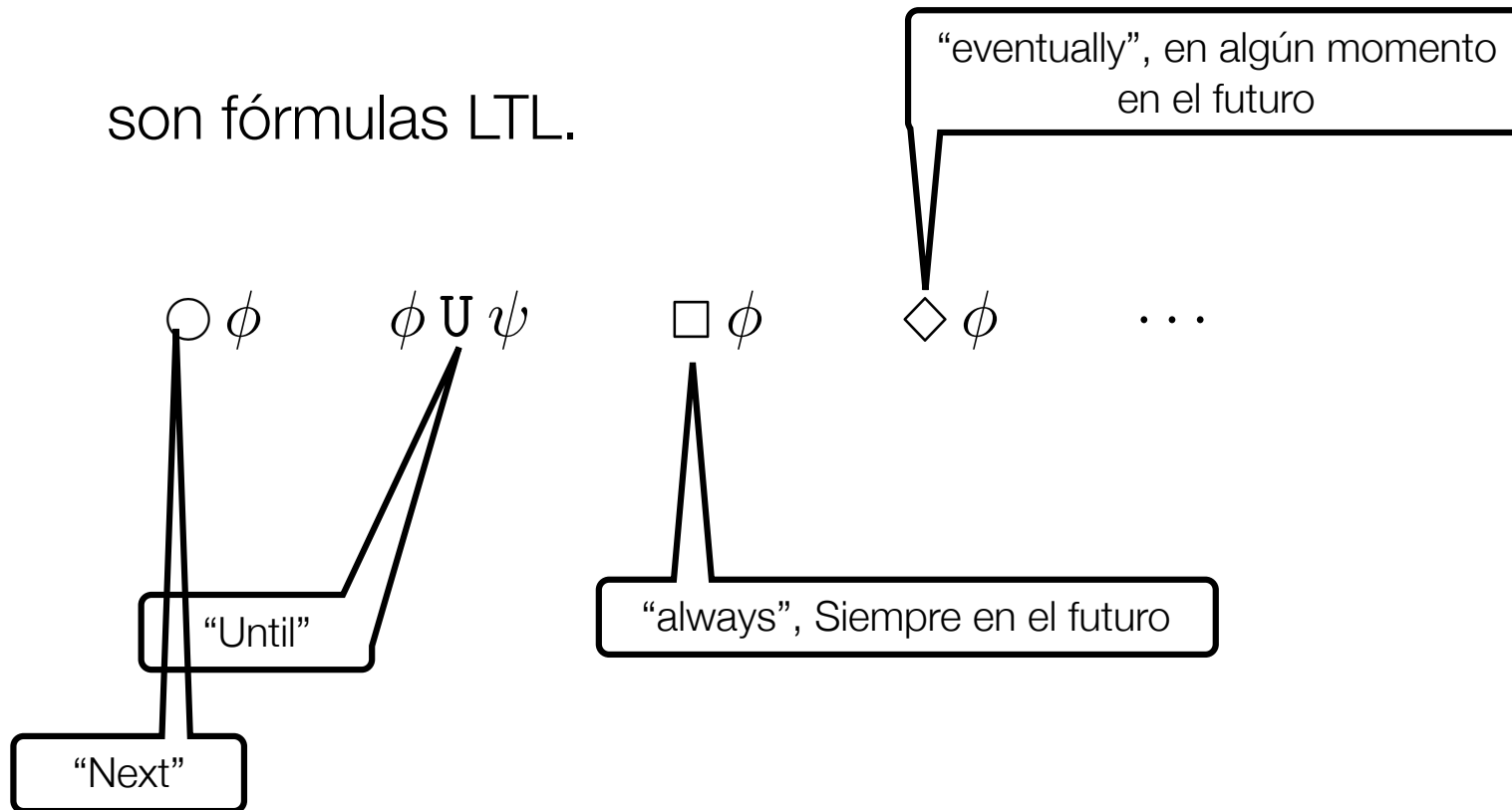
cualquier fórmula construida con los conectivos lógicos
proposicionales

$$\neg\phi \quad \phi \wedge \psi \quad \phi \vee \psi \quad \phi \rightarrow \psi \quad \dots$$

son fórmulas LTL, y

Sintaxis de LTL

cualquier fórmula construida con los operadores temporales
son fórmulas LTL.



