Lógica en Acción

Capítulo 6: Lógica y Acción

http://www.logicinaction.org/

Acciones

Diferentes tipos de acciones:

Acciones

Diferentes tipos de acciones:

- *Ella apaga la luz,*

Acciones

Diferentes tipos de acciones:

- *Ella apaga la luz,*
- *Guardas la leche en el refrigerador,*

Acciones

Diferentes tipos de acciones:

- *Ella apaga la luz*,
- *Guardas la leche en el refrigerador*,
- *La manzana cae al suelo*,

Acciones

Diferentes tipos de acciones:

- *Ella apaga la luz,*
- *Guardas la leche en el refrigerador,*
- *La manzana cae al suelo,*
- *Envío la solicitud después de llenarla,*

Acciones

Diferentes tipos de acciones:

- *Ella apaga la luz,*
- *Guardas la leche en el refrigerador,*
- *La manzana cae al suelo,*
- *Envío la solicitud después de llenarla,*
- *Él pregunta solo cuando sabe la respuesta,*

Acciones

Diferentes tipos de acciones:

- *Ella apaga la luz,*
- *Guardas la leche en el refrigerador,*
- *La manzana cae al suelo,*
- *Envío la solicitud después de llenarla,*
- *Él pregunta solo cuando sabe la respuesta,*
- *Ellos no hacen nada.*

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

- Después de *apagar la luz*, estaremos en la obscuridad.

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

- Después de *apagar la luz*, estaremos en la obscuridad.
- Después *de poner la leche en el refrigerador*, esta se enfriará.

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

- Después de *apagar la luz*, estaremos en la obscuridad.
- Después *de poner la leche en el refrigerador*, esta se enfriará.
- Una vez que *la manzana caiga al suelo* se empezara a descomponer.

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

- Después de *apagar la luz*, estaremos en la obscuridad.
- Después *de poner la leche en el refrigerador*, esta se enfriará.
- Una vez que *la manzana caiga al suelo* se empezara a descomponer.
- Normalmente el jurado recibirá la solicitud después de que *el solicitante la envíe*, pero algunas veces la solicitud puede perderse.

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

- Después de *apagar la luz*, estaremos en la obscuridad.
- Después *de poner la leche en el refrigerador*, esta se enfriará.
- Una vez que *la manzana caiga al suelo* se empezara a descomponer.
- Normalmente el jurado recibirá la solicitud después de que *el solicitante la envíe*, pero algunas veces la solicitud puede perderse.
- Después de que el profesor *hizo una pregunta*, los estudiantes permanecieron callados.

El efecto de una acción

Una acción pueden ser caracterizada de acuerdo a su efecto:

- Después de *apagar la luz*, estaremos en la obscuridad.
- Después *de poner la leche en el refrigerador*, esta se enfriará.
- Una vez que *la manzana caiga al suelo* se empezara a descomponer.
- Normalmente el jurado recibirá la solicitud después de que *el solicitante la envíe*, pero algunas veces la solicitud puede perderse.
- Después de que el profesor *hizo una pregunta*, los estudiantes permanecieron callados.
- Después de *no hacer nada*, todo sigue igual.

Operaciones sobre acciones

Las acciones pueden ser combinadas en diversas formas:

Operaciones sobre acciones

Las acciones pueden ser combinadas en diversas formas:

- **Composición.** Ejecuta una acción después de otra:

Vierte la mezcla sobre la carne y luego cubre el recipiente.

Operaciones sobre acciones

Las acciones pueden ser combinadas en diversas formas:

- **Composición.** Ejecuta una acción después de otra:

  Vierte la mezcla sobre la carne y luego cubre el recipiente.

- **Elección.** Elige entre dos acciones:

  Dame una de las cajas.

Operaciones sobre acciones

Las acciones pueden ser combinadas en diversas formas:

- **Composición.** Ejecuta una acción después de otra:

  Vierte la mezcla sobre la carne y luego cubre el recipiente.

- **Elección.** Elige entre dos acciones:

  Dame una de las cajas.

- **Repetición.** Realiza la misma acción varias veces:

  Bate hasta que el azúcar se integre.

Operaciones sobre acciones

Las acciones pueden ser combinadas en diversas formas:

- **Composición.** Ejecuta una acción después de otra:

  Vierte la mezcla sobre la carne y luego cubre el recipiente.

- **Elección.** Elige entre dos acciones:

  Dame una de las cajas.

- **Repetición.** Realiza la misma acción varias veces:

  Bate hasta que el azúcar se integre.

- **Verificación.** Verifica si cierta condición se cumple:

  Revisa si está lloviendo.

Operaciones sobre acciones

Las acciones pueden ser combinadas en diversas formas:

- **Composición.** Ejecuta una acción después de otra:

  Vierte la mezcla sobre la carne y luego cubre el recipiente.

- **Elección.** Elige entre dos acciones:

  Dame una de las cajas.

- **Repetición.** Realiza la misma acción varias veces:

  Bate hasta que el azúcar se integre.

- **Verificación.** Verifica si cierta condición se cumple:

  Revisa si está lloviendo.

- **Deshacer.** Deshacer una acción:

  Cierra la ventana que acabas de abrir.

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

1. **WHILE P do A**

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

1. **WHILE P do A**
   Puede ser definido como el repetir la verificación de 'P' y la ejecución de 'A', y luego verificar 'no A'.

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

1. **WHILE P do A**
   Puede ser definido como el repetir la verificación de 'P' y la ejecución de 'A', y luego verificar 'no A'.

2. **REPEAT A UNTIL P**

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

1. **WHILE P do A**
   Puede ser definido como el repetir la verificación de 'P' y la ejecución de 'A', y luego verificar 'no A'.

2. **REPEAT A UNTIL P**
   Puede ser definido como la composición de 'A' y luego **WHILE (not P) do A**.

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

1. **WHILE P do A**
   Puede ser definido como el repetir la verificación de 'P' y la ejecución de 'A', y luego verificar 'no A'.

2. **REPEAT A UNTIL P**
   Puede ser definido como la composición de 'A' y luego **WHILE (not P) do A**.

3. **IF P THEN A ELSE B**

Ejemplo: lenguajes de programación

Tres estructuras de control en lenguajes de programación:

1. **WHILE P do A**
   Puede ser definido como el repetir la verificación de 'P' y la ejecución de 'A', y luego verificar 'no A'.

2. **REPEAT A UNTIL P**
   Puede ser definido como la composición de 'A' y luego **WHILE (not P) do A**.

3. **IF P THEN A ELSE B**
   Puede ser definido como una elección entre verificar 'P' y luego ejecutar 'A', o verificar 'not P' y luego ejecutar 'B'.

Representando acciones de manera abstracta (1)

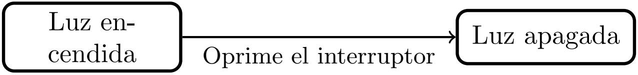
Las acciones pueden ser representadas como transiciones entre estados:

Representando acciones de manera abstracta (1)

Las acciones pueden ser representadas como transiciones entre estados:

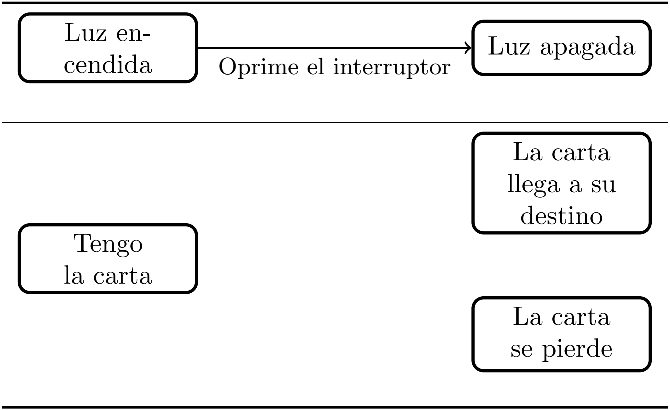
Representando acciones de manera abstracta (1)

Las acciones pueden ser representadas como transiciones entre estados:



