Introducción a una lógica modal para procesos concurrentes

Alejandro Sánchez

Departamento de Informática
Universidad Nacional de San Luis

Universidad Tecnológica Nacional
Facultad Regional Tucumán
14-15 Junio

Contenidos

Introducción a una lógica modal
para procesos concurrentes

- Lógica de Hennessy-Milner
- Formulas regulares

Motivación

Comprobar la correctitud de un sistema
con respecto a una especificacióm

Comprobar equivalencia (ej. $\sim$)
no resulta apropiado para verificar propiedades como

"puede el sistema realizar una acción α y luego una β"

Se debe explorar el espacio de estados del proceso

Sintaxis y semántica – lógica proposicional

$$\varphi, \psi ::= \mathit{true} \mid \mathit{false} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle a \rangle \varphi \mid [a]\varphi$$

La formula es interpretada sobre el LTS del proceso

- *true* es satisfecho en cada estado de un proceso
- *false* nunca es satisfecho
- $\neg\varphi$ es satisfecha cuando φ no lo es
- $\varphi \wedge \psi$ es satisfecha si φ y ψ son satisfechas
- $\varphi \vee \varphi$ es satisfecha si φ o ψ son satisfechas
- $\varphi \rightarrow \psi$ es satisfecha si $\neg\varphi \vee \psi$ es satisfecha

Sintaxis y semántica – posibilidad I

$$\varphi, \psi ::= \mathit{true} \mid \mathit{false} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle a \rangle \varphi \mid [a]\varphi$$

- $\langle a \rangle \varphi$ es satisfecho cuando es posible una acción a y luego φ es satisfecho

Ejemplos

$$\langle a \rangle \langle b \rangle \langle c \rangle \mathit{true}$$

$$\langle a \rangle (\langle b \rangle \mathit{true} \wedge \langle c \rangle \mathit{true})$$

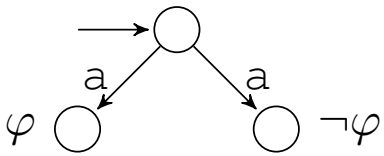
$$\langle a \rangle \neg \langle b \rangle \mathit{true}$$

Sintaxis y semántica – posibilidad II

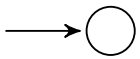
$$\varphi, \psi ::= \mathit{true} \mid \mathit{false} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle a \rangle \varphi \mid [a]\varphi$$

- $\langle a \rangle \varphi$ es satisfecho cuando es posible una acción a y luego φ es satisfecho

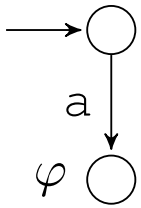
Ejemplos



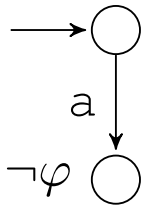
$\models \langle a \rangle \varphi$



$\not\models \langle a \rangle \varphi$



$\models \langle a \rangle \varphi$



$\not\models \langle a \rangle \varphi$

Sintaxis y semántica – necesidad I

$$\varphi, \psi ::= \mathit{true} \mid \mathit{false} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle a \rangle \varphi \mid [a]\varphi$$

- $[a]\varphi$ es satisfecha cuando por cada acción a que puede ser hecha, φ es satisfecha luego de hacer esta acción a

Ejemplos

$$[a]\langle b \rangle \mathit{true}$$

$$[a]\mathit{false}$$

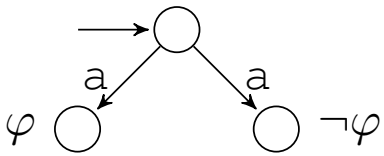
$$[a][b]\mathit{false}$$

Sintaxis y semántica – necesidad II

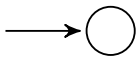
$$\varphi, \psi ::= \mathit{true} \mid \mathit{false} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle a \rangle \varphi \mid [a]\varphi$$

- $[a]\varphi$ es satisfecha cuando por cada acción a que puede ser hecha, φ es satisfecha luego de hacer esta acción a