Operadores básicos en LTL

Los operadores básicos en LTL son:

$$\neg \quad \wedge \quad \circ \quad \cup$$

El resto de las operaciones se derivan de ellos:

- $\vee, \rightarrow, \leftrightarrow, \dots$ se obtienen de la forma usual.
- $\diamond \phi \equiv \text{true} \cup \phi$
- $\square \phi \equiv \neg \diamond \neg \phi$
- $\phi \text{ WU } \psi \equiv (\phi \cup \psi) \vee \square \phi$

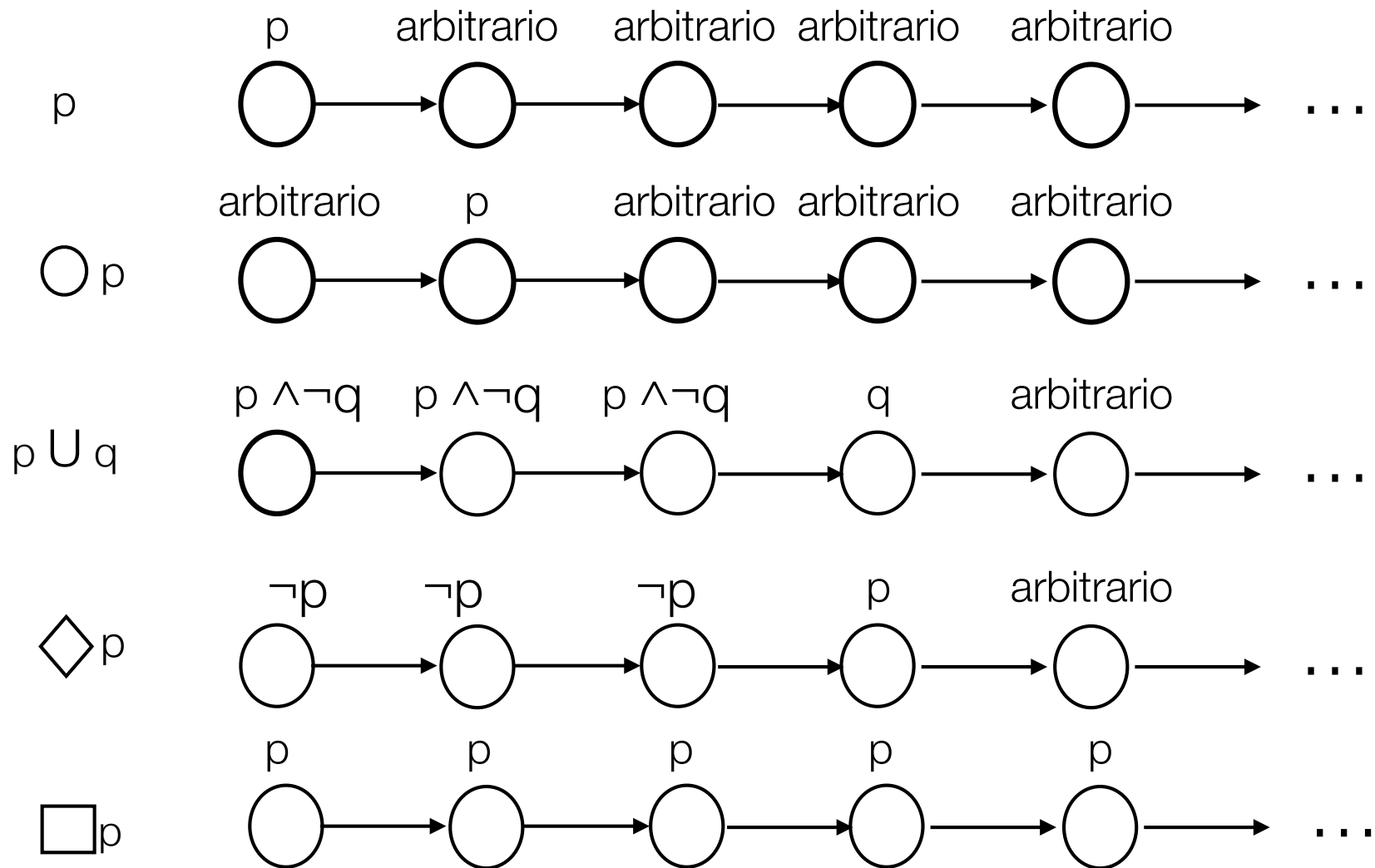
LTL (Lógica Temporal Lineal)

