Lógica Modal

De qué se trata esta materia?

Carlos Areces

1er Cuatrimestre 2012,
Córdoba, Argentina

Generalidades

- En esta materia vamos a hablar de *lógicas modales*.

Generalidades

- En esta materia vamos a hablar de *lógicas modales*.
- La materia va a ser principalmente teórica, pero vamos a ver también algunas aplicaciones.

Generalidades

- En esta materia vamos a hablar de *lógicas modales*.
- La materia va a ser principalmente teórica, pero vamos a ver también algunas aplicaciones.
- Contenidos necesarios para la cursada:

Generalidades

- En esta materia vamos a hablar de *lógicas modales*.
- La materia va a ser principalmente teórica, pero vamos a ver también algunas aplicaciones.
- Contenidos necesarios para la cursada:
  - Buenos conocimientos de lógica proposicional.

Generalidades

- En esta materia vamos a hablar de *lógicas modales*.
- La materia va a ser principalmente teórica, pero vamos a ver también algunas aplicaciones.
- Contenidos necesarios para la cursada:
  - Buenos conocimientos de lógica proposicional.
  - Nociones básicas de lógica de primer orden.

Generalidades

- En esta materia vamos a hablar de *lógicas modales*.
- La materia va a ser principalmente teórica, pero vamos a ver también algunas aplicaciones.
- Contenidos necesarios para la cursada:
  - Buenos conocimientos de lógica proposicional.
  - Nociones básicas de lógica de primer orden.
- Como materia optativa tiene como correlativa *Introducción a la Lógica y la Computación*.

Organización de la materia

- Horario:

Organización de la materia

- Horario:
  - Miércoles de 14hs a 16hs. Aula 11.

Organización de la materia

- Horario:
  - Miércoles de 14hs a 16hs. Aula 11.
  - Jueves de 16hs a 19hs. Aula 11.

Organización de la materia

- Horario:
  - Miércoles de 14hs a 16hs. Aula 11.
  - Jueves de 16hs a 19hs. Aula 11.
- Evaluación: Dos parciales, más ejercicios a entregar durante la cursada. Los que la cursan como materia de posgrado van a tener que trabajar más.

Organización de la materia

- Horario:
  - Miércoles de 14hs a 16hs. Aula 11.
  - Jueves de 16hs a 19hs. Aula 11.
- Evaluación: Dos parciales, más ejercicios a entregar durante la cursada. Los que la cursan como materia de posgrado van a tener que trabajar más.
- Vamos a tener guías de ejercicios. Algunos los vamos a ir resolviendo en el pizarrón.

Organización de la materia

- Horario:
  - Miércoles de 14hs a 16hs. Aula 11.
  - Jueves de 16hs a 19hs. Aula 11.
- Evaluación: Dos parciales, más ejercicios a entregar durante la cursada. Los que la cursan como materia de posgrado van a tener que trabajar más.
- Vamos a tener guías de ejercicios. Algunos los vamos a ir resolviendo en el pizarrón.
- La página de la materia es

  `http://cs.famaf.unc.edu.ar/~careces/lm12`

  Ahí voy a subir las prácticas y las slides de las clases, así que chequeen cada tanto las novedades.

¿Qué es lo que vamos a ver?

¿Qué es lo que vamos a ver?

- Usualmente hablamos de La Lógica: lógica de primer orden.

$$\forall x.(madruga(x) \rightarrow \exists y.(dios(y) \wedge ayuda(y,x)))$$

¿Qué es lo que vamos a ver?

- Usualmente hablamos de La Lógica: lógica de primer orden.

$$\forall x.(madruga(x) \rightarrow \exists y.(dios(y) \wedge ayuda(y, x)))$$

- Aunque también sabemos que existe la lógica proposicional:

$$comerChicle \rightarrow \neg cruzarLaCalle$$

¿Qué es lo que vamos a ver?

- Usualmente hablamos de La Lógica: lógica de primer orden.

$$\forall x.(madruga(x) \rightarrow \exists y.(dios(y) \wedge ayuda(y, x)))$$

- Aunque también sabemos que existe la lógica proposicional:

$$comerChicle \rightarrow \neg cruzarLaCalle$$

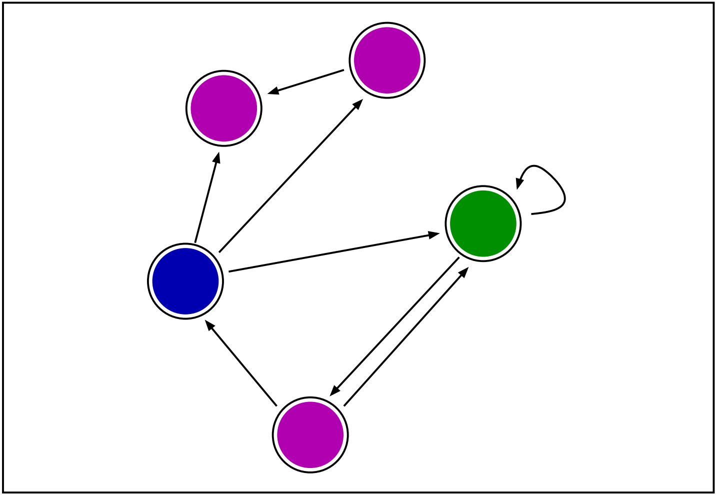
- ¿Cuántas lógicas hay? ¿Dos?

Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:

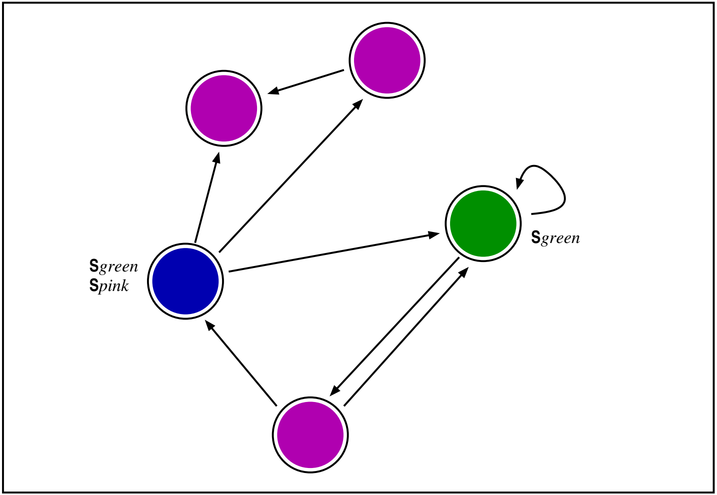
Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:



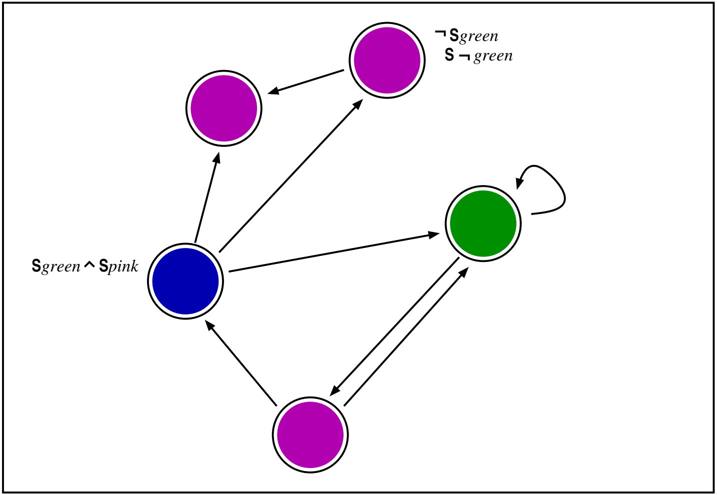
Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:



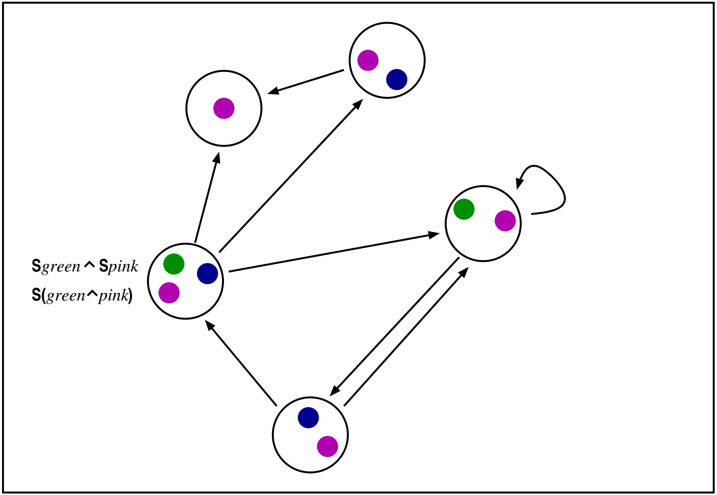
Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:



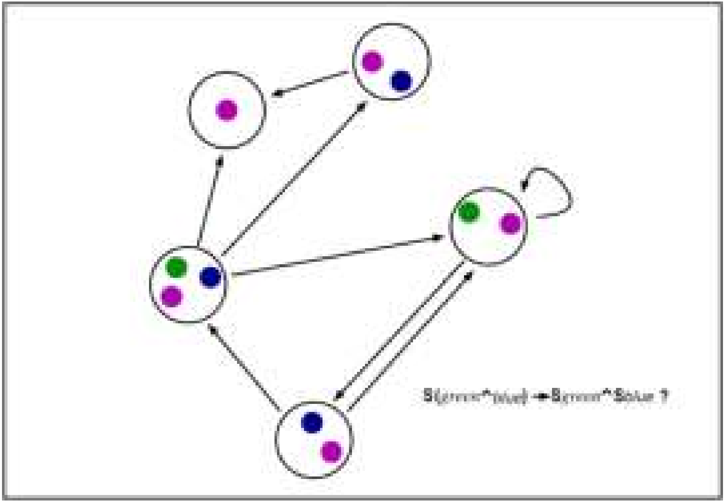
Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:



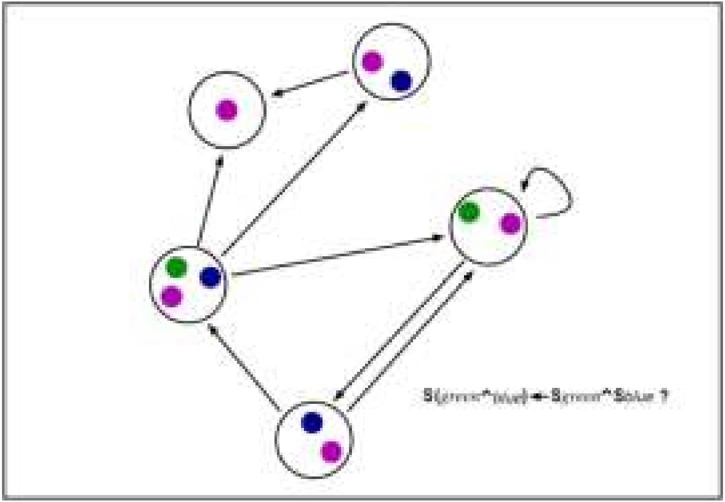
Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:



Estructuras simples para lenguajes simples

Pensemos en un grafo coloreado:



Estructuras simples para lenguajes simples

- Esto que vimos es un ejemplo de una *lógica modal*.

Estructuras simples para lenguajes simples

- Esto que vimos es un ejemplo de una *lógica modal*.
- Aparentemente es un lenguaje muy cómodo para hablar de grafos coloreados. ¿Servirá para algo más? Ya veremos...

Estructuras simples para lenguajes simples

- Esto que vimos es un ejemplo de una *lógica modal*.
- Aparentemente es un lenguaje muy cómodo para hablar de grafos coloreados. ¿Servirá para algo más? Ya veremos...
- Una primera ventaja:
  - Decidir si una fórmula es cierta en lógica de primer orden es indecidible.
  - En esta lógica modal, el problema es computable! (para los que sepan de qué se trata, es PSPACE-complete).

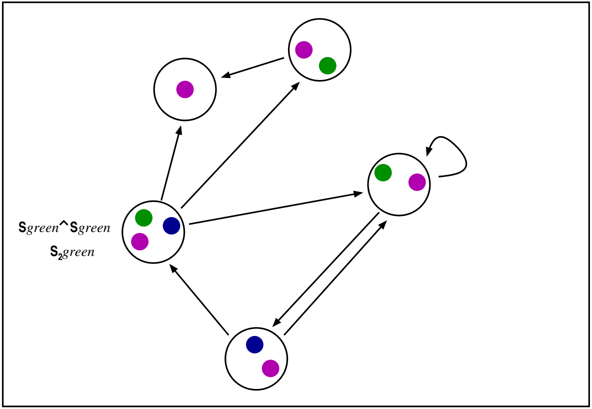
Una Familia de Lenguajes

Una Familia de Lenguajes

- A veces, el lenguaje que acabamos de describir no es adecuado:

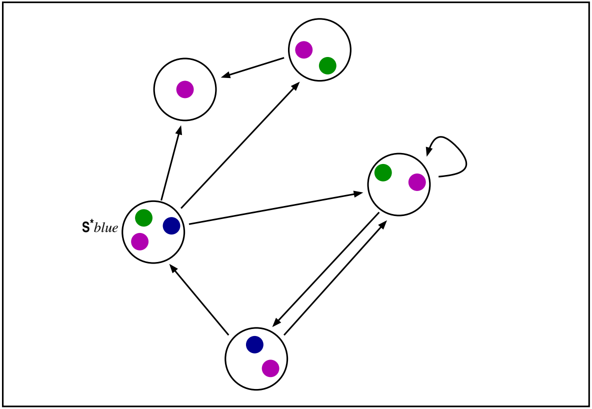
Una Familia de Lenguajes

- A veces, el lenguaje que acabamos de describir no es adecuado:



Una Familia de Lenguajes

- A veces, el lenguaje que acabamos de describir no es adecuado:



Una Familia de Lenguajes

- A veces, el lenguaje que acabamos de describir no es adecuado:
- La Lógica Modal investiga el espectro de posibles lenguajes que pueden usarse para describir estructuras relacionales.

Una Familia de Lenguajes

- A veces, el lenguaje que acabamos de describir no es adecuado:
- La Lógica Modal investiga el espectro de posibles lenguajes que pueden usarse para describir estructuras relacionales.
- Algunas preguntas que uno se puede hacer son:
  - ¿Cuáles son los límites de expresividad de estos lenguajes? Es decir, ¿podemos decir más o menos cosas que con otras lógicas?
  - ¿Podemos definir algoritmos de inferencia para estos lenguajes?
  - ¿Cuán eficientes son?

Una Familia de Lenguajes

- A veces, el lenguaje que acabamos de describir no es adecuado:
- La Lógica Modal investiga el espectro de posibles lenguajes que pueden usarse para describir estructuras relacionales.
- Algunas preguntas que uno se puede hacer son:
  - ¿Cuáles son los límites de expresividad de estos lenguajes? Es decir, ¿podemos decir más o menos cosas que con otras lógicas?
  - ¿Podemos definir algoritmos de inferencia para estos lenguajes?
  - ¿Cuán eficientes son?
- Otra perspectiva es mirarlos desde el punto de vista del diseño de una lógica. Dado un problema en particular:
  - ¿Qué lenguaje lógico me resulta más cómodo de usar?
  - ¿Cuál tiene un buen algoritmo de inferencia?
  - ¿Qué operadores realmente necesito?

Posibles Aplicaciones

Posibles Aplicaciones

- Los lenguajes de la familia de las lógicas modales pueden usarse en áreas muy diversas:

Posibles Aplicaciones

- Los lenguajes de la familia de las lógicas modales pueden usarse en áreas muy diversas:
  - Verificación de Software y Hardware.
  - Representación de Conocimientos.
  - Linguística Computacional.
  - Inteligencia Artificial.
  - Filosofía.
  - Epistemología.
  - …

Posibles Aplicaciones

- Los lenguajes de la familia de las lógicas modales pueden usarse en áreas muy diversas:
  - Verificación de Software y Hardware.
  - Representación de Conocimientos.
  - Linguística Computacional.
  - Inteligencia Artificial.
  - Filosofía.
  - Epistemología.
  - …
- **¿Por qué?**

Posibles Aplicaciones

- Los lenguajes de la familia de las lógicas modales pueden usarse en áreas muy diversas:
  - Verificación de Software y Hardware.
  - Representación de Conocimientos.
  - Linguística Computacional.
  - Inteligencia Artificial.
  - Filosofía.
  - Epistemología.
  - …
- **¿Por qué?** Muchas cosas pueden ser representadas como grafos (i.e., estructuras relacionales).

Posibles Aplicaciones

- Los lenguajes de la familia de las lógicas modales pueden usarse en áreas muy diversas:
  - Verificación de Software y Hardware.
  - Representación de Conocimientos.
  - Linguística Computacional.
  - Inteligencia Artificial.
  - Filosofía.
  - Epistemología.
  - …
- **¿Por qué?** Muchas cosas pueden ser representadas como grafos (i.e., estructuras relacionales).
  Y como vimos, los lenguajes modales fueron desarrollados especialmente para razonar sobre grafos y describir sus propiedades.

No Estamos Solos

No Estamos Solos

- Aunque no parezca, estamos haciendo Lógica Clásica.

No Estamos Solos

- Aunque no parezca, estamos haciendo Lógica Clásica.
- Los lenguajes que estuvimos discutiendo son fragmentos del lenguaje de primer (o segundo) orden. Lo único que hicimos fue elegir sólo una parte del lenguaje que necesitábamos para una aplicación dada.

No Estamos Solos

- Aunque no parezca, estamos haciendo Lógica Clásica.
- Los lenguajes que estuvimos discutiendo son fragmentos del lenguaje de primer (o segundo) orden. Lo único que hicimos fue elegir sólo una parte del lenguaje que necesitábamos para una aplicación dada.
- Esta es exactamente la forma en que vemos hoy por hoy a los lenguajes modales, como una forma de investigar fragmentos particularmente interesantes de los lenguajes clásicos.

No Estamos Solos

- Aunque no parezca, estamos haciendo Lógica Clásica.
- Los lenguajes que estuvimos discutiendo son fragmentos del lenguaje de primer (o segundo) orden. Lo único que hicimos fue elegir sólo una parte del lenguaje que necesitábamos para una aplicación dada.
- Esta es exactamente la forma en que vemos hoy por hoy a los lenguajes modales, como una forma de investigar fragmentos particularmente interesantes de los lenguajes clásicos.
- Donde "interesantes" significa
  - Decidibles, expresivos, de "baja" complejidad, modulares, etc.