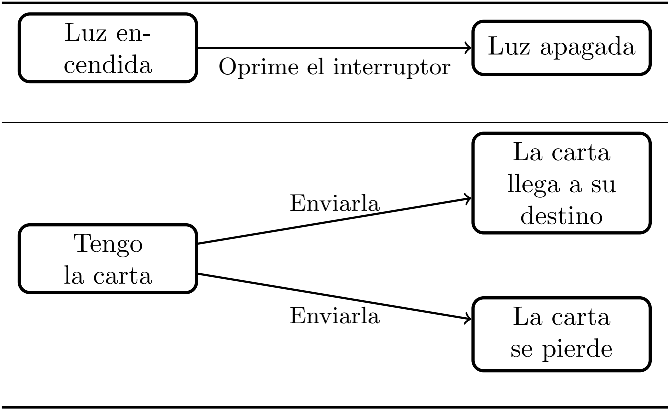
Representando acciones de manera abstracta (1)

Las acciones pueden ser representadas como transiciones entre estados:



Representando acciones de manera abstracta (1)

Las acciones pueden ser representadas como transiciones entre estados:

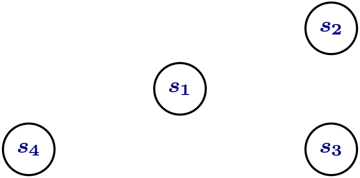


Representando acciones de manera abstracta (2)

Formalmente, si tomamos un conjunto de estados $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots\}$,

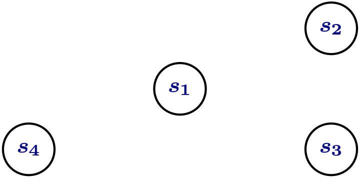
Representando acciones de manera abstracta (2)

Formalmente, si tomamos un conjunto de estados $S = \{s_1, s_2, \ldots\}$,



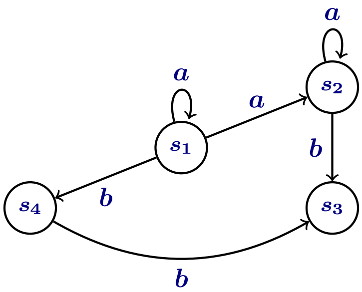
Representando acciones de manera abstracta (2)

Formalmente, si tomamos un conjunto de estados $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots\}$, podemos representar **acciones como relaciones binarias en $\boldsymbol{S}$**.



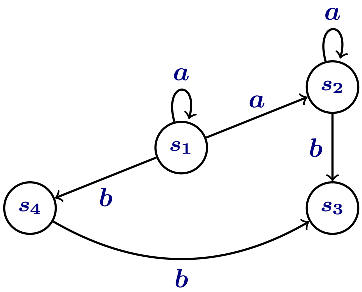
Representando acciones de manera abstracta (2)

Formalmente, si tomamos un conjunto de estados $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots\}$, podemos representar **acciones como relaciones binarias en $\boldsymbol{S}$**.



Representando acciones de manera abstracta (2)

Formalmente, si tomamos un conjunto de estados $S = \{s_1, s_2, \ldots\}$, podemos representar **acciones como relaciones binarias en S**.

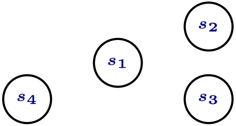


$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$

$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (1)

Sea S un conjunto $\{s_1, s_2, \ldots\}$.

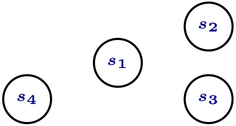


Operaciones sobre relaciones (1)

Sea S un conjunto $\{s_1, s_2, \ldots\}$.

- **La relación de identidad.**

$$I := \{(s, s) \mid s \in S\}$$

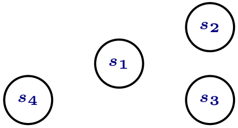


Operaciones sobre relaciones (1)

Sea S un conjunto $\{s_1, s_2, \ldots\}$.

- **La relación de identidad.**

$$I := \{(s, s) \mid s \in S\}$$



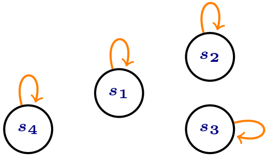
$$I = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$

Operaciones sobre relaciones (1)

Sea S un conjunto $\{s_1, s_2, \dots\}$.

- **La relación de identidad.**

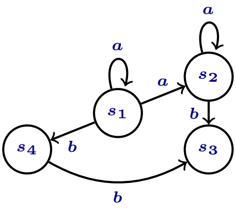
$$I := \{(s, s) \mid s \in S\}$$



$$I = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$

Operaciones sobre relaciones (2)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.



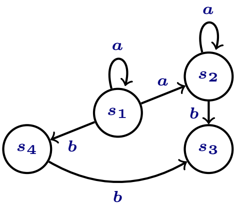
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (2)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Composición.**

$$R_a \circ R_b := \{(s, s') \mid \text{existe un } s'' \in S \text{ tal que } R_a s s'' \text{ y } R_b s'' s'\}$$



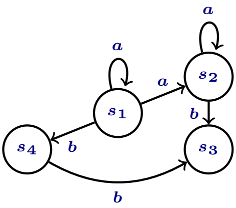
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (2)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Composición.**

$$R_a \circ R_b := \{(s, s') \mid \text{existe un } s'' \in S \text{ tal que } R_a s s'' \text{ y } R_b s'' s'\}$$



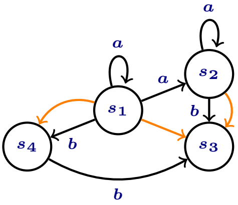
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_a \circ R_b = \{(s_1, s_4), (s_1, s_3), (s_2, s_3)\}$$

Operaciones sobre relaciones (2)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Composición.**

$$R_a \circ R_b := \{(s, s') \mid \text{existe un } s'' \in S \text{ tal que } R_a s s'' \text{ y } R_b s'' s'\}$$



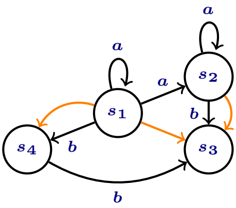
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_a \circ R_b = \{(s_1, s_4), (s_1, s_3), (s_2, s_3)\}$$

Operaciones sobre relaciones (2)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Composición.**

$$R_a \circ R_b := \{(s, s') \mid \text{existe un } s'' \in S \text{ tal que } R_a s s'' \text{ y } R_b s'' s'\}$$



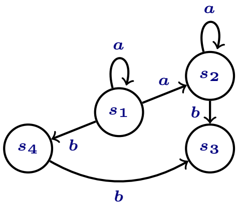
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_a \circ R_b = \{(s_1, s_4), (s_1, s_3), (s_2, s_3)\}$$

En particular, dada una relación R_a, definimos

$$R_a^0 := I, \qquad R_a^1 := R_a \circ R_a^0, \qquad R_a^2 := R_a \circ R_a^1, \qquad R_a^3 := R_a \circ R_a^2, \qquad \ldots$$

Operaciones sobre relaciones (3)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.



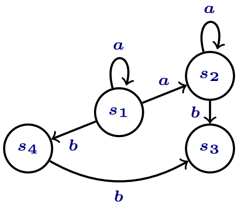
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (3)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Union.**

$$R_a \cup R_b := \{(s, s') \mid R_a ss' \text{ o } R_b ss'\}$$



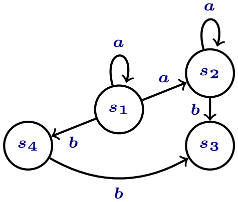
$$R_a := \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$$
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (3)

Sea S un conjunto $\{s_1, s_2, \dots\}$, y sean R_a, R_b relaciones binarias en S.