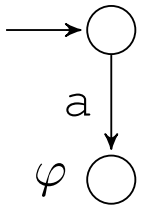
Ejemplos



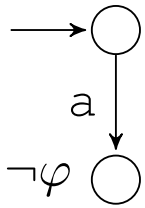
$\not\models [a]\varphi$



$\models [a]\varphi$



$\models [a]\varphi$



$\not\models [a]\varphi$

Identidades

$$\neg\langle a\rangle\varphi = [a]\neg\varphi \qquad \neg[a]\varphi = \langle a\rangle\neg\varphi$$
$$\langle a\rangle \mathit{false} = \mathit{false} \qquad [a]\mathit{true} = \mathit{true}$$
$$\langle a\rangle(\varphi \vee \psi) = \langle a\rangle\varphi \vee \langle a\rangle\psi \qquad [a](\varphi \wedge \psi) = [a]\varphi \wedge [a]\psi$$
$$\langle a\rangle\varphi \wedge [a]\psi \Rightarrow \langle a\rangle(\varphi \wedge \psi)$$

Sintaxis y semántica de formulas de acción

$$\alpha, \beta ::= a_1 \mid \ldots \mid a_n \mid \mathit{true} \mid \mathit{false} \mid \overline{\alpha} \mid \alpha \cap \beta \mid \alpha \cup \beta$$

La formula es interpretada sobre conjuntos de acciones

- $a_1 \mid \ldots \mid a_n$ conjunto unitario con la multiacción en él
- *true* es el conjunto de todas las acciones
- *false* es el conjunto vacío
- $\overline{\alpha}$ es el complemento del conjunto definido por α
- $\mid \alpha \cap \beta$ es la intersección de los conjuntos dados por α y β
- $\mid \alpha \cup \beta$ es la unión de los conjuntos dados por α y β

Semántica para modalidades

$$\langle \alpha \rangle \varphi = \bigvee_{a \in \alpha} \langle a \rangle \varphi$$

$$[\alpha] \varphi = \bigwedge_{a \in \alpha} [a] \varphi$$

Ejemplos

$$\langle true \rangle \langle a \rangle \; true$$

$$[true] \; false$$

$$\langle \overline{a} \rangle \langle b \cup c \rangle \; true$$

$$[\overline{a}] false$$

Propiedades importantes

- Progreso: $\langle true \rangle true$
- Inevitabilidad de a: $\langle true \rangle true \wedge [\overline{a}]\ false$
- Deadlock o terminación: $[true]\ false$

Sintaxis y semántica de formulas regulares I

$$R, S ::= \epsilon \mid \alpha \mid R.S \mid R + S \mid R^* \mid R^+$$

- ϵ representa la secuencia vacía de acciones
$$[\epsilon]\varphi = \langle\epsilon\rangle\varphi = \varphi$$
- α es una acción del conjunto definido por α

Sintaxis y semántica de formulas regulares II

$$R, S ::= \epsilon \mid \alpha \mid R.S \mid R + S \mid R^* \mid R^+$$

- $R.S$ representa la concatenación de las secuencias de acciones R y S

$$\langle a.b.c \rangle \varphi = \langle a \rangle \langle b \rangle \langle c \rangle \varphi$$

$$\langle R.S \rangle \varphi = \langle R \rangle \langle S \rangle \varphi \qquad [R.S] \varphi = [R][S] \varphi$$

- $R + S$ representa la unión de las secuencias en R y en S

  $[a.b + c.d]$ *false* expresa que ni la secuencia $a.b$ ni la secuencia $c.d$ es posible

$$\langle R + S \rangle \varphi = \langle R \rangle \varphi \vee \langle S \rangle \varphi \qquad [R + S] \varphi = [R] \varphi \wedge [S] \varphi$$

Sintaxis y semántica de formulas regulares III

$$R, S ::= \epsilon \mid \alpha \mid R.S \mid R + S \mid R^* \mid R^+$$

- R^* representa cero o más repeticiones de la secuencia R
  $[a^*]$ *true* expresa que cualquier secuencia de a es posible
  $$\Box\varphi = [true^*]\varphi \qquad \Diamond\varphi = \langle true^* \rangle \varphi$$
- R^+ representa una o más repeticiones de la secuencia R
  $[a^+]$ φ expresa que φ es satisfecha en cualquier estado accesible por una secuencia de una o más acciones a