Clase #1

Temario:

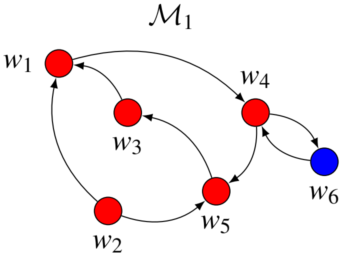
- Lógica de primer orden: sentencias y fórmulas.
- El lenguaje modal visto como un lenguaje de grafos decorados.
- Sintaxis y semántica del lenguaje modal
- Extensiones: Operador inverso, modalidad universal, PDL, lógicas híbridas.

Lógica de Primer Orden

- La noción de verdad en Lógica de Primer Orden tiene que ver con sentencias, es decir, fórmulas sin variables libres.

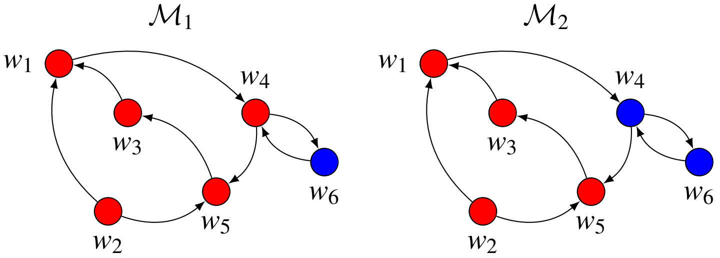
Lógica de Primer Orden

- La noción de verdad en Lógica de Primer Orden tiene que ver con sentencias, es decir, fórmulas sin variables libres.
- Si pensamos en una sentencia φ cualquiera, y un modelo $\mathcal{M}$, la sentencia va a ser verdadera o falsa en todo el modelo. No vamos a poder hablar de una parte del modelo en particular.



Lógica de Primer Orden

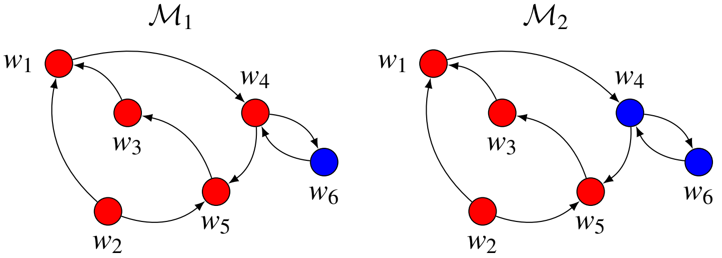
- La noción de verdad en Lógica de Primer Orden tiene que ver con sentencias, es decir, fórmulas sin variables libres.
- Si pensamos en una sentencia φ cualquiera, y un modelo $\mathcal{M}$, la sentencia va a ser verdadera o falsa en todo el modelo. No vamos a poder hablar de una parte del modelo en particular.



- $\mathcal{M}_1 \models \forall x.(red(x) \to (\exists y.xRy \wedge red(y)))$

Lógica de Primer Orden

- La noción de verdad en Lógica de Primer Orden tiene que ver con sentencias, es decir, fórmulas sin variables libres.
- Si pensamos en una sentencia φ cualquiera, y un modelo $\mathcal{M}$, la sentencia va a ser verdadera o falsa en todo el modelo. No vamos a poder hablar de una parte del modelo en particular.



- $\mathcal{M}_1 \models \forall x.(red(x) \rightarrow (\exists y.xRy \wedge red(y)))$
- $\mathcal{M}_2 \not\models \forall x.(red(x) \rightarrow (\exists y.xRy \wedge red(y)))$

Lógica de Primer Orden

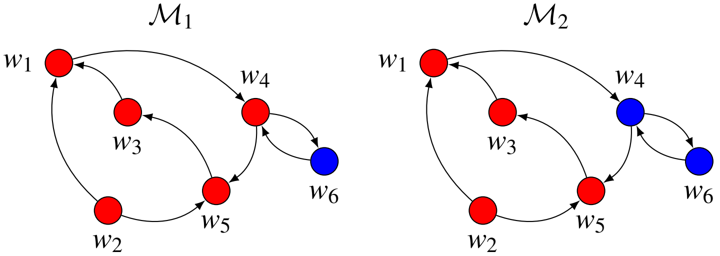
- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.

Lógica de Primer Orden

- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.
- ¿Cómo podemos hacer para hablar de partes del modelo?

Lógica de Primer Orden

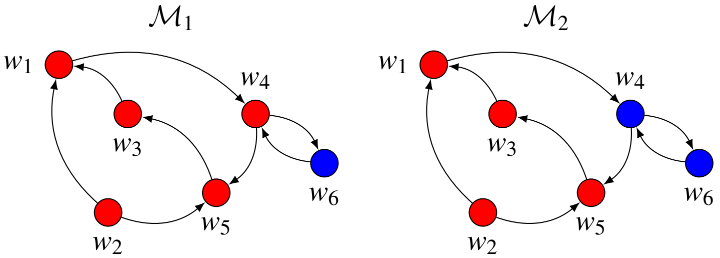
- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.
- ¿Cómo podemos hacer para hablar de partes del modelo?
- Podemos usar fórmulas con variables libres:



- $sRed(x) = red(x) \rightarrow (\exists y.xRy \wedge red(y))$

Lógica de Primer Orden

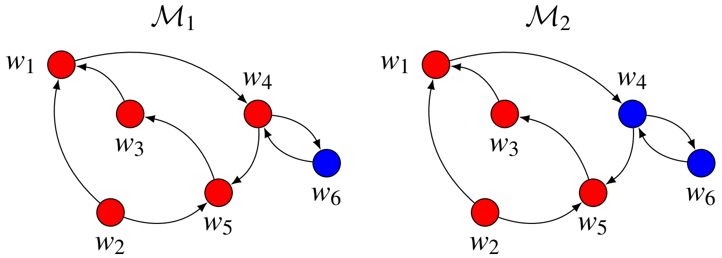
- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.
- ¿Cómo podemos hacer para hablar de partes del modelo?
- Podemos usar fórmulas con variables libres:



- $sRed(x) = red(x) \rightarrow (\exists y.xRy \wedge red(y))$
- $[sRed(x)]^{\mathcal{M}_1} = [red(x) \rightarrow (\exists y.xRy \wedge red(y))]^{\mathcal{M}_1} =$

Lógica de Primer Orden

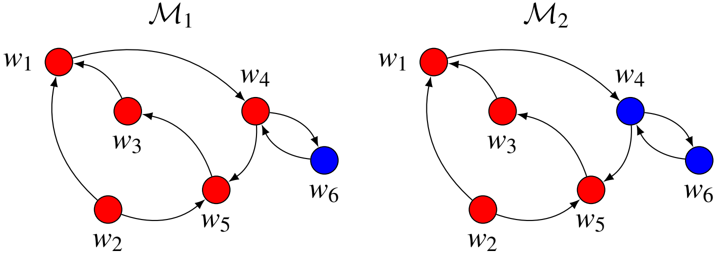
- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.
- ¿Cómo podemos hacer para hablar de partes del modelo?
- Podemos usar fórmulas con variables libres:



- $sRed(x) = red(x) \to (\exists y.xRy \land red(y))$
- $[sRed(x)]^{\mathcal{M}_1} = [red(x) \to (\exists y.xRy \land red(y))]^{\mathcal{M}_1} = \{w_1, \ldots, w_6\}$

Lógica de Primer Orden

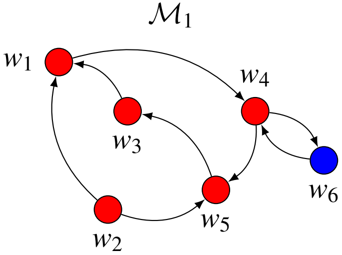
- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.
- ¿Cómo podemos hacer para hablar de partes del modelo?
- Podemos usar fórmulas con variables libres:

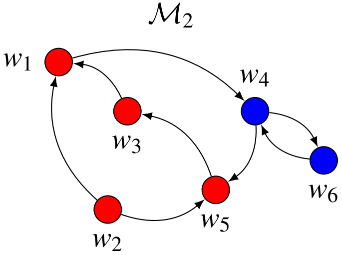


- $sRed(x) = red(x) \rightarrow (\exists y.xRy \wedge red(y))$
- $[sRed(x)]^{\mathcal{M}_1} = [red(x) \rightarrow (\exists y.xRy \wedge red(y))]^{\mathcal{M}_1} = \{w_1, \ldots, w_6\}$
- $[sRed(x)]^{\mathcal{M}_2} = [red(x) \rightarrow (\exists y.xRy \wedge red(y))]^{\mathcal{M}_2} =$

Lógica de Primer Orden

- Esto significa que la extensión de una sentencia φ de LPO es o bien vacío o bien todo el dominio.
- ¿Cómo podemos hacer para hablar de partes del modelo?
- Podemos usar fórmulas con variables libres:





- $sRed(x) = red(x) \rightarrow (\exists y.xRy \wedge red(y))$
- $[sRed(x)]^{\mathcal{M}_1} = [red(x) \rightarrow (\exists y.xRy \wedge red(y))]^{\mathcal{M}_1} = \{w_1, \ldots, w_6\}$
- $[sRed(x)]^{\mathcal{M}_2} = [red(x) \rightarrow (\exists y.xRy \wedge red(y))]^{\mathcal{M}_2} = \{w_2, \ldots, w_6\}$

Perspectiva interna

- De alguna manera, lo que estamos buscando es intentar expresar una noción de perspectiva interna, en donde queremos ver las propiedades de un elemento del modelo con respecto al resto.

