

## Taller 5

Grupo: Giudice Carlos 694/15 | Rosende Federico 222/16

### Ejercicio 2

Punto fijo para bisimulación débil

$$\begin{aligned}\sim 0 &= \{ \\ &(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), \\ &(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), \\ &(2, 0), (2, 1), (2, 2), (2, 3), (2, 4), \\ &(3, 0), (3, 1), (3, 2), (3, 3), (3, 4) \\ &\} \\ \sim 1 &= \{ \\ &(0, 0), \\ &(1, 3), \\ &(2, 1), (2, 4) \\ &(3, 2) \\ &\} \\ \sim 2 &= \{ \\ &(0, 0), \\ &(1, 3), \\ &(2, 1), (2, 4) \\ &(3, 2) \\ &\}\end{aligned}$$

Luego de  $\sim 2$  el algoritmo termina porque es igual a  $\sim 1$  y se encontró el punto fijo.

Por lo tanto, resultan bisimilarias y se exhibe con la relación  $R = \{(0;0), (1;3), (2;1), (2;4), (3;2)\}$

### Ejercicio 3

Atacante  $\rightarrow a$

Defensor  $\rightarrow d$

- $(P_0; Q_0) \rightarrow a$  se mueve por  $a$  en  $P \rightarrow (P_1; Q_0) \rightarrow d$  se mueve por  $a$  en  $Q \rightarrow (P_1; Q_1) \rightarrow a$  se mueve por  $\tau$  en  $P \rightarrow (P_2; Q_1)$

Dos opciones para  $d$ :

- $\rightarrow d$  se queda en el lugar en  $Q \rightarrow (P_2; Q_1) \rightarrow a$  se mueve por  $\tau$  en  $Q \rightarrow (P_2; Q_2) \rightarrow d$  se queda en el lugar en  $P \rightarrow a$  se mueve por  $c$  en  $P \rightarrow (P_3; Q_2) \Rightarrow$  Gana el atacante
- $\rightarrow d$  se mueve por  $\tau$  en  $Q \rightarrow (P_2; Q_2) \rightarrow a$  se mueve por  $c$  en  $P \rightarrow (P_3; Q_2) \Rightarrow$  Gana el atacante

### Ejercicio 5

Se ocultaron todas las etiquetas del LTS de PRIMES que no aparecen en el alfabeto de la especificación. Luego se creó el procesos SYS como la composición de PRIMES con su especificación y se lo animó para corroborar que el comportamiento es el esperado. Posteriormente, se realizó un check de bisimilaridad con la herramienta para respaldar lo observado. Concluimos que el proceso con las etiquetas ocultas resulta bisimilar a la especificación dada.

### Ejercicio 9

Se verificó el comportamiento en la herramienta y se obtuvieron estos resultados:

- Para la propiedad “no hay entradas luego del cierre” la salida indicaba que no había violaciones a la misma.
- Para la propiedad “no hay salidas luego del cierre” la salida exhibió lo siguiente:

---

```

Trace to property violation in NO_EXITS_AFTER_CLOSING:
c.empty
c.open
a.entry
c.close
b.exit

```

Figure 1: Hay salidas luego del cierre.

## Ejercicio 12

1. progress ANY\_LEADER = {proc[i:1..N].leader}
2. progress ONE\_LEADER = {proc[1].leader}

## Ejercicio 13

El contraejemplo es el siguiente:

$X = (x \rightarrow xx \rightarrow X).$

$XX = (xx \rightarrow x \rightarrow XX).$

$\parallel XXX = (XX \parallel X).$

progress EQUIS = {x}

La propiedad de progreso EQUIS vale para el proceso X pero al componerlo con XX se obtiene un proceso que no puede avanzar por ninguna transición por lo que nunca puede valer EQUIS.

## Ejercicio 14

1.  $\Box \langle \rangle \text{enBase}$

$\Rightarrow$  Propiedad de liveness  $\rightarrow$  Un contraejemplo no puede ser finito porque se necesita ver que nunca regrese a base luego de salir.

2.  $\Box(\text{bateriaBaja} \Rightarrow X(\text{modoAhorro} \cup \text{enBase}))$

$\Rightarrow$  Propiedad de safety  $\rightarrow$  Un contraejemplo viene dado por una traza en la que se observa *bateríaBaja* y en el siguiente estado no ocurre *modoAhorro*

3.  $\Box((\text{paredDelante} \Rightarrow X(\text{girandoALaIzquierda} \cup \neg \text{paredDelante})) \wedge (\neg \text{paredDelante} \Rightarrow \neg X \text{ girandoALaIzquierda}))$

$\Rightarrow$  Propiedad de safety  $\rightarrow$  Un contraejemplo está dado por una traza en la que se observa *paredDelante* y no ocurre *girandoALaIzquierda* en el siguiente estado.

Observación:

- Esta especificación limita el comportamiento ya que no puede girar a la izquierda a menos que vea una pared. Podría no limitarse el giro (permitiendo que siga girando aunque no vea una pared) eliminando la parte derecha de la conjunción y obteniendo una propiedad así:

–  $\Box(\text{paredDelante} \Rightarrow X(\text{girandoALaIzquierda} \cup \neg \text{paredDelante}))$

## Ejercicio 17

1.  $\Box(\text{salióDeBase} \Rightarrow \langle \rangle \text{entróABase})$
2.  $\Box(\text{bateríaBaja} \Rightarrow (X \text{ modoAhorroOn} \Rightarrow \langle \rangle (\text{entróABase} \Rightarrow X \text{ modoAhorroOff})))$
3.  $\Box((\text{paredDelanteDetectado} \Rightarrow X \text{ gira1GradoIzquierda}) \wedge (\neg \text{paredDelanteDetectado} \Rightarrow \neg X \text{ gira1GradoIzquierda}))$

Observación:

- Es posible realizar un análisis análogo al que se llevó a cabo en el ejercicio 14.c y definir la propiedad sin la limitación del giro de la siguiente forma:

–  $\Box(\text{paredDelanteDetectado} \Rightarrow X \text{ gira1GradoIzquierda})$

## Ejercicio 20

Las propiedades utilizando LTL resultan de la siguiente forma:

1.  $\Box(c.\text{close} \Rightarrow X(\neg b.\text{entry} \cup c.\text{open}))$
2.  $\Box(c.\text{close} \Rightarrow X(\neg b.\text{exit} \cup c.\text{open}))$

Resultan mucho más claras usando LTL en ambos casos.

Al verificar con MTSA se observa que los resultados son consistentes con lo apreciado al trabajar con observadores.

**No LTL Property violations detected.**

Figure 2: No hay entradas luego del cierre.

```
Violation of LTL property: @NO_EXITS_AFTER_CLOSING
Trace to terminal set of states:
c.empty
c.open
a.entry
c.close
b.exit
c.empty
c.open
a.entry
a.entry
c.close
Cycle in terminal set:
b.exit
b.exit
c.empty
c.open
a.entry
a.entry
c.close
```

Figure 3: Hay salidas luego del cierre.