Combinando las modalidades \Diamond y \Box

$\Diamond \Box \varphi$ “Eventualmente para siempre φ ”

$\Box \Diamond \varphi$ “Infinitamente frecuente φ ”

Semántica de LTL: Intuición



Semántica de LTL

Dado un conjunto de fórmulas proposicionales , un modelo de ésta puede verse como un subconjunto

$$A \subseteq \mathcal{PA}$$

de proposiciones atómicas tal que para toda fórmula ,

ϕ
es verdadera en A

Semántica de LTL

Esta última noción se define inductivamente como sigue

$$\begin{array}{ll} A \models p & \text{sii} \quad p \in A \\ A \models \neg\phi & \text{sii} \quad A \not\models \phi \\ A \models \phi \wedge \psi & \text{sii} \quad A \models \phi \text{ y } A \models \psi \end{array}$$

Semántica de LTL (cont.)

Siguiendo el concepto anterior, y dado que LTL especifica el cambio de la validez de las proposiciones acorde cambia el tiempo, un modelo de una fórmula LTL sera un conjunto de secuencias infinitas de modelos. Es decir:

un modelo de una fórmula LTL es un subconjunto de $(2^{\mathcal{PA}})^{\omega}$



Conjunto Partes

Semántica de LTL

Denotaremos que:

una fórmula LTL ϕ se satisface en una traza σ

($\sigma \in (2^{\mathcal{PA}})^\omega$) por $\sigma \models \phi$

Esta última noción se define inductivamente como sigue:

Semántica de LTL

$\sigma \models p$ sii $p \in \sigma(0)$, para todo $p \in PA$

$\sigma \models \neg\phi$ sii $\sigma \not\models \phi$

$\sigma \models \phi \wedge \psi$ sii $\sigma \models \phi$ y $\sigma \models \psi$

$\sigma \models \phi \vee \psi$ sii $\sigma \models \phi$ o $\sigma \models \psi$

$\sigma \models \bigcirc \phi$ sii $\sigma[1..] \models \phi$

$\sigma \models \Diamond \phi$ sii $\exists j \geq 0 : \sigma[j..] \models \phi$

$\sigma \models \Box \phi$ sii $\forall j \geq 0 : \sigma[j..] \models \phi$

$\sigma \models \phi \mathbf{U} \psi$ sii $\exists j \geq 0 : \sigma[j..] \models \psi$ y $\forall i : 0 \leq i < j : \sigma[i..] \models \phi$

Semántica de LTL

$$\sigma \models p \quad \text{sii} \quad p \in \sigma(0), \quad \text{para todo } p \in PA$$

$$\sigma \models \neg\phi \quad \text{sii} \quad \sigma \not\models \phi$$

$$\sigma \models \phi \wedge \psi \quad \text{sii} \quad \sigma \models \phi \text{ y } \sigma \models \psi$$

$$\sigma \models \phi \vee \psi \quad \text{sii} \quad \sigma \models \phi \text{ o } \sigma \models \psi$$

$$\sigma \models \bigcirc \phi \quad \text{sii} \quad \sigma[1..] \models \phi$$

$$\sigma \models \Diamond \phi \quad \text{sii} \quad \exists j \geq 0 : \sigma[j..] \models \phi$$

$$\sigma \models \Box \phi \quad \text{sii} \quad \forall j \geq 0 : \sigma[j..] \models \phi$$

$$\sigma \models \phi \mathbin{\text{U}} \psi \quad \text{sii} \quad \exists j \geq 0 : \sigma[j..] \models \psi \text{ y } \forall i : 0 \leq i < j : \sigma[i..] \models \phi$$

$\sigma[i..]$ denota el sufijo i -ésimo de σ

Semántica de LTL (cont.)