Perspectiva interna

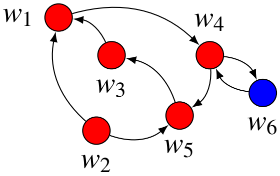
- De alguna manera, lo que estamos buscando es intentar expresar una noción de perspectiva interna, en donde queremos ver las propiedades de un elemento del modelo con respecto al resto.
- Con eso en mente, vamos a dejar por un momento LPO, y vamos a trabajar con un lenguaje especialmente diseñado para eso.

Perspectiva interna

- De alguna manera, lo que estamos buscando es intentar expresar una noción de perspectiva interna, en donde queremos ver las propiedades de un elemento del modelo con respecto al resto.
- Con eso en mente, vamos a dejar por un momento LPO, y vamos a trabajar con un lenguaje especialmente diseñado para eso.
- Vamos a usar un lenguaje modal

Perspectiva interna

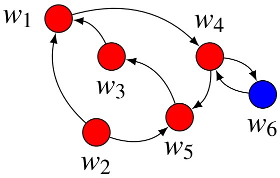
- De alguna manera, lo que estamos buscando es intentar expresar una noción de perspectiva interna, en donde queremos ver las propiedades de un elemento del modelo con respecto al resto.
- Con eso en mente, vamos a dejar por un momento LPO, y vamos a trabajar con un lenguaje especialmente diseñado para eso.
- Vamos a usar un lenguaje modal
- Ahora, si queremos expresar la idea anterior, podemos decir:



$$\mathcal{M}, w_1 \models red \rightarrow \Diamond red$$

Perspectiva interna

- De alguna manera, lo que estamos buscando es intentar expresar una noción de perspectiva interna, en donde queremos ver las propiedades de un elemento del modelo con respecto al resto.
- Con eso en mente, vamos a dejar por un momento LPO, y vamos a trabajar con un lenguaje especialmente diseñado para eso.
- Vamos a usar un lenguaje modal
- Ahora, si queremos expresar la idea anterior, podemos decir:

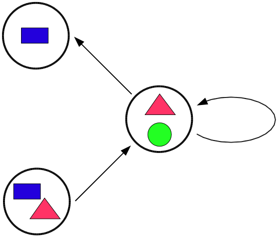


$$\mathcal{M}, w_1 \models red \rightarrow \Diamond red$$

- Pareciera mucho más fácil de escribir así, ¿no?

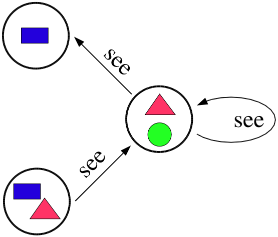
Lenguaje Modal: un lenguaje para grafos decorados

Veamos otros ejemplos. Pensemos en un grafo orientado, con figuras dentro de cada nodo:



Lenguaje Modal: un lenguaje para grafos decorados

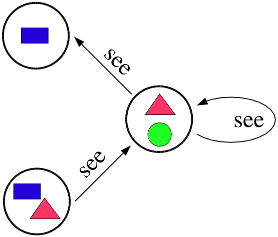
Veamos otros ejemplos. Pensemos en un grafo orientado, con figuras dentro de cada nodo:



Queremos describir qué figuras un nodo puede "ver".

Lenguaje Modal: un lenguaje para grafos decorados

Veamos otros ejemplos. Pensemos en un grafo orientado, con figuras dentro de cada nodo:



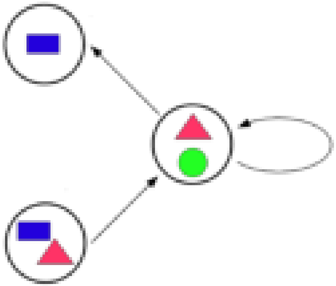
Queremos describir qué figuras un nodo puede "ver".

Desde la perspectiva de un nodo n, el significado de los operadores modales va a ser:

- $\langle see \rangle x$ = "n puede ver la figura x en algún vecino".
- $[see]x$ = "n puede ver la figura x en todos sus vecinos".

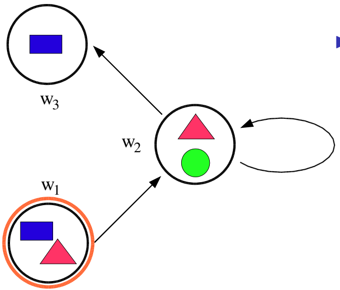
Lenguaje Modal: un lenguaje para grafos decorados

Ahora podemos hacerle "preguntas" al modelo:



Lenguaje Modal: un lenguaje para grafos decorados

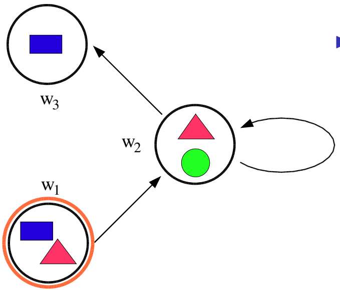
Ahora podemos hacerle “preguntas” al modelo:



- Ve w_1 un rectángulo?

Lenguaje Modal: un lenguaje para grafos decorados

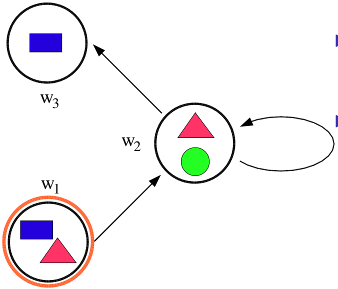
Ahora podemos hacerle “preguntas” al modelo:



- Ve w_1 un rectángulo?
  $\mathcal{M}, w_1 \models \langle see \rangle rectangle$

Lenguaje Modal: un lenguaje para grafos decorados

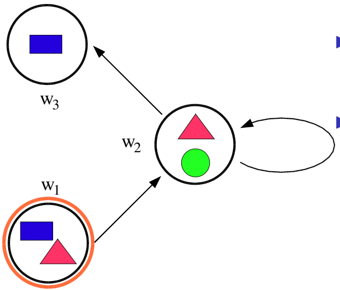
Ahora podemos hacerle "preguntas" al modelo:



- Ve w_1 un rectángulo?
  $\mathcal{M}, w_1 \models \langle see \rangle rectangle$
- ... y en dos "pasos"?

Lenguaje Modal: un lenguaje para grafos decorados

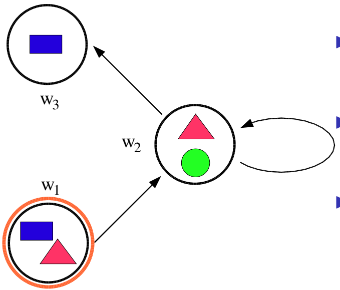
Ahora podemos hacerle “preguntas” al modelo:



- Ve w_1 un rectángulo?
  $\mathcal{M}, w_1 \models \langle see \rangle rectangle$
- ... y en dos “pasos”?
  $\mathcal{M}, w_1 \models \langle see \rangle \langle see \rangle rectangle$

Lenguaje Modal: un lenguaje para grafos decorados

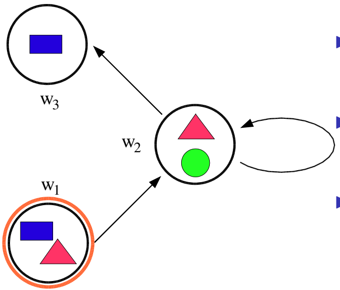
Ahora podemos hacerle “preguntas” al modelo:



- Ve w_1 un rectángulo?
  $\mathcal{M}, w_1 \models \langle see \rangle rectangle$
- ... y en dos “pasos”?
  $\mathcal{M}, w_1 \models \langle see \rangle \langle see \rangle rectangle$
- Ve w_1 un círculo y un triángulo?

