Ingeniería del Software II
5 - Lógica Temporal Lineal

Limitaciones de los 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.

Limitaciones de los métodos para representar p

Alternativamente

$$\left(\left(\Sigma \setminus \{\text{llueve}\}\right)^* \left(\text{llueve}\, \Sigma^* \left(\neg\text{llueve}\right)\right)^*\right)^\omega$$

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:

$$\left(\left(\Sigma \setminus \{\text{llueve}\}\right) + \left(\text{llueve}\, \Sigma^* \left(\neg\text{llueve}\right)\right)\right)^\omega$$

CUIDADO:
notación sugestiva

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

Necesitamos una lógica que exprese directamente este tipo de propiedades.

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: $\text{llueve} \rightarrow \neg\text{llueve}$

Esta propiedad sólo es verdadera si no llueve :-(

2do. intento: $\text{primero\_llueve} \rightarrow \text{luego\_para}$

Dos inconvenientes:

1. Disocia los conceptos de llover y dejar de llover.
2. ¿Cómo diferenciamos que esto ocurre siempre o que esto ocurre sólo una vez?

Otros ejemplos podrán llevar a otros problemas.

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, satisfactibilidad en 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".

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

- Lógica deóntica: □ significa "obligatoriamente"
  ◇ significa "está permitido"
- Lógica temporal: □ significa "siempre en el futuro"
  ◇ 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
- 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 la herramientas de model checking:

- Lógica temporal lineal proposicional (LTL o PLTL)
- Lógica de computaciones ramificadas (CTL)
- Lógica de computaciones ramificadas y temporizadas (TCTL)

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.

Nos concentraremos en el uso de LTL.

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"

$\phi\,\mathtt{U}\,\psi$: "ϕ sucede hasta que suceda ψ" psi sucede en algún momento

$\phi\,\mathtt{WU}\,\psi$: "ϕ sucede siempre o hasta que suceda ψ" no sucede psi necesariamente

En realidad, LTL sólo provee los operadores modales **U** y ○. Todos los otros son derivados de estos dos.

- llueve U ¬no llueve (si no llueve, ya se hizo verdadera).
- (llueve -> (rombo) ¬llueve)

ESTAS DOS NO SIEMPRE SON EQUIVALENTES

Sintáxis 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

$$\neg\phi \qquad \phi \wedge \psi \qquad \phi \vee \psi \qquad \phi \rightarrow \psi \qquad \cdots$$

son fórmulas LTL, y

- cualquier fórmula construida con los operadores temporales

$$\bigcirc \phi \qquad \phi \,\mathtt{U}\, \psi \qquad \Box \phi \qquad \Diamond \phi \qquad \cdots$$

son fórmulas LTL.

Semántica de LTL

Recordar: Dado una teoría proposicional (i.e., un conjunto de fórmulas proposicionales que incluye los axiomas de la lógica proposicional), todo modelo de ésta puede verse como un subconjunto

$A \subseteq \mathcal{PA}$ A son las proposiones atómicas verdaderas

de proposiciones atómicas donde las fórmulas de ese conjunto se hacen verdaderos.

En particular, diremos que

una fórmula proposicional ϕ se satisface en $A \subseteq \mathcal{PA}$ si $A \models \phi$.

Esta última noción se define inductivamente como sigue:

$A \models p$ sii $p \in A$

$A \models \neg\phi$ sii $A \not\models \phi$

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

Función de interpretación I
I: PA -> {bottom, top}
I(¬psi) = {bottom si I(psi) = top
top si I(psi) = bottom}
I(psi /\ phi) = {top si I(psi) = I(phi) = top,
bottom c.c.}

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 de teorías proposicionales.

Es decir:

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

hay una biyección entre partes(PA) = 2^pa como notación, esto permite definir la función de interpretación en función de conjuntos.

Denotaremos que:

una fórmula LTL ϕ se satisface en una traza σ ($\sigma \in \left(2^{\mathcal{PA}}\right)^{\omega}$) por $\sigma \models \phi$.

sigma es una secuencia infinita de conjuntos

Esta última noción se define inductivamente como sigue:

Semántica de LTL (cont.)

sigma describe a partir de 'ahora', es decir el primer conjunto es el tiempo actual.

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

esto tiene que ser una secuencia, estaría mal poner un conjunto

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

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

Todas las expresiones que nos interesan en la materia hablan del futuro, entonces para cambiar el 'a partir de ahora' al futuro, tomo un sufijo de sigma. Hay lógicas modales en las que puedo hablar del pasado, pero no las vamos a ver acá.

hacer cosas por inducción acá es un lio porque estamos entrando en dominios infinitos y puede hacer lio, por eso la definición de la función no es recursiva.

$\sigma[i..]$ 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.,

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

Antes hablaba de propiedades como conjuntos de trazas. Esto es lo mismo.

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

todas las fórmulas phi las puedo expresar como exp regular

- para toda fórmula LTL ϕ, existe un lenguaje ω-regular $L \subseteq \left(2^{\mathcal{PA}}\right)^{\omega}$ tal que $L = \mathcal{L}(\phi)$.
- por otro lado, el lenguaje $\left((\varnothing + \{p\})\ \{p\} \right)^{\omega}$ no es expresable con una fórmula LTL.