El lenguaje de una fórmula LTL se define como el conjunto de todas las trazas que ésta satisface, i.e.,

$$\mathcal{L}(\phi) = \{\sigma \in (2^{\mathcal{PA}})^\omega \mid \sigma \models \phi\}$$

Semántica de LTL

Teorema: Los lenguajes ω -regulares son más expresivos que la lógica LTL, i.e.,

- para toda fórmula LTL ϕ , existe un lenguaje ω -regular

$$L \subseteq (2^{\mathcal{PA}})^{\omega} \text{ tal que } L = \mathcal{L}(\phi) .$$

- por otro lado, no todo lenguaje ω -regular puede expresarse con una fórmula LTL.

Algunas leyes

$$\neg \Box \phi \equiv \Diamond \neg \phi$$

$$\neg \bigcirc \phi \equiv \bigcirc \neg \phi$$

$$\Box \Box \phi \equiv \Box \phi$$

$$\Diamond \Diamond \phi \equiv \Diamond \phi$$

$$\phi \mathbf{U} (\phi \mathbf{U} \psi) \equiv \phi \mathbf{U} \psi$$

$$(\phi \mathbf{U} \psi) \mathbf{U} \psi \equiv \phi \mathbf{U} \psi$$

$$\Diamond \Box \Diamond \phi \equiv \Box \Diamond \phi$$

$$\Box \Diamond \Box \phi \equiv \Diamond \Box \phi$$

$$\Diamond \phi \equiv \text{true} \mathbf{U} \phi$$

$$\Box \phi \equiv \neg \Diamond \neg \phi$$

$$\Diamond \phi \equiv \phi \vee \bigcirc \Diamond \phi$$

$$\Box \phi \equiv \phi \wedge \bigcirc \Box \phi$$

$$\phi \mathbf{U} \psi \equiv \psi \vee (\phi \wedge \bigcirc (\phi \mathbf{U} \psi))$$

$$(\Diamond \phi) \vee (\Diamond \psi) \equiv \Diamond (\phi \vee \psi)$$

$$(\Box \phi) \wedge (\Box \psi) \equiv \Box (\phi \wedge \psi)$$

$$\bigcirc (\phi \mathbf{U} \psi) \equiv (\bigcirc \phi) \mathbf{U} (\bigcirc \psi)$$

Especificación de propiedades con LTL: **Safety**

Nunca va a pasar nada malo

Las propiedades safety **usualmente** tienen la forma:

$\Box \phi$ Siempre ocurre ϕ (o Nunca ocurre $\neg \phi$)

Ejemplos:

Exclusion mutua: Siempre pasa que al menos uno de los dos procesos no esta en la región crítica

$$\Box(\neg crit_1 \vee \neg crit_2)$$

Especificación de propiedades con LTL: **Safety**

Ausencia de deadlock en el problema de los filósofos.

$$\Box \neg \left(\bigwedge_{i=0}^{n-1} \text{espera_fil}_i \wedge \bigwedge_{j=0}^{n-1} \text{ocupado_tenedor}_j \right)$$

Asumimos n filósofos y n tenedores (indexados del 0 al n)

Especificación de propiedades con LTL: **Safety**

Una persona educada no entra sin golpear.

$$\neg(\neg knock \mathcal{U} enter)$$

OJO! si puede golpear dos veces!

Especificación de propiedades con LTL: **Liveness**

Siempre es posible que algo bueno

Las propiedades liveness **usualmente** tienen la forma:

$\diamond \phi$ en algún momento en el futuro ocurre ϕ

Ejemplos:

Un proceso dado termina en algún momento en el futuro.

$\diamond \textit{fin_proc}$

Especificación de propiedades con LTL: **Liveness**

El auto rojo nro. 0 ingresa frecuentemente al puente (progreso).

$\square \diamond red.0.enter$

Especificación de propiedades con LTL: **Liveness**

Los procesos ingresan frecuentemente a su región crítica.

$$\square \diamond crit_1 \wedge \square \diamond crit_2$$

Especificación de propiedades con LTL: **Fairness**

“Bajo ciertas condiciones un evento ocurrirá de manera frecuente”

Hay diversas formas de fairness.

Fairness incondicional

“Un evento ϕ debe ocurrir de manera frecuente”

Ejemplo:

$\square \diamond red.0.enter$

$\square \diamond crit_1 \wedge \square \diamond crit_2$

Especificación de propiedades con LTL: **Fairness**

Weak fairness (fairness condicional débil)

“Siempre ocurre que cuando un evento **permanece continuamente** habilitado, entonces finalmente se ejecutará”

$$\Box(\Diamond \Box \textit{habilitado}(a) \rightarrow \Diamond \textit{ejecutar}(a))$$

En general, podemos pensar que si ϕ y ψ son dos fórmulas, más generalmente podemos escribir

$$\Box(\Diamond \Box \phi \rightarrow \Diamond \psi)$$

Luego ϕ es la condición (débil) para la ocurrencia de ψ .