- **Union.**

$$R_a \cup R_b := \{(s, s') \mid R_a ss' \text{ o } R_b ss'\}$$



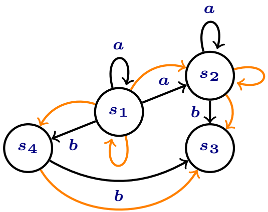
$$\begin{aligned} R_a &:= \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\} \\ R_b &:= \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\} \\ R_a \cup R_b &= \{(s_1, s_1), (s_1, s_2), (s_2, s_2) \\ &\quad (s_1, s_4), (s_2, s_3), (s_4, s_3)\} \end{aligned}$$

Operaciones sobre relaciones (3)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Union.**

$$R_a \cup R_b := \{(s, s') \mid R_a ss' \text{ o } R_b ss'\}$$

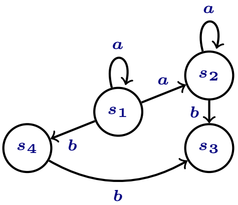


$$\begin{aligned} R_a &:= \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\} \\ R_b &:= \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\} \\ R_a \cup R_b &= \{(s_1, s_1), (s_1, s_2), (s_2, s_2) \\ &\quad (s_1, s_4), (s_2, s_3), (s_4, s_3)\} \end{aligned}$$

Operaciones sobre relaciones (4)

Sea $\boldsymbol{S}$ un conjunto $\{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots\}$, y sean $\boldsymbol{R_a}$, $\boldsymbol{R_b}$ relaciones binarias en $\boldsymbol{S}$.

$$\boldsymbol{R_b} := \{(\boldsymbol{s_1}, \boldsymbol{s_4}), (\boldsymbol{s_2}, \boldsymbol{s_3}), (\boldsymbol{s_4}, \boldsymbol{s_3})\}$$



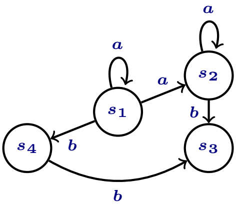
Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n s s' \text{ para algún } n \in \mathbb{N}\}$$

$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

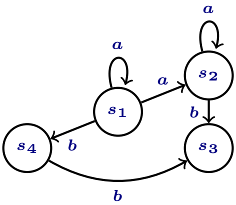


Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n ss' \text{ para algún } n \in \mathbb{N}\}$$



$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

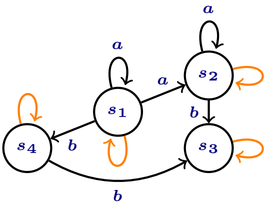
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n ss' \text{ para algún } n \in \mathbb{N}\}$$



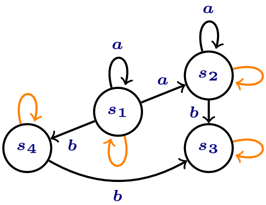
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n ss' \text{ para algún } n \in \mathbb{N}\}$$



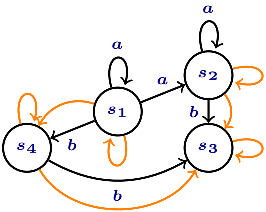
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$
$$R_b^1 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n s s' \text{ para algún } n \in \mathbb{N}\}$$



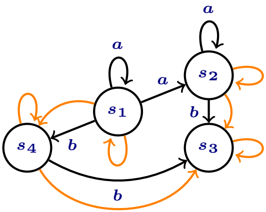
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$
$$R_b^1 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n s s' \text{ para algún } n \in \mathbb{N}\}$$



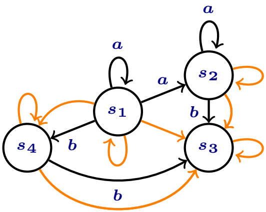
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$
$$R_b^1 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^2 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n ss' \text{ para algún } n \in \mathbb{N}\}$$



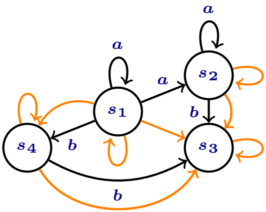
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$
$$R_b^1 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^2 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n ss' \text{ para algún } n \in \mathbb{N}\}$$



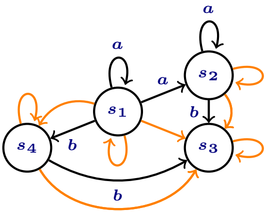
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$
$$R_b^1 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^2 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}$$
$$R_b^3 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

$$R_a^* := \{(s, s') \mid R_a^n s s' \text{ para algún } n \in \mathbb{N}\}$$



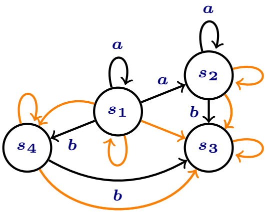
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^0 = \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\}$$
$$R_b^1 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$R_b^2 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}$$
$$R_b^3 = \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}$$
$$\vdots$$

Operaciones sobre relaciones (4)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Repetición cero o mas veces.**

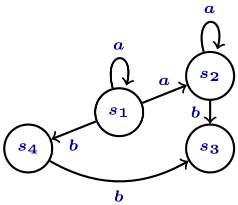
$$R_a^* := \{(s, s') \mid R_a^n s s' \text{ para algún } n \in \mathbb{N}\}$$



$$\begin{aligned}
R_b &:= \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\} \\
R_b^0 &= \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4)\} \\
R_b^1 &= \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\} \\
R_b^2 &= \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\} \\
R_b^3 &= \{(s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\} \\
&\vdots \\
R_b^* &= \{(s_1, s_1), (s_2, s_2), (s_3, s_3), (s_4, s_4), \\
&\qquad (s_1, s_4), (s_2, s_3), (s_4, s_3), (s_1, s_3)\}
\end{aligned}$$

Operaciones sobre relaciones (5)

Sea $\boldsymbol{S}$ un conjunto $\{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots\}$, y sean $\boldsymbol{R_a}$, $\boldsymbol{R_b}$ relaciones binarias en $\boldsymbol{S}$.



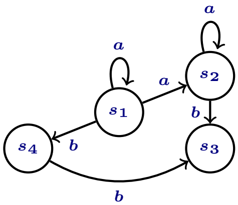
$$\boldsymbol{R_b} := \{(\boldsymbol{s_1}, \boldsymbol{s_4}), (\boldsymbol{s_2}, \boldsymbol{s_3}), (\boldsymbol{s_4}, \boldsymbol{s_3})\}$$

Operaciones sobre relaciones (5)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Opuesta.**

$$\breve{R_a} := \{(s', s) \mid R_a s s'\}$$



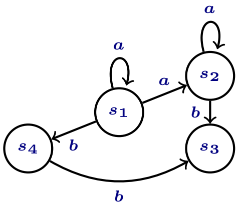
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$

Operaciones sobre relaciones (5)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Opuesta.**

$$\breve{R_a} := \{(s', s) \mid R_a s s'\}$$



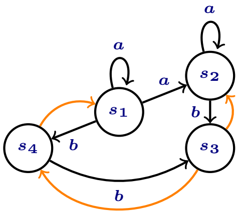
$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$\breve{R_b} = \{(s_4, s_1), (s_3, s_2), (s_3, s_4)\}$$

Operaciones sobre relaciones (5)

Sea S un conjunto $\{s_1, s_2, \ldots\}$, y sean R_a, R_b relaciones binarias en S.

- **Opuesta.**

$$\breve{R_a} := \{(s', s) \mid R_a s s'\}$$



$$R_b := \{(s_1, s_4), (s_2, s_3), (s_4, s_3)\}$$
$$\breve{R_b} = \{(s_4, s_1), (s_3, s_2), (s_3, s_4)\}$$

Sintaxis (1)

El lenguaje de la **lógica dinámica proposicional (LDP)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **fórmulas** se construyen mediante la siguiente regla.

Sintaxis (1)

El lenguaje de la **lógica dinámica proposicional (*LDP*)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **fórmulas** se construyen mediante la siguiente regla.
  - Todo enunciado básico es una fórmula

$$\boldsymbol{p}, \quad \boldsymbol{q}, \quad \boldsymbol{r}, \quad \ldots$$

Sintaxis (1)

El lenguaje de la **lógica dinámica proposicional (*LDP*)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **fórmulas** se construyen mediante la siguiente regla.
  - Todo enunciado básico es una fórmula

$$p, \quad q, \quad r, \quad \ldots$$

  - Si φ y ψ son fórmulas, también lo son:

$$\neg\varphi, \quad \varphi \wedge \psi, \quad \varphi \vee \psi, \quad \varphi \to \psi, \quad \varphi \leftrightarrow \psi$$

Sintaxis (1)

El lenguaje de la **lógica dinámica proposicional (*LDP*)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **fórmulas** se construyen mediante la siguiente regla.
  - Todo enunciado básico es una fórmula

$$p, \quad q, \quad r, \quad \ldots$$

  - Si φ y ψ son fórmulas, también lo son:

$$\neg\varphi, \quad \varphi \wedge \psi, \quad \varphi \vee \psi, \quad \varphi \to \psi, \quad \varphi \leftrightarrow \psi$$

  - Si φ es una fórmula y α una acción, la siguiente es una fórmula:

$$\langle\alpha\rangle\,\varphi$$

Sintaxis (2)

El lenguaje de la **lógica dinámica proposicional (LDP)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **acciones** se construyen mediante las siguientes reglas

Sintaxis (2)

El lenguaje de la **lógica dinámica proposicional (*LDP*)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **acciones** se construyen mediante las siguientes reglas
  - Toda acción básica es una acción

$$\boldsymbol{a}, \quad \boldsymbol{b}, \quad \boldsymbol{c}, \quad \ldots$$

Sintaxis (2)

El lenguaje de la **lógica dinámica proposicional (LDP)** tiene dos partes, **fórmulas** φ y **acciones** α.