Lenguaje Modal: un lenguaje para grafos decorados

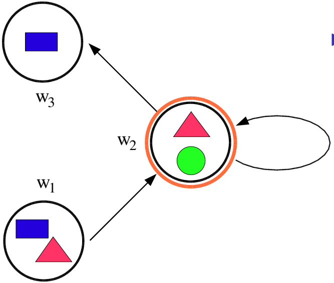
Ahora podemos hacerle "preguntas" al modelo:



- Ve w_1 un rectángulo?
  $\mathcal{M}, w_1 \models \langle see \rangle rectangle$
- ... y en dos "pasos"?
  $\mathcal{M}, w_1 \models \langle see \rangle \langle see \rangle rectangle$
- Ve w_1 un círculo y un triángulo?
  $\mathcal{M}, w_1 \models \langle see \rangle circle \wedge \langle see \rangle triangle$

Lenguaje Modal: un lenguaje para grafos decorados

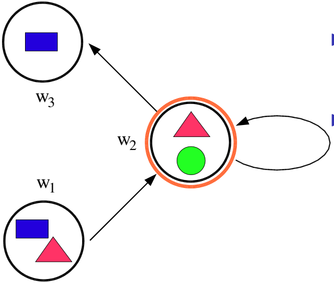
Ahora podemos hacerle "preguntas" al modelo:



- Ve w_2 un círculo y un rectángulo?
  $\mathcal{M}, w_2 \models \langle see \rangle circle \wedge \langle see \rangle rectangle$

Lenguaje Modal: un lenguaje para grafos decorados

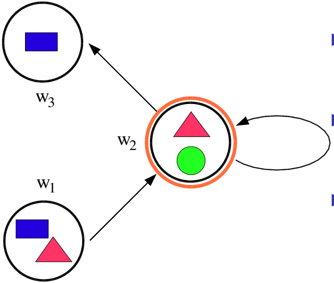
Ahora podemos hacerle "preguntas" al modelo:



- Ve w_2 un círculo y un rectángulo?
  $\mathcal{M}, w_2 \models \langle see \rangle circle \wedge \langle see \rangle rectangle$
- ... y en el mismo vecino?
  $\mathcal{M}, w_2 \not\models \langle see \rangle (circle \wedge rectangle)$

Lenguaje Modal: un lenguaje para grafos decorados

Ahora podemos hacerle “preguntas” al modelo:



- Ve w_2 un círculo y un rectángulo?
  $\mathcal{M}, w_2 \models \langle see \rangle circle \wedge \langle see \rangle rectangle$
- ... y en el mismo vecino?
  $\mathcal{M}, w_2 \not\models \langle see \rangle (circle \wedge rectangle)$
- w_2 ve círculos en todos lados?
  $\mathcal{M}, w_2 \not\models [see] circle$

Lenguaje Modal: un lenguaje para grafos decorados

También podemos verlo como un lenguaje para describir procesos.

Lenguaje Modal: un lenguaje para grafos decorados

También podemos verlo como un lenguaje para describir procesos.

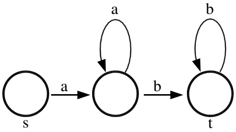
- Esto significa ver a los elementos del modelo como un conjunto de estados computacionales.

Lenguaje Modal: un lenguaje para grafos decorados

También podemos verlo como un lenguaje para describir procesos.

- Esto significa ver a los elementos del modelo como un conjunto de estados computacionales.
- Y ver a las relaciones binarias como acciones que transforman un estado en otro.

Veamos este modelo:

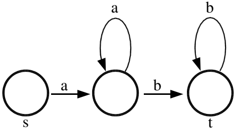


Lenguaje Modal: un lenguaje para grafos decorados

También podemos verlo como un lenguaje para describir procesos.

- Esto significa ver a los elementos del modelo como un conjunto de estados computacionales.
- Y ver a las relaciones binarias como acciones que transforman un estado en otro.

  Veamos este modelo:



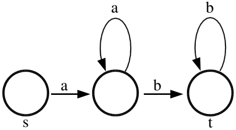
- Esto muestra un autómata finito para el lenguaje $a^n b^m$.

Lenguaje Modal: un lenguaje para grafos decorados

También podemos verlo como un lenguaje para describir procesos.

- Esto significa ver a los elementos del modelo como un conjunto de estados computacionales.
- Y ver a las relaciones binarias como acciones que transforman un estado en otro.

  Veamos este modelo:



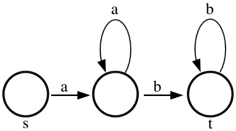
- Esto muestra un autómata finito para el lenguaje $a^n b^m$.
- En este caso podemos trabajar con dos diamantes $\langle a \rangle$ y $\langle b \rangle$.

Lenguaje Modal: un lenguaje para grafos decorados

También podemos verlo como un lenguaje para describir procesos.

- Esto significa ver a los elementos del modelo como un conjunto de estados computacionales.
- Y ver a las relaciones binarias como acciones que transforman un estado en otro.

  Veamos este modelo:



- Esto muestra un autómata finito para el lenguaje $a^n b^m$.
- En este caso podemos trabajar con dos diamantes $\langle a \rangle$ y $\langle b \rangle$.
- Está claro que todas las fórmulas con la forma

$$\langle a \rangle \ldots \langle a \rangle \langle b \rangle \ldots \langle b \rangle t$$

son satisfechas en el nodo s.

Lenguaje Modal: una perspectiva realmente interna

Miremos este ejemplo:

- Hay varias habitaciones, pintadas de rojo y negro, y en cada una hay un teletransportador (wow!).

Lenguaje Modal: una perspectiva realmente interna

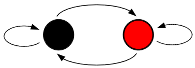
Miremos este ejemplo:

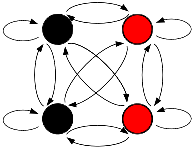
- Hay varias habitaciones, pintadas de rojo y negro, y en cada una hay un teletransportador (wow!).
- Este transportador nos mueve entre algunas de las habitaciones siguiendo alguno de los posibles caminos (uno entra en el transportador y aparece en alguna otra habitación al azar).

Lenguaje Modal: una perspectiva realmente interna

Miremos este ejemplo:

- Hay varias habitaciones, pintadas de rojo y negro, y en cada una hay un teletransportador (wow!).
- Este transportador nos mueve entre algunas de las habitaciones siguiendo alguno de los posibles caminos (uno entra en el transportador y aparece en alguna otra habitación al azar).
- ¿Podemos diferenciar entre estos dos 'laberintos'?

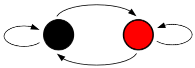


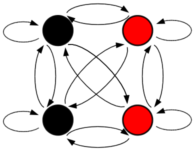


Lenguaje Modal: una perspectiva realmente interna

Miremos este ejemplo:

- Hay varias habitaciones, pintadas de rojo y negro, y en cada una hay un teletransportador (wow!).
- Este transportador nos mueve entre algunas de las habitaciones siguiendo alguno de los posibles caminos (uno entra en el transportador y aparece en alguna otra habitación al azar).
- ¿Podemos diferenciar entre estos dos 'laberintos'?





- Aunque uno de los modelos tiene solo 2 elementos y el otro 4, no hay forma de notar esto 'desde adentro' de los modelos. Como vamos a ver, esta es una característica importante de las lógicas modales relacionada con la expresividad.

Lenguaje Modal Básico: sintaxis y semántica

Vamos a definir ahora la sintaxis formal del lenguaje modal básico.

- La signatura del lenguaje va a consistir de dos conjuntos infinitos numerables, disjuntos entre sí:
  - $\text{PROP} = \{p_1, p_2, \dots\}$, el conjunto de *variables proposicionales*.
  - $\text{REL} = \{R_1, R_2, \dots\}$, el conjunto de *símbolos de relación*.

Lenguaje Modal Básico: sintaxis y semántica

Vamos a definir ahora la sintaxis formal del lenguaje modal básico.

- La signatura del lenguaje va a consistir de dos conjuntos infinitos numerables, disjuntos entre sí:
  - $\text{PROP} = \{p_1, p_2, \ldots\}$, el conjunto de *variables proposicionales*.
  - $\text{REL} = \{R_1, R_2, \ldots\}$, el conjunto de *símbolos de relación*.
- El conjunto de fórmulas FORM en la signatura $\langle \text{PROP}, \text{REL} \rangle$ está definido como:

$$\text{FORM} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle \varphi$$

en donde $p \in \text{PROP}, R \in \text{REL}$ y $\varphi, \psi \in \text{FORM}$.

Lenguaje Modal Básico: sintaxis y semántica

Vamos a definir ahora la sintaxis formal del lenguaje modal básico.

- La signatura del lenguaje va a consistir de dos conjuntos infinitos numerables, disjuntos entre sí:
  - $\text{PROP} = \{p_1, p_2, \ldots\}$, el conjunto de *variables proposicionales*.
  - $\text{REL} = \{R_1, R_2, \ldots\}$, el conjunto de *símbolos de relación*.
- El conjunto de fórmulas FORM en la signatura $\langle \text{PROP}, \text{REL} \rangle$ está definido como:

  $$\text{FORM} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle \varphi$$

  en donde $p \in \text{PROP}$, $R \in \text{REL}$ y $\varphi, \psi \in \text{FORM}$.
- El operador $[R]$ se define como $[R]\varphi = \neg\langle R \rangle \neg\varphi$ (de igual forma que $\forall x.\varphi$ es $\neg\exists x.\neg\varphi$)

Lenguaje Modal Básico: sintaxis y semántica

Para poder definir formalmente la semántica, vamos primero a ver qué es un modelo de Kripke.

Lenguaje Modal Básico: sintaxis y semántica

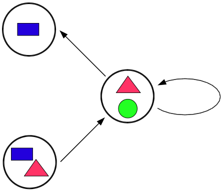
Para poder definir formalmente la semántica, vamos primero a ver qué es un modelo de Kripke.

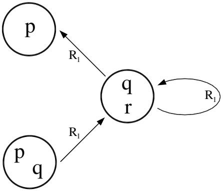
- Un modelo de Kripke es una estructura $\langle W, \{R_i\}, V\rangle$ donde
  - W es un conjunto no vacío de elementos.
  - $\{R_i\}$ es un conjunto de relaciones binarias en W.
  - $V : \textsc{prop} \to \wp(W)$ es una función de valuación ($V(p)$ es el conjunto de elementos donde vale p).

Lenguaje Modal Básico: sintaxis y semántica

Para poder definir formalmente la semántica, vamos primero a ver qué es un modelo de Kripke.

- Un modelo de Kripke es una estructura $\langle W, \{R_i\}, V\rangle$ donde
  - W es un conjunto no vacío de elementos.
  - $\{R_i\}$ es un conjunto de relaciones binarias en W.
  - $V : \textsc{prop} \to \wp(W)$ es una función de valuación ($V(p)$ es el conjunto de elementos donde vale p).
- Intuitivamente, un modelo de Kripke es un grafo dirigido con "decoraciones".





Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

  $\mathcal{M}, w \models p$ iff

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$\mathcal{M}, w \models p \qquad \text{iff} \quad w \in V(p), \text{ para } p \in \text{PROP}$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$\begin{array}{lll} \mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\ \mathcal{M}, w \models \neg\varphi & \text{iff} & \end{array}$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi
\end{array}
$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} &
\end{array}
$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi
\end{array}
$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \textsc{prop} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle R\rangle\varphi & \text{iff} &
\end{array}
$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle R\rangle\varphi & \text{iff} & \exists w' \in W \text{ tq } R(w, w') \text{ y } \mathcal{M}, w' \models \varphi
\end{array}
$$

Lenguaje Modal Básico: sintaxis y semántica

Ahora sí, podemos definir la semántica de la lógica modal básica:

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad es

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle R \rangle\varphi & \text{iff} & \exists w' \in W \text{ tq } R(w, w') \text{ y } \mathcal{M}, w' \models \varphi
\end{array}
$$

- Vamos a decir que φ es válida en un modelo $\mathcal{M}$ sii en todos los estados $w \in W$ vale $\mathcal{M}, w \models \varphi$, y en ese caso escribimos $\mathcal{M} \models \varphi$.

Extensiones

- Hasta ahora hicimos un repaso de lógica proposicional, lógica de primer orden y vimos la lógica modal básica.

Extensiones

- Hasta ahora hicimos un repaso de lógica proposicional, lógica de primer orden y vimos la lógica modal básica.
- Volviendo a la pregunta de cuántas lógicas había, ya sabemos que la respuesta no es dos.

Extensiones

- Hasta ahora hicimos un repaso de lógica proposicional, lógica de primer orden y vimos la lógica modal básica.
- Volviendo a la pregunta de cuántas lógicas había, ya sabemos que la respuesta no es dos.
- Como se podrán imaginar, tampoco es tres.

Extensiones

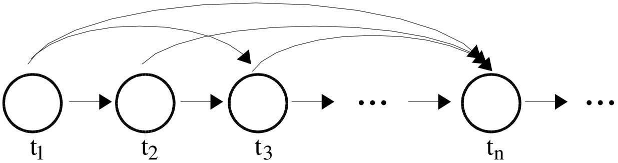
- Hasta ahora hicimos un repaso de lógica proposicional, lógica de primer orden y vimos la lógica modal básica.
- Volviendo a la pregunta de cuántas lógicas había, ya sabemos que la respuesta no es dos.
- Como se podrán imaginar, tampoco es tres.
- Así como existe el operador $\langle R \rangle$, tenemos un amplio "menú" de operadores modales para usar.

Extensiones

- Hasta ahora hicimos un repaso de lógica proposicional, lógica de primer orden y vimos la lógica modal básica.
- Volviendo a la pregunta de cuántas lógicas había, ya sabemos que la respuesta no es dos.
- Como se podrán imaginar, tampoco es tres.
- Así como existe el operador $\langle R \rangle$, tenemos un amplio "menú" de operadores modales para usar.
- Al combinar estos operadores, conseguimos diferentes lógicas

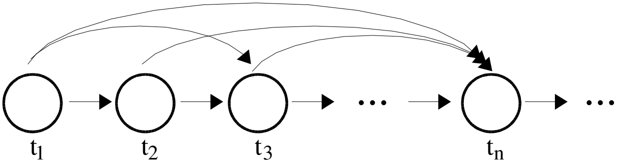
Extensiones: Operador Inverso

- Pensemos en la siguiente "línea temporal"



Extensiones: Operador Inverso

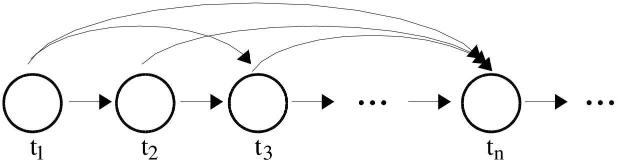
- Pensemos en la siguiente "línea temporal"



- Claramente, esta estructura puede ser vista como un modelo de Kripke.

Extensiones: Operador Inverso

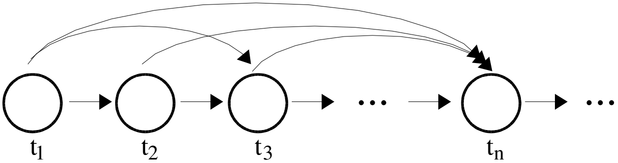
- Pensemos en la siguiente "línea temporal"



- Claramente, esta estructura puede ser vista como un modelo de Kripke.
- El operador $\langle R \rangle$ significa "en algún momento en el futuro".

Extensiones: Operador Inverso

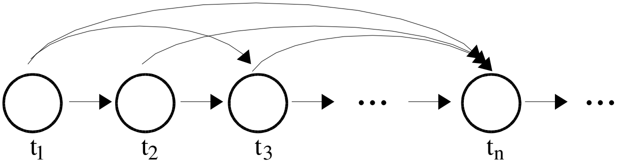
- Pensemos en la siguiente "línea temporal"



- Claramente, esta estructura puede ser vista como un modelo de Kripke.
- El operador $\langle R \rangle$ significa "en algún momento en el futuro".
- Y el operador $[R]$ dice "en todo momento futuro".

Extensiones: Operador Inverso

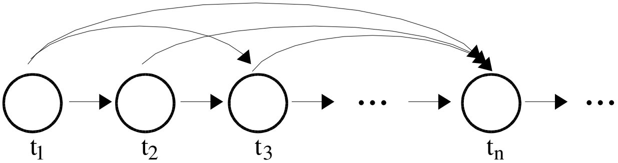
- Pensemos en la siguiente “línea temporal”



- Claramente, esta estructura puede ser vista como un modelo de Kripke.
- El operador $\langle R \rangle$ significa “en algún momento en el futuro”.
- Y el operador $[R]$ dice “en todo momento futuro”.
- Con este lenguaje, ¿podremos decir “en algún momento en el pasado”?

Extensiones: Operador Inverso

- Pensemos en la siguiente "línea temporal"



- Claramente, esta estructura puede ser vista como un modelo de Kripke.
- El operador $\langle R \rangle$ significa "en algún momento en el futuro".
- Y el operador $[R]$ dice "en todo momento futuro".
- Con este lenguaje, ¿podremos decir "en algún momento en el pasado"?
- Más aún, esta idea la podemos querer usar en cualquier tipo de modelo de Kripke, no sólo en los temporales.

Extensiones: Operador Inverso

- La necesidad de expresar esto, nos lleva a agregar el operador inverso, que vamos a notar como $\langle R \rangle^{-1}$.

Extensiones: Operador Inverso

- La necesidad de expresar esto, nos lleva a agregar el operador inverso, que vamos a notar como $\langle R \rangle^{-1}$.
- Para extender la lógica modal básica con este operador, primero tenemos que agregarlo a nuestra sintaxis:

$$\text{FORM} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle\varphi \mid \langle R \rangle^{-1}\varphi$$

Extensiones: Operador Inverso

- La necesidad de expresar esto, nos lleva a agregar el operador inverso, que vamos a notar como $\langle R \rangle^{-1}$.
- Para extender la lógica modal básica con este operador, primero tenemos que agregarlo a nuestra sintaxis:

$$\text{FORM} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle\varphi \mid \langle R \rangle^{-1}\varphi$$

- ¿Necesitamos hacer algún cambio en los modelos?

Extensiones: Operador Inverso

- La necesidad de expresar esto, nos lleva a agregar el operador inverso, que vamos a notar como $\langle R\rangle^{-1}$.
- Para extender la lógica modal básica con este operador, primero tenemos que agregarlo a nuestra sintaxis:

$$\text{FORM} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R\rangle\varphi \mid \langle R\rangle^{-1}\varphi$$

- ¿Necesitamos hacer algún cambio en los modelos?
- Parecería que no...

Extensiones: Operador Inverso

- La necesidad de expresar esto, nos lleva a agregar el operador inverso, que vamos a notar como $\langle R\rangle^{-1}$.
- Para extender la lógica modal básica con este operador, primero tenemos que agregarlo a nuestra sintaxis:

$$\text{FORM} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R\rangle\varphi \mid \langle R\rangle^{-1}\varphi$$

- ¿Necesitamos hacer algún cambio en los modelos?
- Parecería que no...
- Ahora lo que nos falta es extender la semántica.

Extensiones: Operador Inverso

- La relación de satisfacibilidad que teníamos antes era:

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle R \rangle \varphi & \text{iff} & \exists w' \in W \text{ tq } R(w, w') \text{ y } \mathcal{M}, w' \models \varphi
\end{array}
$$

Extensiones: Operador Inverso

- La relación de satisfacibilidad que teníamos antes era:

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle R \rangle \varphi & \text{iff} & \exists w' \in W \text{ tq } R(w, w') \text{ y } \mathcal{M}, w' \models \varphi
\end{array}
$$

- ¿Cómo podemos hacer para definir el comportamiento de este nuevo operador?