Especificación de propiedades con LTL: **Fairness**

Weak fairness (fairness condicional débil)

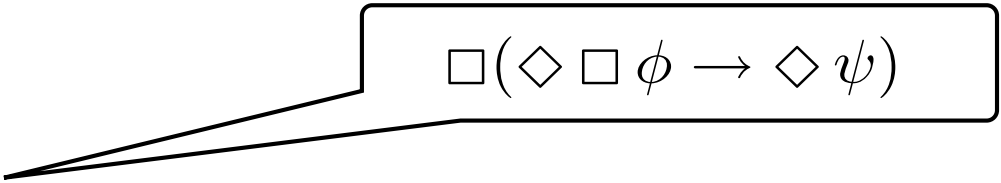
La fórmula anterior se puede escribir equivalentemente de las siguientes maneras:

$$\Diamond \Box \phi \rightarrow \Box \Diamond \psi$$

$$\Box (\Box \phi \rightarrow \Diamond \psi)$$

Especificación de propiedades con LTL: **Fairness**

Weak fairness (fairness condicional débil)


$$\Box(\Diamond \Box \phi \rightarrow \Diamond \psi)$$

La fórmula anterior se puede escribir equivalentemente de las siguientes maneras:

$$\Diamond \Box \phi \rightarrow \Box \Diamond \psi$$

$$\Box(\Box \phi \rightarrow \Diamond \psi)$$

Especificación de propiedades con LTL: **Fairness**

Strong fairness (fairness condicional fuerte)

“Siempre ocurre que, si un evento está **frecuentemente** habilitado, entonces finalmente se ejecutará”

$$\Box(\Box \Diamond \textit{habilitado}(a) \rightarrow \Diamond \textit{ejecutar}(a))$$

Como antes, podemos escribir

equivalentemente

$$\Box \Diamond \phi \rightarrow \Box \Diamond \psi$$

donde ϕ será la condición (fuerte) para la ocurrencia de ψ .

Especificación de propiedades con LTL: **Fairness**

Strong fairness (fairness condicional fuerte)

“Siempre ocurre que, si un evento está **frecuentemente** habilitado, entonces finalmente se ejecutará”

$$\Box(\Box \Diamond \textit{habilitado}(a) \rightarrow \Diamond \textit{ejecutar}(a))$$

Como antes, podemos escribir

$$\Box(\Box \Diamond \phi \rightarrow \Diamond \psi)$$

equivalentemente

$$\Box \Diamond \phi \rightarrow \Box \Diamond \psi$$

donde ϕ será la condición (fuerte) para la ocurrencia de ψ .

Otras propiedades con LTL

Respuesta: Cada vez que ocurre p se emite una respuesta q .

$$\Box(p \rightarrow \Diamond q)$$

Ejemplo: Todo mensaje enviado se recibe en algún momento:

$$\Box(\text{enviar}(m) \rightarrow \Diamond \text{recibir}(m))$$

Persistencia: A partir de algún momento q se hace invariante, (i.e., q persiste).

$$\Diamond \Box q$$

Otras propiedades con LTL

La siguiente es una forma condicional de persistencia

$$\Diamond(p \rightarrow \Box q)$$

Ejemplo: En un protocolo de elección de líder, el nodo electo debe permanecer líder:

$$\Diamond(\textit{electo_nodo}_i \rightarrow \Box \textit{líder_nodo}_i)$$

Especificar las siguientes propiedades

Una persona educada golpea una única vez antes de entrar (no entra sin golpear ni golpea más de una vez)

Propiedades de un semáforo:



Si el semáforo esta en rojo no puede inmediatamente ponerse en verde

Una vez que el semáforo se ponga en rojo, se podrá en algún momento en verde luego de estar en amarillo por algún tiempo.

Bibliografía

Cap 5 (Sección 5.1), Principles of Model Checking,
Christel Baier and Joost-Pieter Katoen