- Las **acciones** se construyen mediante las siguientes reglas
  - Toda acción básica es una acción

$$a, \quad b, \quad c, \quad \ldots$$

  - Si α y β son acciones, también lo son:

$$\alpha;\beta\,, \quad \alpha \cup \beta\,, \quad \alpha^*$$

Sintaxis (2)

El lenguaje de la **lógica dinámica proposicional (*LDP*)** tiene dos partes, **fórmulas** φ y **acciones** $\boldsymbol{\alpha}$.

- Las **acciones** se construyen mediante las siguientes reglas
  - Toda acción básica es una acción
    $$\boldsymbol{a}, \quad \boldsymbol{b}, \quad \boldsymbol{c}, \quad \ldots$$
  - Si $\boldsymbol{\alpha}$ y $\boldsymbol{\beta}$ son acciones, también lo son:
    $$\boldsymbol{\alpha};\boldsymbol{\beta}\,, \quad \boldsymbol{\alpha} \cup \boldsymbol{\beta}\,, \quad \boldsymbol{\alpha}^*$$
  - Si φ es una fórmula la siguiente es una acción:
    $$?\varphi$$

Intuiciones y abreviaciones

$\boldsymbol{\alpha};\boldsymbol{\beta}$
$\boldsymbol{\alpha}\cup\boldsymbol{\beta}$
$\boldsymbol{\alpha}^{*}$
$?\boldsymbol{\varphi}$
$\langle\boldsymbol{\alpha}\rangle\,\boldsymbol{\varphi}$

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	
α^*	
$?\varphi$	
$\langle\alpha\rangle\,\varphi$	

Intuiciones y abreviaciones

$\alpha ; \beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	
$?\varphi$	
$\langle \alpha \rangle \varphi$	

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	
$\langle\alpha\rangle\,\varphi$	

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	**verificación**: verifica si φ es verdadera o no.
$\langle\alpha\rangle\,\varphi$	

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	**verificación**: verifica si φ es verdadera o no.
$\langle\alpha\rangle\,\varphi$	α puede ser ejecutada de forma que, después de hacerlo, φ es verdadera.

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	**verificación**: verifica si φ es verdadera o no.
$\langle\alpha\rangle\,\varphi$	α puede ser ejecutada de forma que, después de hacerlo, φ es verdadera.

Abreviaremos $p \vee \neg p$ como $\top$.

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	**verificación**: verifica si φ es verdadera o no.
$\langle\alpha\rangle\,\varphi$	α puede ser ejecutada de forma que, después de hacerlo, φ es verdadera.

Abreviaremos $p \vee \neg p$ como $\top$.

Abreviaremos $\neg\top$ como $\bot$.

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	**verificación**: verifica si φ es verdadera o no.
$\langle\alpha\rangle\,\varphi$	α puede ser ejecutada de forma que, después de hacerlo, φ es verdadera.

Abreviaremos $p \vee \neg p$ como $\top$.

Abreviaremos $\neg\top$ como $\bot$.

Abreviaremos $\neg\langle\alpha\rangle\,\neg\varphi$ como $[\alpha]\,\varphi$.

Intuiciones y abreviaciones

$\alpha;\beta$	**composición**: ejecuta α y luego β.
$\alpha \cup \beta$	**elección no determinista**: ejecuta α o β.
α^*	**repetición**: ejecuta α cero, una, o cualquier número *finito* de veces.
$?\varphi$	**verificación**: verifica si φ es verdadera o no.
$\langle\alpha\rangle\,\varphi$	α puede ser ejecutada de forma que, después de hacerlo, φ es verdadera.

Abreviaremos $p \vee \neg p$ como $\top$.

Abreviaremos $\neg\top$ como $\bot$.

Abreviaremos $\neg\langle\alpha\rangle\,\neg\varphi$ como $[\alpha]\,\varphi$.

$[\alpha]\,\varphi$	Después de cualquier ejecución de α, φ es verdadera.

Ejemplos de fórmulas

$$\langle\boldsymbol{\alpha}\rangle\,\top$$

$$[\boldsymbol{\alpha}]\,\bot$$

$$\langle\boldsymbol{\alpha}\rangle\,\varphi\wedge\neg[\boldsymbol{\alpha}]\,\varphi$$

Ejemplos de fórmulas

$\langle \alpha \rangle \top$	α puede ser ejecutada.
$[\alpha] \bot$	
$\langle \alpha \rangle \varphi \land \neg [\alpha] \varphi$	

Ejemplos de fórmulas

$\langle \boldsymbol{\alpha} \rangle \top$	$\boldsymbol{\alpha}$ puede ser ejecutada.
$[\boldsymbol{\alpha}] \bot$	$\boldsymbol{\alpha}$ no puede ser ejecutada.
$\langle \boldsymbol{\alpha} \rangle \varphi \land \neg [\boldsymbol{\alpha}] \varphi$	

Ejemplos de fórmulas

$\langle\boldsymbol{\alpha}\rangle\,\top$	$\boldsymbol{\alpha}$ puede ser ejecutada.
$[\boldsymbol{\alpha}]\,\bot$	$\boldsymbol{\alpha}$ no puede ser ejecutada.
$\langle\boldsymbol{\alpha}\rangle\,\varphi \wedge \neg[\boldsymbol{\alpha}]\,\varphi$	$\boldsymbol{\alpha}$ puede ser ejecutada en al menos dos formas diferentes.

Los modelos (1)

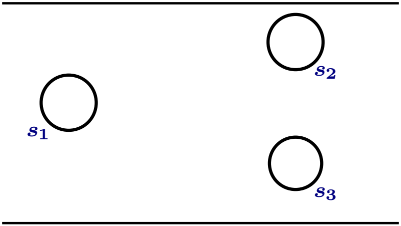
Las estructuras en las cuales evaluamos fórmulas en LDP, **sistemas de transición etiquetados (*STE*)**, tienen tres componentes:

$$\boldsymbol{M} = \langle \quad , \quad\quad , \quad \rangle$$

Los modelos (1)

Las estructuras en las cuales evaluamos fórmulas en LDP, **sistemas de transición etiquetados (*STE*)**, tienen tres componentes:

- S: un conjunto no vació de **estados**,

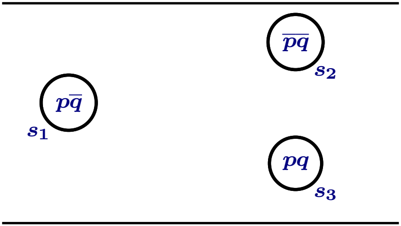


$$M = \langle S, \quad , \quad \rangle$$

Los modelos (1)

Las estructuras en las cuales evaluamos fórmulas en LDP, **sistemas de transición etiquetados (*STE*)**, tienen tres componentes:

- $\boldsymbol{S}$: un conjunto no vació de **estados**,
- $\boldsymbol{V}$: una **valuación** indicando qué enunciados básicos son verdaderos en cada estado $\boldsymbol{s} \in \boldsymbol{S}$, y

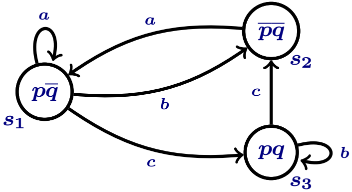


$$\boldsymbol{M} = \langle \boldsymbol{S}, \quad , \boldsymbol{V} \rangle$$

Los modelos (1)

Las estructuras en las cuales evaluamos fórmulas en LDP, **sistemas de transición etiquetados (*STE*)**, tienen tres componentes:

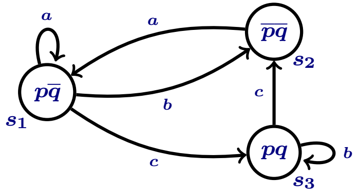
- $\boldsymbol{S}$: un conjunto no vació de **estados**,
- $\boldsymbol{V}$: una **valuación** indicando qué enunciados básicos son verdaderos en cada estado $\boldsymbol{s} \in \boldsymbol{S}$, y
- $\boldsymbol{R_a}$: una **relación binaria** por cada acción básica $\boldsymbol{a}$.



$$\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$$

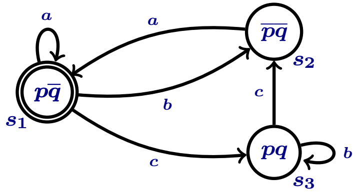
The models (2)

A un **sistema de transición etiquetado** en el cual distinguimos un estado (el estado *inicial*) se le llama a **sistema de transición etiquetado marcado** o **gráfica de procesos**.