Extensiones: Operador Inverso

- La relación de satisfacibilidad que teníamos antes era:

$$\begin{array}{lll} \mathcal{M}, w \models p & \text{iff} & w \in V(p), \text{ para } p \in \text{PROP} \\ \mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\ \mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ y } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models \langle R \rangle \varphi & \text{iff} & \exists w' \in W \text{ tq } R(w, w') \text{ y } \mathcal{M}, w' \models \varphi \end{array}$$

- ¿Cómo podemos hacer para definir el comportamiento de este nuevo operador?
- Tenemos que agregar el caso del operador a la definición:

$$\mathcal{M}, w \models \langle R \rangle^{-1} \varphi \quad \text{iff} \quad \exists w' \in W \text{ tq } R(w', w) \text{ y } \mathcal{M}, w' \models \varphi$$

Extensiones: Modalidad Universal

- Otro “feature” que podemos querer es la capacidad de describir alguna propiedad global, que debe cumplirse en todo el modelo.

Extensiones: Modalidad Universal

- Otro "feature" que podemos querer es la capacidad de describir alguna propiedad global, que debe cumplirse en todo el modelo.
- Supongamos que estamos modelando un zoológico, y que estamos interesados en la alimentación de los osos y su relación con los cuidadores de osos.

Extensiones: Modalidad Universal

- Otro "feature" que podemos querer es la capacidad de describir alguna propiedad global, que debe cumplirse en todo el modelo.
- Supongamos que estamos modelando un zoológico, y que estamos interesados en la alimentación de los osos y su relación con los cuidadores de osos.
- Queremos, por un lado, que estas fórmulas se satisfagan en todo el modelo:

$$\begin{array}{cc} \mathit{oso} \vee \mathit{humano} & \mathit{oso} \rightarrow \langle \text{MADRE} \rangle \mathit{oso} \\ \mathit{oso} \rightarrow \neg \mathit{humano} & \mathit{oso} \rightarrow [\text{ALIMENTADO}](\mathit{cuidador} \vee \mathit{madre}) \end{array}$$

Extensiones: Modalidad Universal

- Otro "feature" que podemos querer es la capacidad de describir alguna propiedad global, que debe cumplirse en todo el modelo.
- Supongamos que estamos modelando un zoológico, y que estamos interesados en la alimentación de los osos y su relación con los cuidadores de osos.
- Queremos, por un lado, que estas fórmulas se satisfagan en todo el modelo:

$$\begin{array}{cc} \mathit{oso} \vee \mathit{humano} & \mathit{oso} \to \langle \textsc{madre} \rangle \mathit{oso} \\ \mathit{oso} \to \neg \mathit{humano} & \mathit{oso} \to [\textsc{alimentado}](\mathit{cuidador} \vee \mathit{madre}) \end{array}$$

- Y, por otro lado, poder preguntar si existirá algún estado donde:

Extensiones: Modalidad Universal

- Otro “feature” que podemos querer es la capacidad de describir alguna propiedad global, que debe cumplirse en todo el modelo.
- Supongamos que estamos modelando un zoológico, y que estamos interesados en la alimentación de los osos y su relación con los cuidadores de osos.
- Queremos, por un lado, que estas fórmulas se satisfagan en todo el modelo:

$$\begin{array}{cc} oso \vee humano & oso \to \langle \textsc{madre} \rangle oso \\ oso \to \neg humano & oso \to [\textsc{alimentado}](cuidador \vee madre) \end{array}$$

- Y, por otro lado, poder preguntar si existirá algún estado donde:
  - $oso \wedge \langle \textsc{madre} \rangle (oso \wedge humano)$

Extensiones: Modalidad Universal

- Otro “feature” que podemos querer es la capacidad de describir alguna propiedad global, que debe cumplirse en todo el modelo.
- Supongamos que estamos modelando un zoológico, y que estamos interesados en la alimentación de los osos y su relación con los cuidadores de osos.
- Queremos, por un lado, que estas fórmulas se satisfagan en todo el modelo:

$$\begin{array}{cc} oso \lor humano & oso \to \langle \textsc{madre} \rangle oso \\ oso \to \neg humano & oso \to [\textsc{alimentado}](cuidador \lor madre) \end{array}$$

- Y, por otro lado, poder preguntar si existirá algún estado donde:
  - $oso \land \langle \textsc{madre} \rangle (oso \land humano)$
  - $oso \land \langle \textsc{alimentado} \rangle (\neg madre \land \neg humano)$

Extensiones: Modalidad Universal

- Para poder decir esto, tenemos que agregar la modalidad universal, que notamos como A.

Extensiones: Modalidad Universal

- Para poder decir esto, tenemos que agregar la modalidad universal, que notamos como A.
- La fórmula $A\varphi$ significa que φ es cierta en todos los puntos del modelo.

Extensiones: Modalidad Universal

- Para poder decir esto, tenemos que agregar la modalidad universal, que notamos como A.
- La fórmula $A\varphi$ significa que φ es cierta en todos los puntos del modelo.
- ¿Cómo es la definición formal de la semántica de este operador?

Extensiones: Modalidad Universal

- Para poder decir esto, tenemos que agregar la modalidad universal, que notamos como A.
- La fórmula $A\varphi$ significa que φ es cierta en todos los puntos del modelo.
- ¿Cómo es la definición formal de la semántica de este operador?
- Y una pregunta (para pensar), ¿este operador es lo mismo que el $\forall$ de LPO?

Extensiones: PDL (Propositional Dynamic Logic)

- Imaginemos que queremos modelar el comportamiento de programas.

Extensiones: PDL (Propositional Dynamic Logic)

- Imaginemos que queremos modelar el comportamiento de programas.
- Para eso, para cada programa (no determinístico) π vamos a tener una modalidad $\langle\pi\rangle$ (tenemos infinitas modalidades!).

Extensiones: PDL (Propositional Dynamic Logic)

- Imaginemos que queremos modelar el comportamiento de programas.
- Para eso, para cada programa (no determinístico) π vamos a tener una modalidad $\langle\pi\rangle$ (tenemos infinitas modalidades!).
- La interpretación de $\langle\pi\rangle\varphi$ va a ser: 'alguna ejecución que termina de π desde el estado actual nos lleva a un estado donde vale φ'.

Extensiones: PDL (Propositional Dynamic Logic)

- Imaginemos que queremos modelar el comportamiento de programas.
- Para eso, para cada programa (no determinístico) π vamos a tener una modalidad $\langle \pi \rangle$ (tenemos infinitas modalidades!).
- La interpretación de $\langle \pi \rangle \varphi$ va a ser: 'alguna ejecución que termina de π desde el estado actual nos lleva a un estado donde vale φ'.
- Lo que queremos también es hacer explícita la estructura inductiva de los programas: los programas se pueden componer, iterar, etc, formando nuevos programas.

Extensiones: PDL (Propositional Dynamic Logic)

Si tenemos programas 'simples' a, b, c, etc. (y por lo tanto tenemos las modalidades $\langle a \rangle, \langle b \rangle, \langle c \rangle, \ldots$) podemos construir programas más complejos de la siguiente manera:

Extensiones: PDL (Propositional Dynamic Logic)

Si tenemos programas 'simples' a, b, c, etc. (y por lo tanto tenemos las modalidades $\langle a\rangle, \langle b\rangle, \langle c\rangle, \ldots$) podemos construir programas más complejos de la siguiente manera:

- (*elección*) Si π_1 y π_2 son programas, entonces $\pi_1 \cup \pi_2$ es un programa. El programa $\pi_1 \cup \pi_2$ ejecuta no determinísticamente π_1 ó π_2.

Extensiones: PDL (Propositional Dynamic Logic)

Si tenemos programas 'simples' a, b, c, etc. (y por lo tanto tenemos las modalidades $\langle a \rangle, \langle b \rangle, \langle c \rangle, \ldots$) podemos construir programas más complejos de la siguiente manera:

- (*elección*) Si π_1 y π_2 son programas, entonces $\pi_1 \cup \pi_2$ es un programa. El programa $\pi_1 \cup \pi_2$ ejecuta no determinísticamente π_1 ó π_2.
- (*composición*) Si π_1 y π_2 son programas, entonces $\pi_1; \pi_2$ es un programa. El programa $\pi_1; \pi_2$ ejecuta primero π_1 y luego π_2.

Extensiones: PDL (Propositional Dynamic Logic)

Si tenemos programas 'simples' a, b, c, etc. (y por lo tanto tenemos las modalidades $\langle a \rangle, \langle b \rangle, \langle c \rangle, \ldots$) podemos construir programas más complejos de la siguiente manera:

- (*elección*) Si π_1 y π_2 son programas, entonces $\pi_1 \cup \pi_2$ es un programa. El programa $\pi_1 \cup \pi_2$ ejecuta no determinísticamente π_1 ó π_2.
- (*composición*) Si π_1 y π_2 son programas, entonces $\pi_1; \pi_2$ es un programa. El programa $\pi_1; \pi_2$ ejecuta primero π_1 y luego π_2.
- (*iteración*) Si π es un programa, entonces π^* es un programa. El programa π^* ejecuta π un número finito (o quizás cero) de veces.