Liveness

$$\langle true^* \rangle \varphi$$

- Liveness: una condición estará eventualmente presente en el sistema

$$[send]\langle true^* . receive \rangle true$$

luego de enviar un mensaje,
este es eventualmente recibido

Safety

$$[true^*]\varphi$$

- Safety: el sistema no exhibe una propiedad

$$[true^*.enter.\overline{leave}^*.enter]false$$

es imposible realizar dos llegadas consecutivas sin una partida intermedia

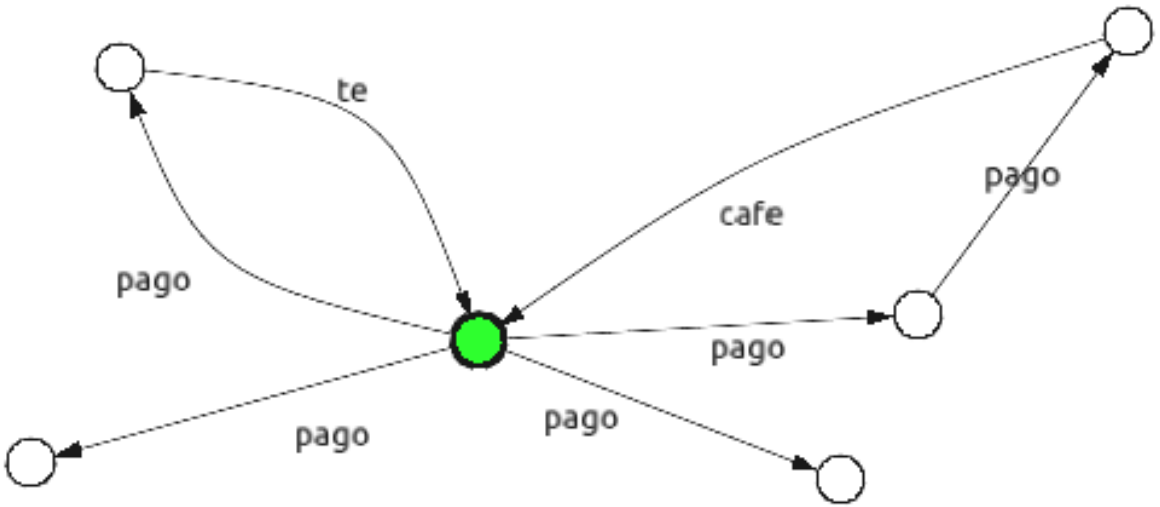
$$[true^*]\langle true\rangle true$$

ausencia de deadlock

Propiedades de la máquina expendedora de bebidas I

siempre, eventualmente, un pago es posible

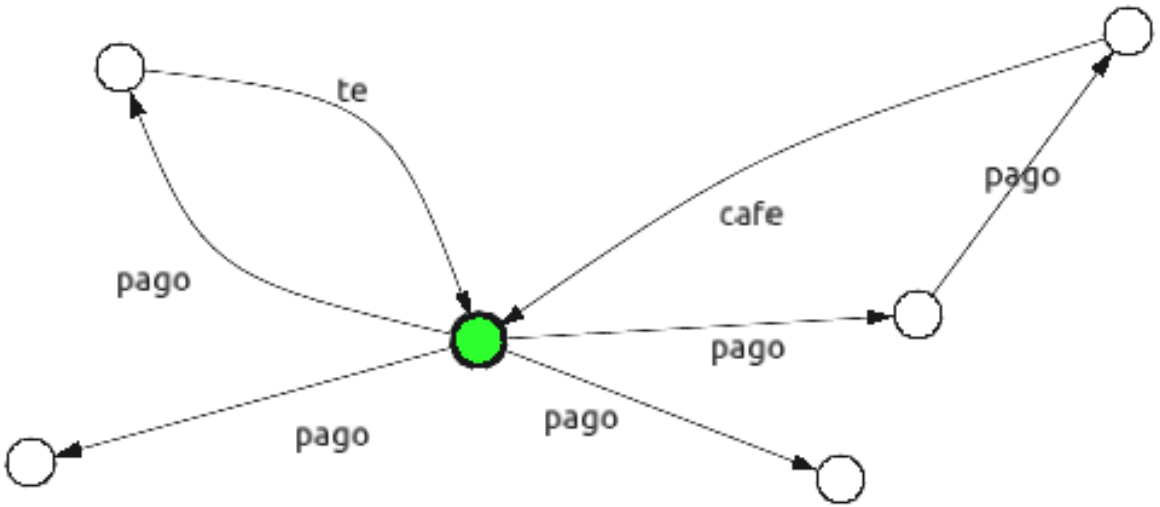
$$[true^*]\langle true^*.pago\rangle true$$



Propiedades de la máquina expendedora de bebidas II

siempre, después de un té o un café, sigue un pago

$$[true^*.te.\overline{pago}]false \wedge [true^*.cafe.\overline{pago}]false$$



Ejercicios

- Corrija la máquina expendedora para que satisfaga
$$[true^*]\langle true^*.pago\rangle true$$
- Defina propiedades y verifíquelas en el caso de la integración de la actualización de clientes