The models (2)

A un **sistema de transición etiquetado** en el cual distinguimos un estado (el estado *inicial*) se le llama a **sistema de transición etiquetado marcado** o **gráfica de procesos**.



Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

$$(\boldsymbol{M}, \boldsymbol{s}) \models \boldsymbol{p} \qquad \text{si y solo si} \quad \boldsymbol{p} \in \boldsymbol{V}(\boldsymbol{s})$$

Evaluando fórmulas

Sea (M, s) un sistema de transición etiquetado marcado, con $M = \langle S, R_a, V \rangle$:

$$
\begin{array}{lll}
(M, s) \models p & \text{si y solo si} & p \in V(s) \\
(M, s) \models \neg\varphi & \text{si y solo si} & \text{no es el caso que } (M, s) \models \varphi
\end{array}
$$

Evaluando fórmulas

Sea (M, s) un sistema de transición etiquetado marcado, con $M = \langle S, R_a, V \rangle$:

$$
\begin{array}{lll}
(M, s) \models p & \text{si y solo si} & p \in V(s) \\
(M, s) \models \neg\varphi & \text{si y solo si} & \text{no es el caso que } (M, s) \models \varphi \\
(M, s) \models \varphi \vee \psi & \text{si y solo si} & (M, s) \models \varphi \text{ o } (M, s) \models \psi
\end{array}
$$

Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

$(\boldsymbol{M}, \boldsymbol{s}) \models \boldsymbol{p}$	si y solo si	$\boldsymbol{p} \in \boldsymbol{V}(\boldsymbol{s})$
$(\boldsymbol{M}, \boldsymbol{s}) \models \neg\varphi$	si y solo si	no es el caso que $(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$
$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi \vee \psi$	si y solo si	$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$ o $(\boldsymbol{M}, \boldsymbol{s}) \models \psi$
...	si y solo si	...

Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

$(\boldsymbol{M}, \boldsymbol{s}) \models \boldsymbol{p}$	si y solo si	$\boldsymbol{p} \in \boldsymbol{V}(\boldsymbol{s})$
$(\boldsymbol{M}, \boldsymbol{s}) \models \neg\varphi$	si y solo si	no es el caso que $(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$
$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi \vee \psi$	si y solo si	$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$ o $(\boldsymbol{M}, \boldsymbol{s}) \models \psi$
...	si y solo si	...
$(\boldsymbol{M}, \boldsymbol{s}) \models \langle\alpha\rangle\varphi$	si y solo si	existe un $\boldsymbol{t} \in \boldsymbol{S}$ tal que $\boldsymbol{R_\alpha st}$ y $(\boldsymbol{M}, \boldsymbol{t}) \models \varphi$

Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

$(\boldsymbol{M}, \boldsymbol{s}) \models \boldsymbol{p}$	si y solo si	$\boldsymbol{p} \in \boldsymbol{V}(\boldsymbol{s})$
$(\boldsymbol{M}, \boldsymbol{s}) \models \neg\varphi$	si y solo si	no es el caso que $(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$
$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi \vee \psi$	si y solo si	$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$ o $(\boldsymbol{M}, \boldsymbol{s}) \models \psi$
...	si y solo si	...
$(\boldsymbol{M}, \boldsymbol{s}) \models \langle\alpha\rangle\,\varphi$	si y solo si	existe un $\boldsymbol{t} \in \boldsymbol{S}$ tal que $\boldsymbol{R_\alpha st}$ y $(\boldsymbol{M}, \boldsymbol{t}) \models \varphi$

Si $\boldsymbol{\alpha}$ no es una acción básica, la relación $\boldsymbol{R_\alpha}$ se define como

Evaluando fórmulas

Sea (M, s) un sistema de transición etiquetado marcado, con $M = \langle S, R_a, V \rangle$:

$(M, s) \models p$	si y solo si	$p \in V(s)$
$(M, s) \models \neg\varphi$	si y solo si	no es el caso que $(M, s) \models \varphi$
$(M, s) \models \varphi \vee \psi$	si y solo si	$(M, s) \models \varphi$ o $(M, s) \models \psi$
...	si y solo si	...
$(M, s) \models \langle\alpha\rangle\,\varphi$	si y solo si	existe un $t \in S$ tal que $R_\alpha st$ y $(M, t) \models \varphi$

Si α no es una acción básica, la relación R_α se define como

$$R_{\alpha;\beta} := R_\alpha \circ R_\beta$$

Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

$(\boldsymbol{M}, \boldsymbol{s}) \models \boldsymbol{p}$	si y solo si	$\boldsymbol{p} \in \boldsymbol{V}(\boldsymbol{s})$
$(\boldsymbol{M}, \boldsymbol{s}) \models \neg\varphi$	si y solo si	no es el caso que $(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$
$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi \vee \psi$	si y solo si	$(\boldsymbol{M}, \boldsymbol{s}) \models \varphi$ o $(\boldsymbol{M}, \boldsymbol{s}) \models \psi$
...	si y solo si	...
$(\boldsymbol{M}, \boldsymbol{s}) \models \langle \alpha \rangle \varphi$	si y solo si	existe un $\boldsymbol{t} \in \boldsymbol{S}$ tal que $\boldsymbol{R}_{\alpha} \boldsymbol{st}$ y $(\boldsymbol{M}, \boldsymbol{t}) \models \varphi$

Si $\boldsymbol{\alpha}$ no es una acción básica, la relación $\boldsymbol{R}_{\alpha}$ se define como

$$\boldsymbol{R}_{\alpha;\beta} := \boldsymbol{R}_{\alpha} \circ \boldsymbol{R}_{\beta}$$
$$\boldsymbol{R}_{\alpha\cup\beta} := \boldsymbol{R}_{\alpha} \cup \boldsymbol{R}_{\beta}$$

Evaluando fórmulas

Sea $(\boldsymbol{M}, \boldsymbol{s})$ un sistema de transición etiquetado marcado, con $\boldsymbol{M} = \langle \boldsymbol{S}, \boldsymbol{R_a}, \boldsymbol{V} \rangle$:

$$
\begin{array}{lll}
(\boldsymbol{M}, \boldsymbol{s}) \models \boldsymbol{p} & \text{si y solo si} & \boldsymbol{p} \in \boldsymbol{V}(\boldsymbol{s}) \\
(\boldsymbol{M}, \boldsymbol{s}) \models \neg \varphi & \text{si y solo si} & \text{no es el caso que } (\boldsymbol{M}, \boldsymbol{s}) \models \varphi \\
(\boldsymbol{M}, \boldsymbol{s}) \models \varphi \vee \psi & \text{si y solo si} & (\boldsymbol{M}, \boldsymbol{s}) \models \varphi \;\text{ o }\; (\boldsymbol{M}, \boldsymbol{s}) \models \psi \\
\ldots & \text{si y solo si} & \ldots \\
(\boldsymbol{M}, \boldsymbol{s}) \models \langle \alpha \rangle \varphi & \text{si y solo si} & \text{existe un } \boldsymbol{t} \in \boldsymbol{S} \text{ tal que } \boldsymbol{R_\alpha st} \text{ y } (\boldsymbol{M}, \boldsymbol{t}) \models \varphi
\end{array}
$$

Si $\boldsymbol{\alpha}$ no es una acción básica, la relación $\boldsymbol{R_\alpha}$ se define como

$$
\begin{aligned}
\boldsymbol{R_{\alpha;\beta}} &:= \boldsymbol{R_\alpha} \circ \boldsymbol{R_\beta} \\
\boldsymbol{R_{\alpha\cup\beta}} &:= \boldsymbol{R_\alpha} \cup \boldsymbol{R_\beta} \\
\boldsymbol{R_{\alpha^*}} &:= (\boldsymbol{R_\alpha})^*
\end{aligned}
$$

Evaluando fórmulas

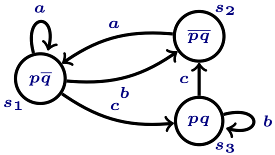
Sea (M, s) un sistema de transición etiquetado marcado, con $M = \langle S, R_a, V \rangle$:

$$
\begin{array}{lll}
(M, s) \models p & \text{si y solo si} & p \in V(s) \\
(M, s) \models \neg\varphi & \text{si y solo si} & \text{no es el caso que } (M, s) \models \varphi \\
(M, s) \models \varphi \vee \psi & \text{si y solo si} & (M, s) \models \varphi \ \text{ o } \ (M, s) \models \psi \\
\ldots & \text{si y solo si} & \ldots \\
(M, s) \models \langle\alpha\rangle\, \varphi & \text{si y solo si} & \text{existe un } t \in S \text{ tal que } R_\alpha st \text{ y } (M, t) \models \varphi
\end{array}
$$

