Lógicas Modales: Clásicas y Dinámicas

Raúl A. Fervari

Supervisor: Carlos Areces

Primer año de Doctorado (Desde mayo)

fervari@famaf.unc.edu.ar

Oficina 403

Grupo de Procesamiento de Lenguaje Natural
Facultad de Matemática, Astronomía y Física
Universidad Nacional de Córdoba

4 de diciembre de 2010

Área de interés

En que trabajamos?

Nuestro principal área de estudio es la lógica, vista como una disciplina que estudia diferentes lenguajes formales teniendo en cuenta aspectos semánticos, sintácticos, de teoría de modelos y computacionales.

Área de interés

En que trabajamos?

Nuestro principal área de estudio es la lógica, vista como una disciplina que estudia diferentes lenguajes formales teniendo en cuenta aspectos semánticos, sintácticos, de teoría de modelos y computacionales.

Lógica Computacional

Particularmente trabajamos en lógica computacional: no nos interesan solamente los "por qué?", sino que prestamos especial atención a los "cómo?", los "para qué?" y los "cuánto cuesta?".

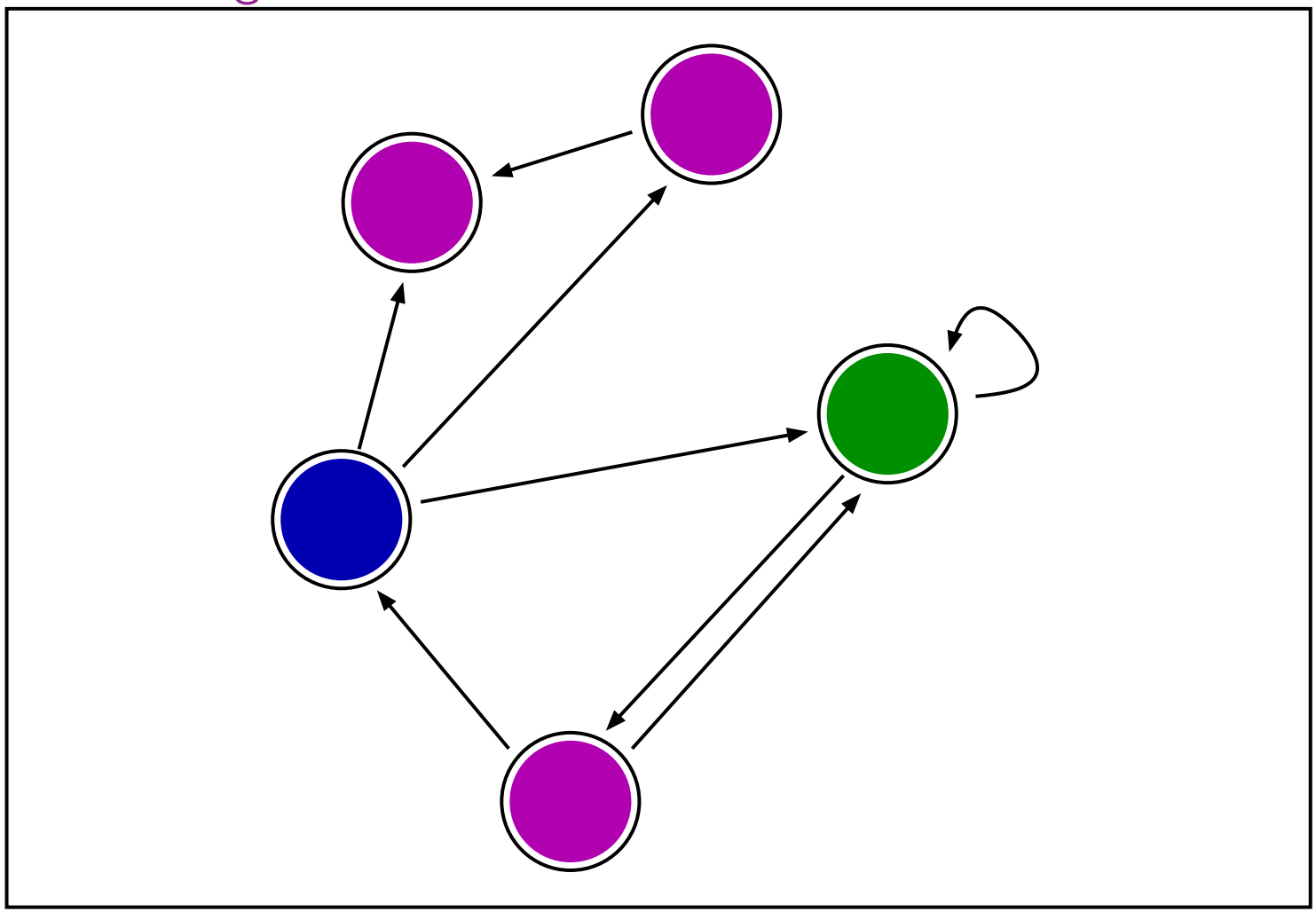
Lógicas Modales: Estructuras Simples / Lenguajes Simples