Extensiones: PDL (Propositional Dynamic Logic)

Si tenemos programas ‘simples’ a, b, c, etc. (y por lo tanto tenemos las modalidades $\langle a\rangle, \langle b\rangle, \langle c\rangle, \ldots$) podemos construir programas más complejos de la siguiente manera:

- (*elección*) Si π_1 y π_2 son programas, entonces $\pi_1 \cup \pi_2$ es un programa. El programa $\pi_1 \cup \pi_2$ ejecuta no determinísticamente π_1 ó π_2.
- (*composición*) Si π_1 y π_2 son programas, entonces $\pi_1; \pi_2$ es un programa. El programa $\pi_1; \pi_2$ ejecuta primero π_1 y luego π_2.
- (*iteración*) Si π es un programa, entonces π^* es un programa. El programa π^* ejecuta π un número finito (o quizás cero) de veces.

Esto significa que si $\langle \pi_1\rangle$ y $\langle \pi_2\rangle$ son modalidades, entonces también lo son $\langle \pi_1 \cup \pi_2\rangle$, $\langle \pi_1; \pi_2\rangle$ y $\langle \pi^*\rangle$.

Extensiones: PDL (Propositional Dynamic Logic)

- Tenemos que interpretar apropiadamente las modalidades para que cada operador tenga el significado esperado.

Extensiones: PDL (Propositional Dynamic Logic)

- Tenemos que interpretar apropiadamente las modalidades para que cada operador tenga el significado esperado.
- Definamos

$$\begin{array}{rcl} R_{\pi_1 \cup \pi_2} & = & R_{\pi_1} \cup R_{\pi_2} \\ R_{\pi_1 ; \pi_2} & = & R_{\pi_1} \circ R_{\pi_2} \text{ (la composición de las relaciones)} \\ R_{\pi^*} & = & (R_{\pi_1})^* \text{ (la clausura reflexo-transitiva de } R_{\pi_1}) \end{array}$$

Extensiones: PDL (Propositional Dynamic Logic)

- Tenemos que interpretar apropiadamente las modalidades para que cada operador tenga el significado esperado.
- Definamos

$$
\begin{array}{rcl}
R_{\pi_1 \cup \pi_2} & = & R_{\pi_1} \cup R_{\pi_2} \\
R_{\pi_1 ; \pi_2} & = & R_{\pi_1} \circ R_{\pi_2} \text{ (la composición de las relaciones)} \\
R_{\pi^*} & = & (R_{\pi_1})^* \text{ (la clausura reflexo-transitiva de } R_{\pi_1})
\end{array}
$$

- Entonces,

$$\mathcal{M}, w \models \langle \pi \rangle \varphi \text{ sii } \exists w'.R_\pi(w, w') \text{ y } \mathcal{M}, w' \models \varphi$$

Extensiones: PDL (Propositional Dynamic Logic)

- Con estas definiciones, ¿qué nos dice esta fórmula?

$$\langle \pi^* \rangle \varphi \leftrightarrow \varphi \vee \langle \pi ; \pi^* \rangle \varphi$$

Extensiones: PDL (Propositional Dynamic Logic)

- Con estas definiciones, ¿qué nos dice esta fórmula?

$$\langle \pi^* \rangle \varphi \leftrightarrow \varphi \vee \langle \pi ; \pi^* \rangle \varphi$$

- ¿Debería ser válida?

Extensiones: PDL (Propositional Dynamic Logic)

- Con estas definiciones, ¿qué nos dice esta fórmula?

$$\langle \pi^* \rangle \varphi \leftrightarrow \varphi \vee \langle \pi ; \pi^* \rangle \varphi$$

- ¿Debería ser válida? Demostrarlo!

Extensiones: PDL (Propositional Dynamic Logic)

- Con estas definiciones, ¿qué nos dice esta fórmula?

$$\langle \pi^* \rangle \varphi \leftrightarrow \varphi \vee \langle \pi ; \pi^* \rangle \varphi$$

- ¿Debería ser válida? Demostrarlo!
- ¿Y esta?

$$[\pi^*](\varphi \rightarrow [\pi]\varphi) \rightarrow (\varphi \rightarrow [\pi^*]\varphi)$$

Extensiones: PDL (Propositional Dynamic Logic)

- Con estas definiciones, ¿qué nos dice esta fórmula?

$$\langle \pi^* \rangle \varphi \leftrightarrow \varphi \vee \langle \pi ; \pi^* \rangle \varphi$$

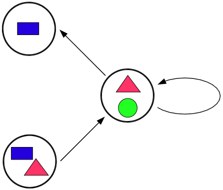
- ¿Debería ser válida? Demostrarlo!
- ¿Y esta?

$$[\pi^*](\varphi \to [\pi]\varphi) \to (\varphi \to [\pi^*]\varphi)$$

- Pensar en el esquema de inducción…

Extensiones: Lógica Híbrida

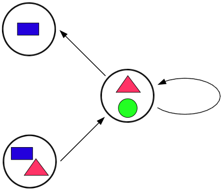
Volvamos al ejemplo de las figuras geométricas. ¿Cómo hacemos para decir?



- ¿Puede w_1 verse a sí mismo?
- ¿Son realmente w_1 y w_2 estados distintos?

Extensiones: Lógica Híbrida

Volvamos al ejemplo de las figuras geométricas. ¿Cómo hacemos para decir?



- ¿Puede w_1 verse a sí mismo?
- ¿Son realmente w_1 y w_2 estados distintos?

- Lo que necesitamos para expresar esto es poder nombrar mundos, y una noción de igualdad.

Extensiones: Lógica Híbrida

$\mathcal{HL}(@)$ es la extensión de la lógica modal básica con:

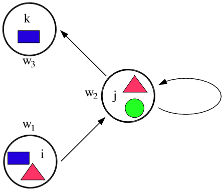
- nombres (nominales): son un nuevo conjunto de símbolos atómicos, que representan a los estados. La clave es que cada nominal tiene que ser cierto en un único estado. En general se escriben como $i, j, k, \ldots$
- @: el operador 'at'. $@_i\varphi$ es verdadera sii φ es verdadera en el mundo denotado por i.

Extensiones: Lógica Híbrida

$\mathcal{HL}(@)$ es la extensión de la lógica modal básica con:

- nombres (nominales): son un nuevo conjunto de símbolos atómicos, que representan a los estados. La clave es que cada nominal tiene que ser cierto en un único estado. En general se escriben como $i, j, k, \ldots$
- @: el operador 'at'. $@_i\varphi$ es verdadera sii φ es verdadera en el mundo denotado por i.

Ahora podemos expresar...



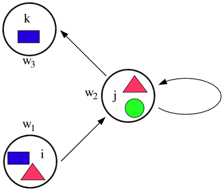
- ¿Puede w_1 verse a sí mismo? $@_i\langle see\rangle i$

Extensiones: Lógica Híbrida

$\mathcal{HL}(@)$ es la extensión de la lógica modal básica con:

- nombres (nominales): son un nuevo conjunto de símbolos atómicos, que representan a los estados. La clave es que cada nominal tiene que ser cierto en un único estado. En general se escriben como $i, j, k, \ldots$
- @: el operador 'at'. $@_i\varphi$ es verdadera sii φ es verdadera en el mundo denotado por i.

Ahora podemos expresar...



- ¿Puede w_1 verse a sí mismo?
  $@_i\langle see\rangle i$
- ¿Son realmente w_1 y w_2 estados distintos?
  $@_i\neg j$

Extensiones: Lógica Híbrida

Entonces, la definición de la sintaxis es:

- A la signatura $\langle \textsc{prop}, \textsc{rel} \rangle$ que teníamos antes, le tenemos que agregar un nuevo conjunto $\textsc{nom} = \{i, j, k, \ldots\}$ de nominales.

Extensiones: Lógica Híbrida

Entonces, la definición de la sintaxis es:

- A la signatura $\langle \text{PROP}, \text{REL} \rangle$ que teníamos antes, le tenemos que agregar un nuevo conjunto $\text{NOM} = \{i, j, k, \ldots\}$ de nominales.
- Las fórmulas bien formadas ahora son:

$$\text{FORM} := p \mid i \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle \varphi \mid @_i \varphi$$

en donde $p \in \text{PROP}$, $R \in \text{REL}$, $i \in \text{NOM}$ y $\varphi, \psi \in \text{FORM}$.

Extensiones: Lógica Híbrida

Entonces, la definición de la sintaxis es:

- A la signatura $\langle \text{PROP}, \text{REL} \rangle$ que teníamos antes, le tenemos que agregar un nuevo conjunto $\text{NOM} = \{i, j, k, \ldots\}$ de nominales.
- Las fórmulas bien formadas ahora son:

$$\text{FORM} := p \mid i \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle \varphi \mid @_i \varphi$$

en donde $p \in \text{PROP}$, $R \in \text{REL}$, $i \in \text{NOM}$ y $\varphi, \psi \in \text{FORM}$.

- ¿Hay que hacer algún cambio en lo modelos? ¿Cómo es la semántica de $\mathcal{HL}(@)$?

Extensiones: Lógica Híbrida

Entonces, la definición de la sintaxis es:

- A la signatura $\langle \text{PROP}, \text{REL} \rangle$ que teníamos antes, le tenemos que agregar un nuevo conjunto $\text{NOM} = \{i, j, k, \ldots\}$ de nominales.
- Las fórmulas bien formadas ahora son:

$$\text{FORM} := p \mid i \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R \rangle \varphi \mid @_i \varphi$$

  en donde $p \in \text{PROP}$, $R \in \text{REL}$, $i \in \text{NOM}$ y $\varphi, \psi \in \text{FORM}$.
- ¿Hay que hacer algún cambio en lo modelos? ¿Cómo es la semántica de $\mathcal{HL}(@)$?
- Ejercicio!