

# Verificación de Sistemas Reactivos

## *Una Introducción a Lógicas Modales*

Alejandro Sánchez

Departamento de Informática  
Universidad Nacional de San Luis

Maestría en Ingeniería de Software  
Maestría en Calidad del Software  
Especialización en Ingeniería de Software  
25-26 Octubre 2013

# Estructura de la presentación

## 1 Lógica de Hennessy-Milner

- Sintaxis y semántica
- Identidades

## 2 Formulas de acción

- Sintaxis y semántica
- Propiedades relevantes

## 3 Formulas regulares

- Sintaxis y semántica
- Tipos de propiedades
- Ejemplos

Comprobar la correctitud de un sistema  
con respecto a una especificación

Comprobar equivalencia (ej.  $\sim$ )  
no resulta apropiado para verificar propiedades como  
“puede el sistema realizar una acción  $\alpha$  y luego una  $\beta$ ”

Se debe explorar el espacio de estados del proceso

# Estructura de la presentación

## 1 Lógica de Hennessy-Milner

- Sintaxis y semántica
- Identidades

## 2 Formulas de acción

- Sintaxis y semántica
- Propiedades relevantes

## 3 Formulas regulares

- Sintaxis y semántica
- Tipos de propiedades
- Ejemplos

# Sintaxis y semántica – lógica proposicional

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

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  $\varphi$  no lo es
- $\varphi \wedge \psi$  es satisfecha si  $\varphi$  y  $\psi$  son satisfechas
- $\varphi \vee \psi$  es satisfecha si  $\varphi$  o  $\psi$  son satisfechas
- $\varphi \rightarrow \psi$  es satisfecha si  $\neg\varphi \vee \psi$  es satisfecha

# Sintaxis y semántica – posibilidad I

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

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

## Ejemplos

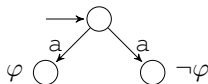
$$\begin{aligned} &\langle a \rangle \langle b \rangle \langle c \rangle \text{true} \\ &\langle a \rangle (\langle b \rangle \text{true} \wedge \langle c \rangle \text{true}) \\ &\langle a \rangle \neg \langle b \rangle \text{true} \end{aligned}$$

# Sintaxis y semántica – posibilidad II

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

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

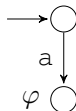
## Ejemplos



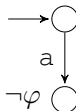
$$\models \langle a \rangle \varphi$$



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



$$\models \langle a \rangle \varphi$$



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

# Sintaxis y semántica – necesidad I

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

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

## Ejemplos

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

$[a]\text{false}$

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

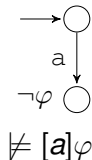
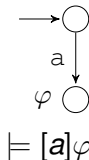
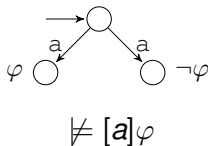


# Sintaxis y semántica – necesidad II

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

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

## Ejemplos



# Identidades

$$\neg \langle a \rangle \varphi = [a] \neg \varphi$$

$$\langle a \rangle \text{false} = \text{false}$$

$$\langle a \rangle (\varphi \vee \psi) = \langle a \rangle \varphi \vee \langle a \rangle \psi$$

$$\langle a \rangle \varphi \wedge [a] \psi \Rightarrow \langle a \rangle (\varphi \wedge \psi)$$

$$\neg [a] \varphi = \langle a \rangle \neg \varphi$$

$$[a] \text{true} = \text{true}$$

$$[a] (\varphi \wedge \psi) = [a] \varphi \wedge [a] \psi$$

# Estructura de la presentación

## 1 Lógica de Hennessy-Milner

- Sintaxis y semántica
- Identidades

## 2 Formulas de acción

- Sintaxis y semántica
- Propiedades relevantes

## 3 Formulas regulares

- Sintaxis y semántica
- Tipos de propiedades
- Ejemplos

# Sintaxis y semántica de formulas de acción

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

La formula es interpretada sobre conjuntos de acciones

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

# Semántica para modalidades

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

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

# Ejemplos

$\langle true \rangle \langle a \rangle \text{ true}$

$[true] \text{ false}$

$\langle \bar{a} \rangle \langle b \cup c \rangle \text{ true}$

$[\bar{a}] \text{ false}$

# Propiedades importantes

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

# Estructura de la presentación

## 1 Lógica de Hennessy-Milner

- Sintaxis y semántica
- Identidades

## 2 Formulas de acción

- Sintaxis y semántica
- Propiedades relevantes

## 3 Formulas regulares

- Sintaxis y semántica
- Tipos de propiedades
- Ejemplos



# Sintaxis y semántica de formulas regulares I

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

- $\epsilon$  representa la secuencia vacía de acciones

$$[\epsilon]\varphi = \langle \epsilon \rangle \varphi = \varphi$$

- $\alpha$  es una acción del conjunto definido por  $\alpha$

# 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  $S$

$$\begin{aligned} \langle a.b.c \rangle \varphi &= \langle a \rangle \langle b \rangle \langle c \rangle \varphi \\ \langle R.S \rangle \varphi &= \langle R \rangle \langle S \rangle \varphi & [R.S] \varphi &= [R][S] \varphi \end{aligned}$$

- $R + S$  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

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

# 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

$$\Box\varphi = [\text{true}^*]\varphi \quad \Diamond\varphi = \langle \text{true}^* \rangle\varphi$$

- $R^+$  representa una o más repeticiones de la secuencia  $R$  [ $a^+$ ]  $\varphi$  expresa que  $\varphi$  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

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

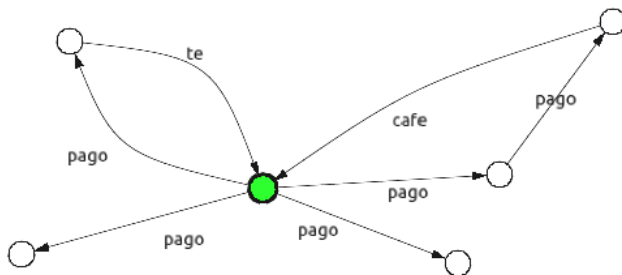
es imposible realizar dos llegadas consecutivas sin una partida intermedia

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

ausencia de deadlock

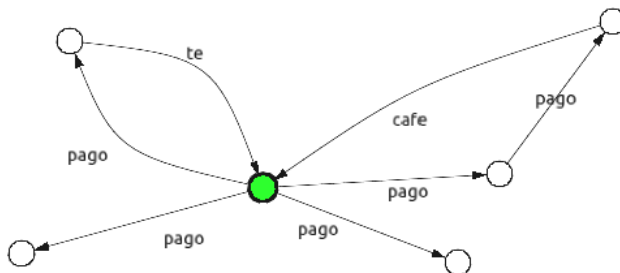
# Propiedades de la máquina expendedora de bebidas I

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