Lógicas Modales: Estructuras Simples / Lenguajes Simples

Pensemos en un grafo coloreado:

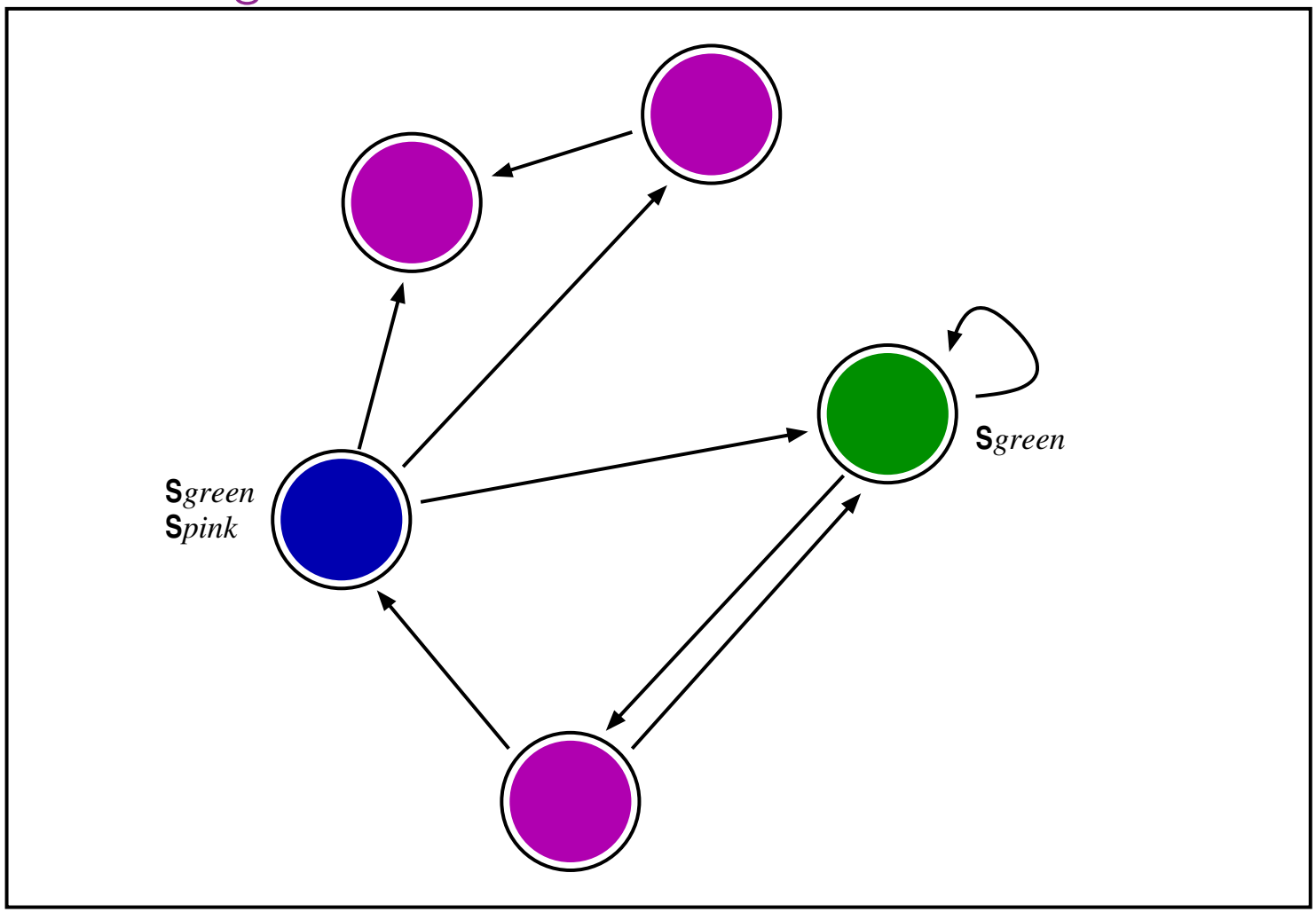
Lógicas Modales: Estructuras Simples / Lenguajes Simples