Si α no es una acción básica, la relación R_α se define como

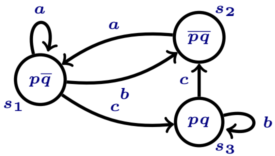
$$
\begin{aligned}
R_{\alpha;\beta} &:= R_\alpha \circ R_\beta \\
R_{\alpha\cup\beta} &:= R_\alpha \cup R_\beta \\
R_{\alpha^*} &:= (R_\alpha)^* \\
R_{?\varphi} &:= \{(s, s) \in S \times S \mid (M, s) \models \varphi\}
\end{aligned}
$$

Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

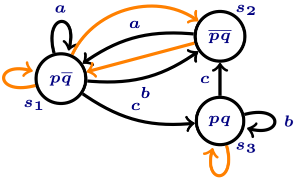
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\} \\
R_b &:= \{(s_1, s_2), (s_3, s_3)\} \\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

$$R_{a \cup b} =$$

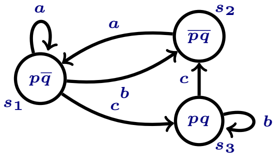
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a \cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$

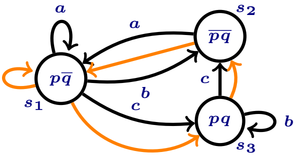
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a\cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$
$$R_{a\cup c} =$$

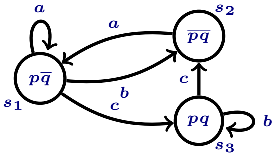
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\} \\
R_b &:= \{(s_1, s_2), (s_3, s_3)\} \\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

$$
\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

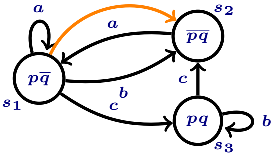
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a \cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$
$$R_{a \cup c} = \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}$$
$$R_{c;c} =$$

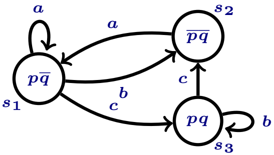
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a\cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$
$$R_{a\cup c} = \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}$$
$$R_{c;c} = \{(s_1, s_2)\}$$

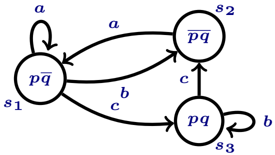
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a\cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$
$$R_{a\cup c} = \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}$$
$$R_{c;c} = \{(s_1, s_2)\}$$
$$R_{b;b} =$$

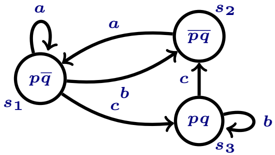
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a \cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$
$$R_{a \cup c} = \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}$$
$$R_{c;c} = \{(s_1, s_2)\}$$
$$R_{b;b} = \{\ \}$$

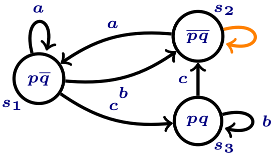
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$R_{a\cup b} = \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}$$
$$R_{a\cup c} = \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}$$
$$R_{c;c} = \{(s_1, s_2)\}$$
$$R_{b;b} = \{\ \}$$
$$R_{?\neg(p\vee q)} =$$

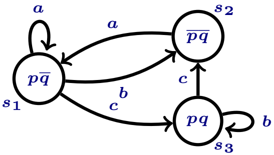
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\}\\
R_b &:= \{(s_1, s_2), (s_3, s_3)\}\\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

$$
\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}\\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}\\
R_{c;c} &= \{(s_1, s_2)\}\\
R_{b;b} &= \{\ \}\\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\}
\end{aligned}
$$

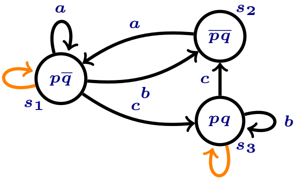
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\}\\
R_b &:= \{(s_1, s_2), (s_3, s_3)\}\\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

$$
\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}\\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}\\
R_{c;c} &= \{(s_1, s_2)\}\\
R_{b;b} &= \{\ \}\\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\}\\
R_{?(p\vee q)} &=
\end{aligned}
$$

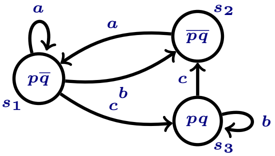
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\} \\
R_{c;c} &= \{(s_1, s_2)\} \\
R_{b;b} &= \{\ \} \\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\} \\
R_{?(p\vee q)} &= \{(s_1, s_1), (s_3, s_3)\}
\end{aligned}$$

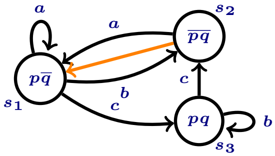
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\} \\
R_{c;c} &= \{(s_1, s_2)\} \\
R_{b;b} &= \{\ \} \\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\} \\
R_{?(p\vee q)} &= \{(s_1, s_1), (s_3, s_3)\} \\
R_{?\neg(p\vee q);a;?(p\vee q)} &=
\end{aligned}$$

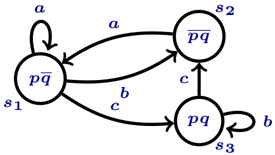
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\} \\
R_b &:= \{(s_1, s_2), (s_3, s_3)\} \\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

$$
\begin{aligned}
R_{a \cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a \cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\} \\
R_{c;c} &= \{(s_1, s_2)\} \\
R_{b;b} &= \{\ \} \\
R_{?\neg(p \vee q)} &= \{(s_2, s_2)\} \\
R_{?(p \vee q)} &= \{(s_1, s_1), (s_3, s_3)\} \\
R_{?\neg(p \vee q);a;?(p \vee q)} &= \{(s_2, s_1)\}
\end{aligned}
$$

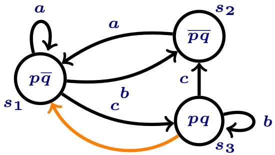
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$\begin{aligned}
R_{a \cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a \cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\} \\
R_{c;c} &= \{(s_1, s_2)\} \\
R_{b;b} &= \{\ \} \\
R_{?\neg(p \vee q)} &= \{(s_2, s_2)\} \\
R_{?(p \vee q)} &= \{(s_1, s_1), (s_3, s_3)\} \\
R_{?\neg(p \vee q);a;?(p \vee q)} &= \{(s_2, s_1)\} \\
R_{c;a} &=
\end{aligned}$$

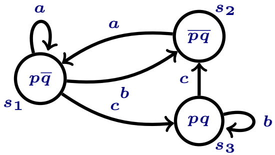
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$
\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}\\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}\\
R_{c;c} &= \{(s_1, s_2)\}\\
R_{b;b} &= \{\ \}\\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\}\\
R_{?(p\vee q)} &= \{(s_1, s_1), (s_3, s_3)\}\\
R_{?\neg(p\vee q);a;?(p\vee q)} &= \{(s_2, s_1)\}\\
R_{c;a} &= \{(s_3, s_1)\}
\end{aligned}
$$

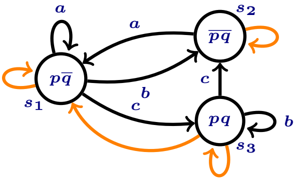
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\}\\
R_b &:= \{(s_1, s_2), (s_3, s_3)\}\\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

$$
\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\}\\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\}\\
R_{c;c} &= \{(s_1, s_2)\}\\
R_{b;b} &= \{\ \}\\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\}\\
R_{?(p\vee q)} &= \{(s_1, s_1), (s_3, s_3)\}\\
R_{?\neg(p\vee q);a;?(p\vee q)} &= \{(s_2, s_1)\}\\
R_{c;a} &= \{(s_3, s_1)\}\\
R_{(c;a)^*} &=
\end{aligned}
$$