No todo lenguaje regular se puede expresar como fórmula LTL

(rombo) P = $\sum^*$ ({p}+{p,q}) $\sum^w$

Operadores básicos en LTL

Los operadores básicos en LTL son:

$$\neg \qquad \wedge \qquad \bigcirc \qquad \mathtt{U}$$

El resto de las operaciones se derivan de ellos:

- $\vee$, $\rightarrow$, $\leftrightarrow$, $\ldots$ se obtienen de la forma usual.
- $\Diamond \phi \equiv \mathtt{true}\, \mathtt{U}\, \phi$
- $\Box \phi \equiv \neg \Diamond \neg \phi$
- $\phi\, \mathtt{WU}\, \psi \equiv (\phi\, \mathtt{U}\, \psi) \vee \Box \phi$

Algunas leyes

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

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

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

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

$$\phi \,\mathtt{U}\, (\phi \,\mathtt{U}\, \psi) \equiv \phi \,\mathtt{U}\, \psi$$

$$(\phi \,\mathtt{U}\, \psi) \,\mathtt{U}\, \psi \equiv \phi \,\mathtt{U}\, \psi$$

frecuentemente

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

persistencia

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

siempre existe un momento en el que phi valga siempre

$$\Diamond \phi \equiv \mathtt{true}\, \mathtt{U}\, \phi$$

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

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

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

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

la prop de abajo no vale para la conjunción

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

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

$$\bigcirc(\phi \,\mathtt{U}\, \psi) \equiv (\bigcirc \phi) \,\mathtt{U}\, (\bigcirc \psi)$$

Especificación de propiedades con LTL: Safety

tienen la pinta de un invariante

Las propiedades safety usualmente tienen la forma:

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

Ejemplos:

para pensar prop de safety, me conviene pensar en qué no quiero que ocurra.

- Exclusión mutua.

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

- Ausencia de deadlock en el problema de los filósofos

$$\Box\,\neg\left(\bigwedge_{i=0}^{n} espera\_fil_i \wedge \bigwedge_{j=0}^{n} ocupado\_tenedor_j\right)$$

- Una persona educada no entra sin golpear.

$$\neg(\neg knock\,\mathtt{U}\ enter)$$

Especificación de propiedades con LTL: Safety

Las propiedades safety usualmente tienen la forma:

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

Ejemplos:

- Exclusión mutua.

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

- Ausencia de deadlock en el problema de los filósofos

$$\Box\neg\left(\bigwedge_{i=0}^{n} espera\_fil_i \wedge \bigwedge_{j=}^{n}\right.$$

- Una persona educada no ent

$$\neg(\neg knock\,\mathtt{U}\ enter)$$

que es equivalente a

$$\neg enter\,\mathtt{WU}\,(knock \wedge \neg enter)$$

Especificación de propiedades con LTL: Liveness

Las propiedades liveness usualmente tienen la forma:

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

hay veces que las propiedades no se pueden reducir a usar el rombo al principio.

Ejemplos:

- Un proceso dado termina en algún momento en el futuro.

$\Diamond fin\_proc$

- El auto rojo nro. 0 ingresa frecuentemen
(progreso).

$\Box \Diamond red.0.enter$

siempre ocurre que entra el auto rojo

para todo j >= 0, existe i>=j tal que entra_rojo pertenece sigma(i)

así como está, no es una propiedad de liveness, Le agregamos la regla para q sea de liveness

Recordar:

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

- Los procesos ingresan frecuentemente a su región crítica.

$\Box \Diamond crit_1 \wedge \Box \Diamond crit_2$

|_| (rombo crit_1 v rombo crit_2) está mal porque se hace válida sólo si entra 1 sola, y yo quiero decir que entran a las 2

Especificación de propiedades con LTL: Fairness

"Bajo ciertas condiciones un evento ocurrirá de manera frecuente"

!! En LTSA no se puede chequear Fairness

Hay diversas formas de fairness.

Progreso (o fairness incondicional)

"Un evento ϕ debe ocurrir de manera frecuente"

Ejemplo:

$$\Box \Diamond red.0.enter$$

$$\Box \Diamond crit_1 \wedge \Box \Diamond crit_2$$

Especificación de propiedades con LTL: Fairness

Weak fairness (fairness condicional débil)

"Siempre ocurre que cuando un evento permanece continuamente habilitado, entonces finalmente se ejecutará"

$$\Box(\underbrace{\Diamond \Box}_{\text{persistencia}} habilitado(a) \rightarrow \Diamond\, ejecutar(a))$$

Más generalmente, si ϕ y ψ son dos fórmulas, weak fairness podría escribirse

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

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

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)

En muchos casos, weak fairness no es lo suficientemente fuerte. Consideremos, por ejemplo, el siguiente programa concurrente:

```
COIN = ( toss -> heads -> COIN
       | toss -> tails -> COIN).
```

Una traza de ejecución para COIN, que no viola weak fairness, es la siguiente:

```
toss, heads, toss, heads, toss, heads, toss, heads, ...
```

Para evitar este tipo de trazas, debe considerarse un tipo más fuerte de fairness denominado strong fairness.

Especificación de propiedades con LTL: Fairness

Strong fairness (fairness condicional fuerte)

"Siempre ocurre que, si un evento está frecuentemente habilitado, entonces finalmente se ejecutará"

$$\Box(\Box \Diamond habilitado(a) \rightarrow \Diamond ejecutar(a))$$

Como antes, podemos escribir

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

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

La fórmula anterior se puede escribir equivalentemente como

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

Especificación de propiedades con LTL: Otras propiedades

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:

$$\Box(enviar(m) \rightarrow \Diamond\, recibir(m))$$

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

$$\Diamond\, \Box\, q$$

La siguiente es una forma condicional de persistencia

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

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

$$\Box(electo\_nodo_i \rightarrow \Box\, líder\_nodo_i)$$