Pensemos en un grafo coloreado:



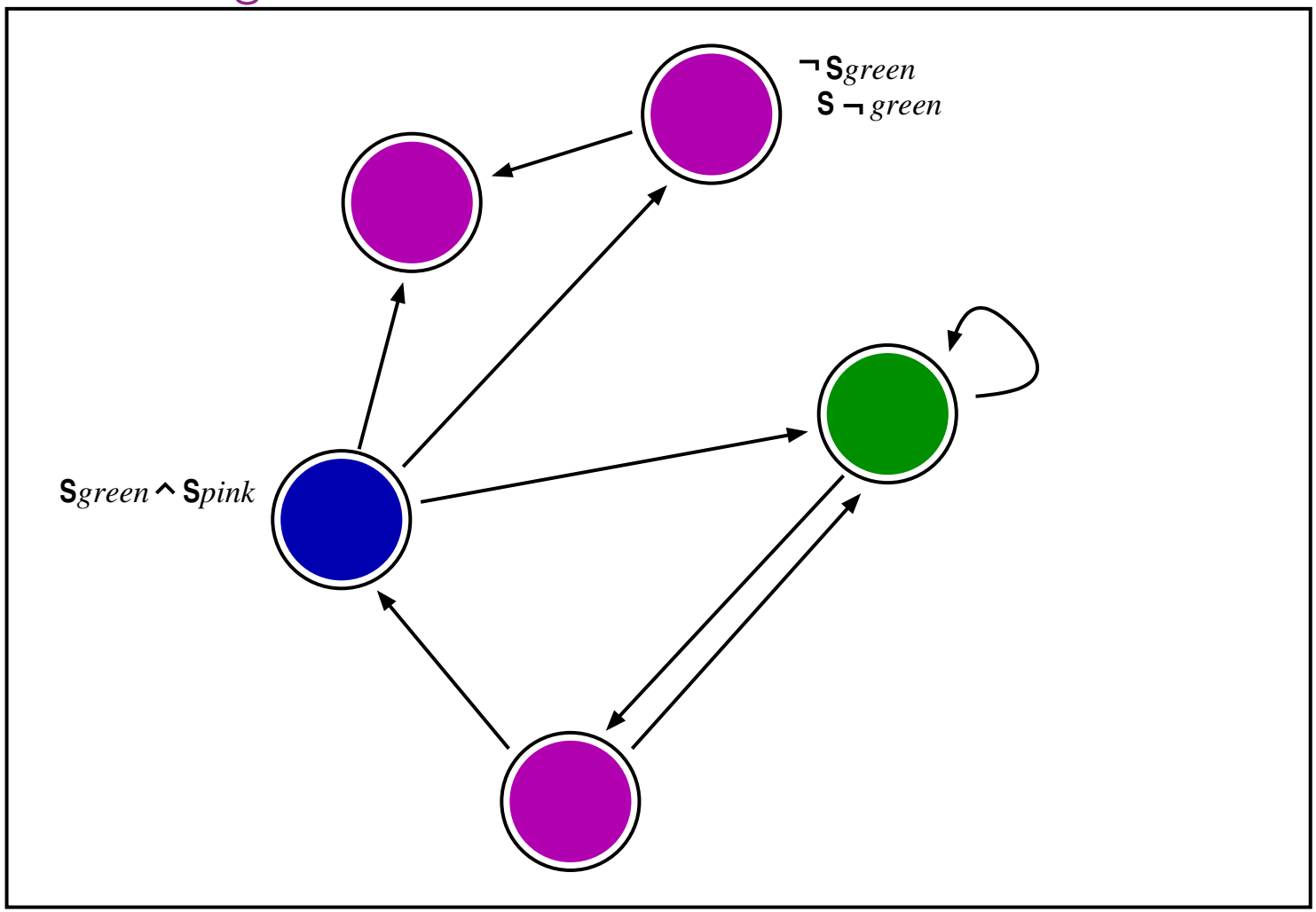
Lógicas Modales: Estructuras Simples / Lenguajes Simples