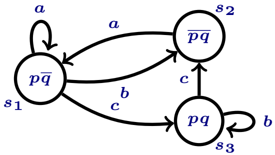
Ejemplo: construyendo relaciones mas complejas



$$R_a := \{(s_1, s_1), (s_2, s_1)\}$$
$$R_b := \{(s_1, s_2), (s_3, s_3)\}$$
$$R_c := \{(s_1, s_3), (s_3, s_2)\}$$

$$\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\} \\
R_{c;c} &= \{(s_1, s_2)\} \\
R_{b;b} &= \{\ \} \\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\} \\
R_{?(p\vee q)} &= \{(s_1, s_1), (s_3, s_3)\} \\
R_{?\neg(p\vee q);a;?(p\vee q)} &= \{(s_2, s_1)\} \\
R_{c;a} &= \{(s_3, s_1)\} \\
R_{(c;a)^*} &= \{(s_3, s_1), (s_1, s_1), (s_2, s_2), (s_3, s_3)\}
\end{aligned}$$

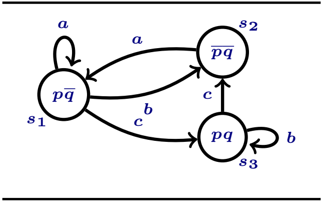
Ejemplo: construyendo relaciones mas complejas



$$
\begin{aligned}
R_a &:= \{(s_1, s_1), (s_2, s_1)\} \\
R_b &:= \{(s_1, s_2), (s_3, s_3)\} \\
R_c &:= \{(s_1, s_3), (s_3, s_2)\}
\end{aligned}
$$

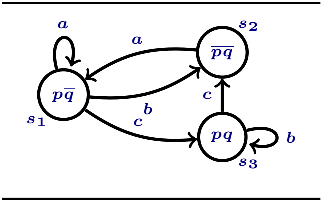
$$
\begin{aligned}
R_{a\cup b} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_2), (s_3, s_3)\} \\
R_{a\cup c} &= \{(s_1, s_1), (s_2, s_1), (s_1, s_3), (s_3, s_2)\} \\
R_{c;c} &= \{(s_1, s_2)\} \\
R_{b;b} &= \{\ \} \\
R_{?\neg(p\vee q)} &= \{(s_2, s_2)\} \\
R_{?(p\vee q)} &= \{(s_1, s_1), (s_3, s_3)\} \\
R_{?\neg(p\vee q);a;?(p\vee q)} &= \{(s_2, s_1)\} \\
R_{c;a} &= \{(s_3, s_1)\} \\
R_{(c;a)^*} &= \{(s_3, s_1), (s_1, s_1), (s_2, s_2), (s_3, s_3)\}
\end{aligned}
$$

Ejemplo: evaluando fórmulas



M

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \wedge \neg [a \cup b]\, p$?

$(M, s_1) \models [b]\, \bot$?

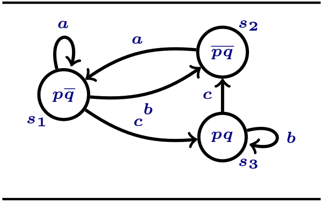
$(M, s_2) \models \langle a \rangle \top \rightarrow \langle b \rangle \top$?

$(M, s_2) \models \langle c^* \rangle \top$?

$(M, s_3) \models [(c; a)^*]\, p$?

$(M, s_3) \models [?p]\, p$?

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \wedge \neg [a \cup b]\, p$ ✓

$(M, s_1) \models [b]\, \bot$?

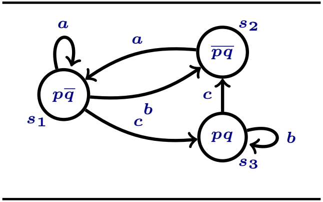
$(M, s_2) \models \langle a \rangle \top \rightarrow \langle b \rangle \top$?

$(M, s_2) \models \langle c^* \rangle \top$?

$(M, s_3) \models [(c; a)^*]\, p$?

$(M, s_3) \models [?p]\, p$?

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \land \neg [a \cup b]\, p$ ✓

$(M, s_1) \models [b]\, \bot$ ✗

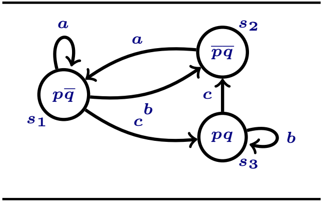
$(M, s_2) \models \langle a \rangle\, \top \rightarrow \langle b \rangle\, \top$?

$(M, s_2) \models \langle c^* \rangle\, \top$?

$(M, s_3) \models [(c; a)^*]\, p$?

$(M, s_3) \models [?p]\, p$?

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \wedge \neg [a \cup b]\, p$ ✓

$(M, s_1) \models [b] \perp$ ✗

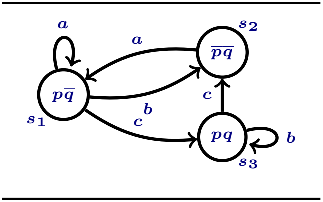
$(M, s_2) \models \langle a \rangle \top \rightarrow \langle b \rangle \top$ ✗

$(M, s_2) \models \langle c^* \rangle \top$?

$(M, s_3) \models [(c; a)^*]\, p$?

$(M, s_3) \models [?p]\, p$?

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \wedge \neg[a \cup b]\, p$ ✓

$(M, s_1) \models [b]\, \bot$ ✗

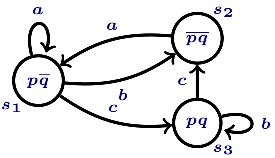
$(M, s_2) \models \langle a \rangle \top \rightarrow \langle b \rangle \top$ ✗

$(M, s_2) \models \langle c^* \rangle \top$ ✓

$(M, s_3) \models [(c; a)^*]\, p$?

$(M, s_3) \models [?p]\, p$?

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \wedge \neg [a \cup b]\, p$ ✓

$(M, s_1) \models [b]\, \bot$ ✗

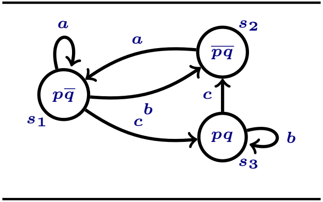
$(M, s_2) \models \langle a \rangle \top \rightarrow \langle b \rangle \top$ ✗

$(M, s_2) \models \langle c^* \rangle \top$ ✓

$(M, s_3) \models [(c; a)^*]\, p$ ✓

$(M, s_3) \models [?p]\, p$?

Ejemplo: evaluando fórmulas



M

$(M, s_1) \models \langle a \cup b \rangle\, p \wedge \neg [a \cup b]\, p$ ✓

$(M, s_1) \models [b]\, \bot$ ✗

$(M, s_2) \models \langle a \rangle\, \top \rightarrow \langle b \rangle\, \top$ ✗

$(M, s_2) \models \langle c^* \rangle\, \top$ ✓

$(M, s_3) \models [(c; a)^*]\, p$ ✓

$(M, s_3) \models [?p]\, p$ ✓

Sistema de derivación (1)

Las fórmulas válidas de ***LDP*** pueden ser derivadas a partir de los siguientes principios:

Sistema de derivación (1)

Las fórmulas válidas de ***LDP*** pueden ser derivadas a partir de los siguientes principios:

1. Todas las tautologías proposicionales.

Sistema de derivación (1)

Las fórmulas válidas de ***LDP*** pueden ser derivadas a partir de los siguientes principios:

1. Todas las tautologías proposicionales.
2. $[\alpha]\,(\varphi \to \psi) \to ([\alpha]\,\varphi \to [\alpha]\,\psi)$ para cualquier acción α.

Sistema de derivación (1)

Las fórmulas válidas de ***LDP*** pueden ser derivadas a partir de los siguientes principios:

1. Todas las tautologías proposicionales.
2. $[\alpha]\,(\varphi \rightarrow \psi) \rightarrow ([\alpha]\,\varphi \rightarrow [\alpha]\,\psi)$ para cualquier acción α.
3. **Modus ponens** (MP): si tienes φ y $\varphi \rightarrow \psi$, entonces deriva ψ.

Sistema de derivación (1)

Las fórmulas válidas de ***LDP*** pueden ser derivadas a partir de los siguientes principios:

1. Todas las tautologías proposicionales.
2. $[\alpha](\varphi \to \psi) \to ([\alpha]\varphi \to [\alpha]\psi)$ para cualquier acción α.
3. **Modus ponens** (MP): si tienes φ y $\varphi \to \psi$, entonces deriva ψ.
4. **Necessitation** (Nec): si tienes φ entonces deriva $[\alpha]\varphi$ para cualquier acción α.