Pensemos en un grafo coloreado:



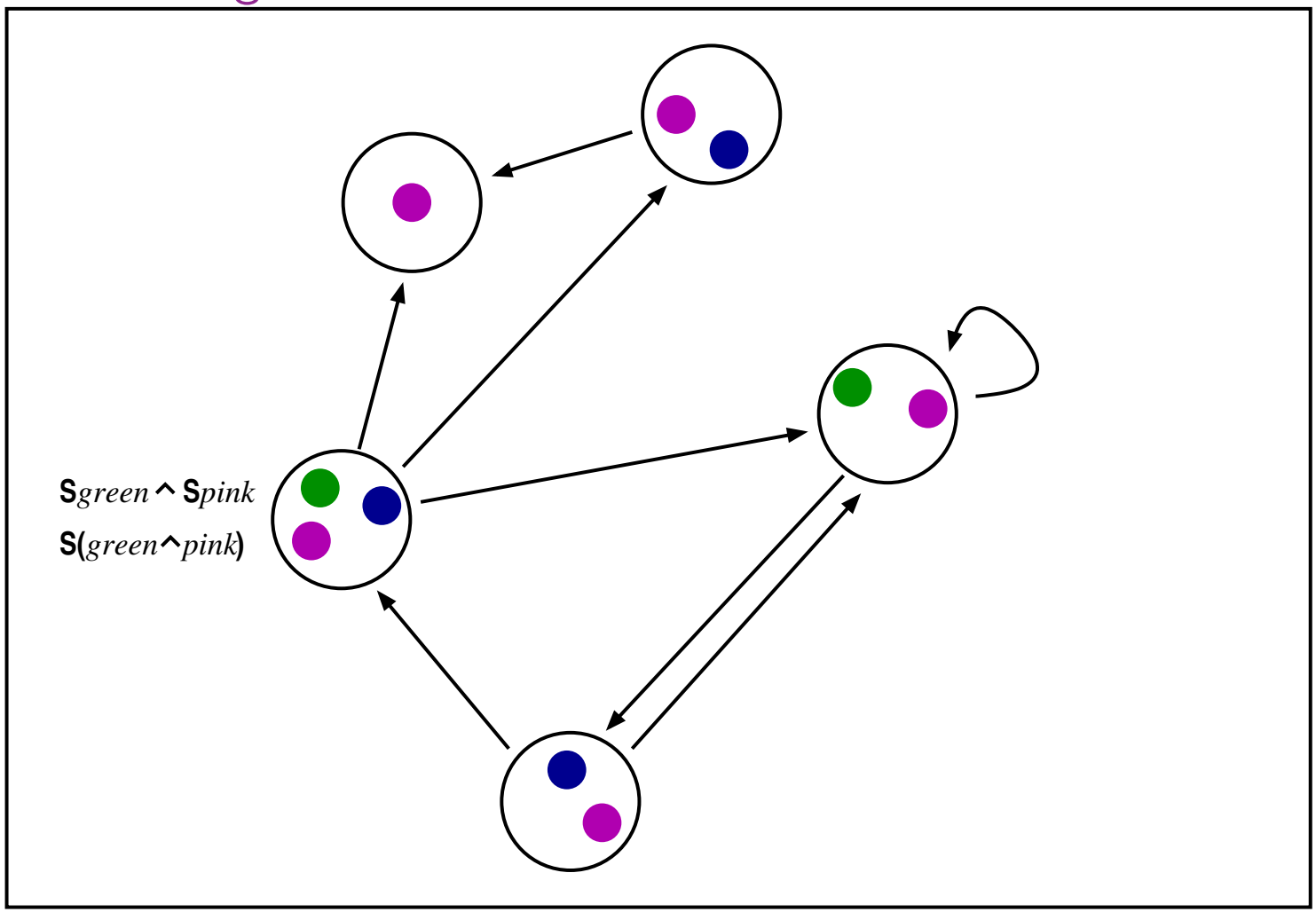
Lógicas Modales: Estructuras Simples / Lenguajes Simples