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones**:

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**

$$[?\psi]\,\varphi \leftrightarrow (\psi \rightarrow \varphi)$$

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**
$$[?\psi]\,\varphi \leftrightarrow (\psi \rightarrow \varphi)$$
   - **Composición:**
$$[\alpha;\beta]\,\varphi \leftrightarrow [\alpha]\,[\beta]\,\varphi$$

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**
     $$[?\psi]\,\varphi \leftrightarrow (\psi \rightarrow \varphi)$$
   - **Composición:**
     $$[\alpha;\beta]\,\varphi \leftrightarrow [\alpha]\,[\beta]\,\varphi$$
   - **Elección:**
     $$[\alpha \cup \beta]\,\varphi \leftrightarrow ([\alpha]\,\varphi \wedge [\beta]\,\varphi)$$

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**
   $$[?\psi]\,\varphi \leftrightarrow (\psi \rightarrow \varphi)$$
   - **Composición:**
   $$[\alpha;\beta]\,\varphi \leftrightarrow [\alpha]\,[\beta]\,\varphi$$
   - **Elección:**
   $$[\alpha \cup \beta]\,\varphi \leftrightarrow ([\alpha]\,\varphi \wedge [\beta]\,\varphi)$$
   - **Repetición:**

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**
     $$[?\psi]\,\varphi \leftrightarrow (\psi \rightarrow \varphi)$$
   - **Composición:**
     $$[\alpha;\beta]\,\varphi \leftrightarrow [\alpha]\,[\beta]\,\varphi$$
   - **Elección:**
     $$[\alpha \cup \beta]\,\varphi \leftrightarrow ([\alpha]\,\varphi \wedge [\beta]\,\varphi)$$
   - **Repetición:**
     - **Combinación:**
       $$[\alpha^*]\,\varphi \leftrightarrow (\varphi \wedge [\alpha]\,[\alpha^*]\,\varphi)$$

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**
     $$[?\psi]\,\varphi \leftrightarrow (\psi \to \varphi)$$
   - **Composición:**
     $$[\alpha;\beta]\,\varphi \leftrightarrow [\alpha]\,[\beta]\,\varphi$$
   - **Elección:**
     $$[\alpha \cup \beta]\,\varphi \leftrightarrow ([\alpha]\,\varphi \wedge [\beta]\,\varphi)$$
   - **Repetición:**
     - **Combinación:**
       $$[\alpha^*]\,\varphi \leftrightarrow (\varphi \wedge [\alpha]\,[\alpha^*]\,\varphi)$$
     - **Inducción:**
       $$\Big(\varphi \wedge [\alpha^*]\,(\varphi \to [\alpha]\,\varphi)\Big) \to [\alpha^*]\,\varphi$$

Sistema de derivación (2)

5. **Principios para operaciones sobre acciones:**
   - **Verificaión:**
     $$[?\psi]\,\varphi \leftrightarrow (\psi \rightarrow \varphi)$$
   - **Composición:**
     $$[\alpha;\beta]\,\varphi \leftrightarrow [\alpha]\,[\beta]\,\varphi$$
   - **Elección:**
     $$[\alpha \cup \beta]\,\varphi \leftrightarrow ([\alpha]\,\varphi \wedge [\beta]\,\varphi)$$
   - **Repetición:**
     - **Combinación:**
       $$[\alpha^*]\,\varphi \leftrightarrow (\varphi \wedge [\alpha]\,[\alpha^*]\,\varphi)$$
     - **Inducción:**
       $$\Big(\varphi \wedge [\alpha^*]\,(\varphi \rightarrow [\alpha]\,\varphi)\Big) \rightarrow [\alpha^*]\,\varphi$$

Un **teorema** es una fórmula que puede ser derivada en un número **finito** de pasos siguiendo los principios anteriores.

Ejemplo

Demuestre que $[(\alpha \cup \beta);\gamma]\,\varphi \leftrightarrow ([\alpha;\gamma]\,\varphi \wedge [\beta;\gamma]\,\varphi)$ es válida.

Ejemplo

Demuestre que $[(\alpha \cup \beta);\gamma]\,\varphi \leftrightarrow ([\alpha;\gamma]\,\varphi \wedge [\beta;\gamma]\,\varphi)$ es válida.

De izquierda a derecha:

Ejemplo

Demuestre que $[(\alpha \cup \beta);\gamma]\,\varphi \leftrightarrow ([\alpha;\gamma]\,\varphi \wedge [\beta;\gamma]\,\varphi)$ es válida.

De izquierda a derecha:

1.	$[(\alpha \cup \beta);\gamma]\,\varphi$	Suposición

Ejemplo

Demuestre que $[(\alpha \cup \beta);\gamma]\,\varphi \leftrightarrow ([\alpha;\gamma]\,\varphi \land [\beta;\gamma]\,\varphi)$ es válida.

De izquierda a derecha:

1.	$[(\alpha \cup \beta);\gamma]\,\varphi$	Suposición
2.	$[\alpha \cup \beta]\,[\gamma]\,\varphi$	Composición sobre paso 1

Ejemplo

Demuestre que $[(\alpha \cup \beta); \gamma]\, \varphi \leftrightarrow ([\alpha; \gamma]\, \varphi \wedge [\beta; \gamma]\, \varphi)$ es válida.

De izquierda a derecha:

1.	$[(\alpha \cup \beta); \gamma]\, \varphi$	Suposición
2.	$[\alpha \cup \beta]\, [\gamma]\, \varphi$	Composición sobre paso 1
3.	$[\alpha]\, [\gamma]\, \varphi \wedge [\beta]\, [\gamma]\, \varphi$	Elección sobre paso 2

Ejemplo

Demuestre que $[(\alpha \cup \beta); \gamma]\, \varphi \leftrightarrow ([\alpha; \gamma]\, \varphi \wedge [\beta; \gamma]\, \varphi)$ es válida.

De izquierda a derecha:

1.	$[(\alpha \cup \beta); \gamma]\, \varphi$	Suposición
2.	$[\alpha \cup \beta]\, [\gamma]\, \varphi$	Composición sobre paso 1
3.	$[\alpha]\, [\gamma]\, \varphi \wedge [\beta]\, [\gamma]\, \varphi$	Elección sobre paso 2
4.	$[\alpha; \gamma]\, \varphi \wedge [\beta; \gamma]\, \varphi$	Composición sobre paso 3

Ejemplo

Demuestre que $[(\alpha \cup \beta);\gamma]\,\varphi \leftrightarrow ([\alpha;\gamma]\,\varphi \wedge [\beta;\gamma]\,\varphi)$ es válida.

De izquierda a derecha:

1.	$[(\alpha \cup \beta);\gamma]\,\varphi$	Suposición
2.	$[\alpha \cup \beta]\,[\gamma]\,\varphi$	Composición sobre paso 1
3.	$[\alpha]\,[\gamma]\,\varphi \wedge [\beta]\,[\gamma]\,\varphi$	Elección sobre paso 2
4.	$[\alpha;\gamma]\,\varphi \wedge [\beta;\gamma]\,\varphi$	Composición sobre paso 3

De derecha a izquierda la demostración es similar.

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

1. **WHILE** φ **do** α:

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

1. **WHILE** φ **do** α:

$$(?\varphi; \alpha)^*; ?\neg\varphi$$

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

1. **WHILE** φ **do** α:
$$(?\varphi;\alpha)^*;?\neg\varphi$$
2. **REPEAT** α **UNTIL** φ:

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

1. **WHILE** φ **do** α:
$$(?\varphi; \alpha)^*; ?\neg\varphi$$
2. **REPEAT** α **UNTIL** φ:
$$\alpha; (?\neg\varphi; \alpha)^*; ?\varphi$$

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

1. **WHILE** φ **do** α:
$$(?\varphi; \alpha)^*; ?\neg\varphi$$
2. **REPEAT** α **UNTIL** φ:
$$\alpha; (?\neg\varphi; \alpha)^*; ?\varphi$$
3. **IF** φ **THEN** α **ELSE** β:

LDP como lenguaje de programación

Con ***LDP*** podemos definir acciones representando estructuras de control en lenguajes de programación.

1. **WHILE** φ **do** α:
$$(?\varphi;\alpha)^*;?\neg\varphi$$
2. **REPEAT** α **UNTIL** φ:
$$\alpha;(?\neg\varphi;\alpha)^*;?\varphi$$
3. **IF** φ **THEN** α **ELSE** β:
$$(?\varphi;\alpha)\cup(?\neg\varphi;\beta)$$