Pensemos en un grafo coloreado:



Lógicas Modales: Estructuras Simples / Lenguajes Simples

Pensemos en un grafo coloreado:



Lógicas Modales: Estructuras Simples / Lenguajes Simples

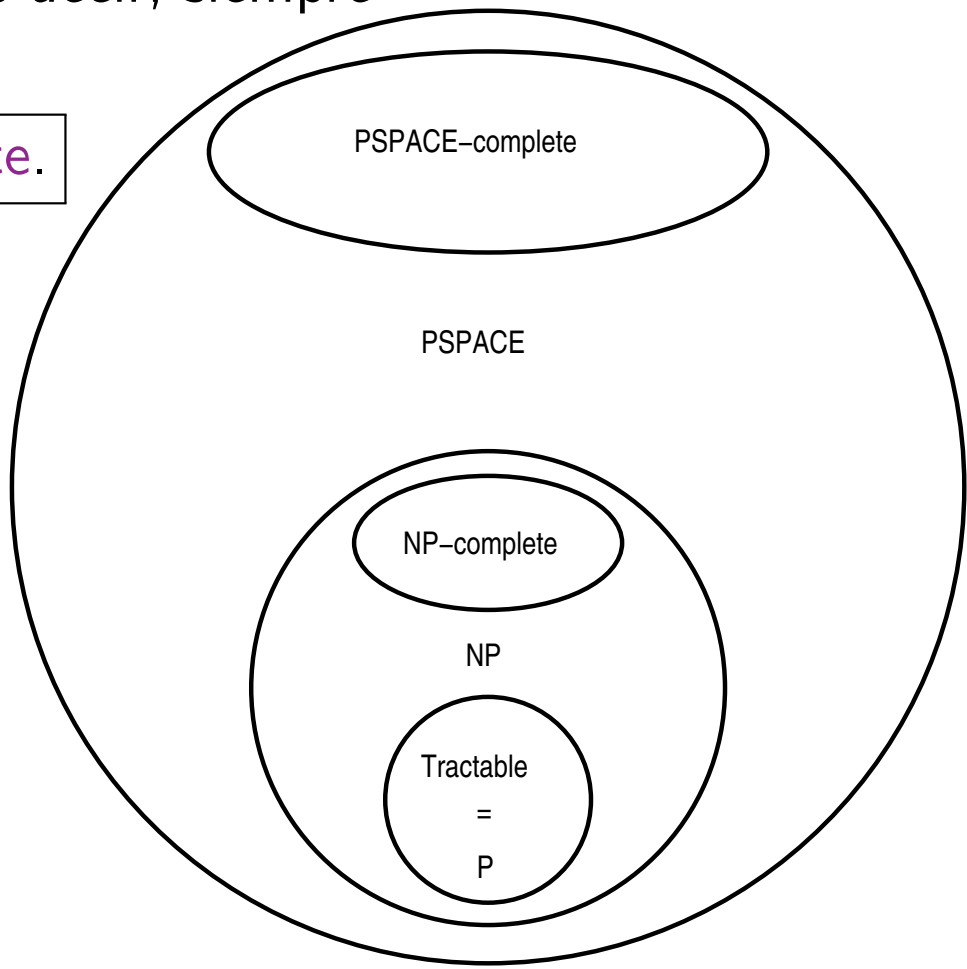
- Aún para este lenguaje muy simple, decidir si una fórmula es un teorema (es decir, siempre cierta) es

  PSPACE-complete.

Lógicas Modales: Estructuras Simples / Lenguajes Simples

- Aún para este lenguaje muy simple, decidir si una fórmula es un teorema (es decir, siempre cierta) es

PSPACE-complete.



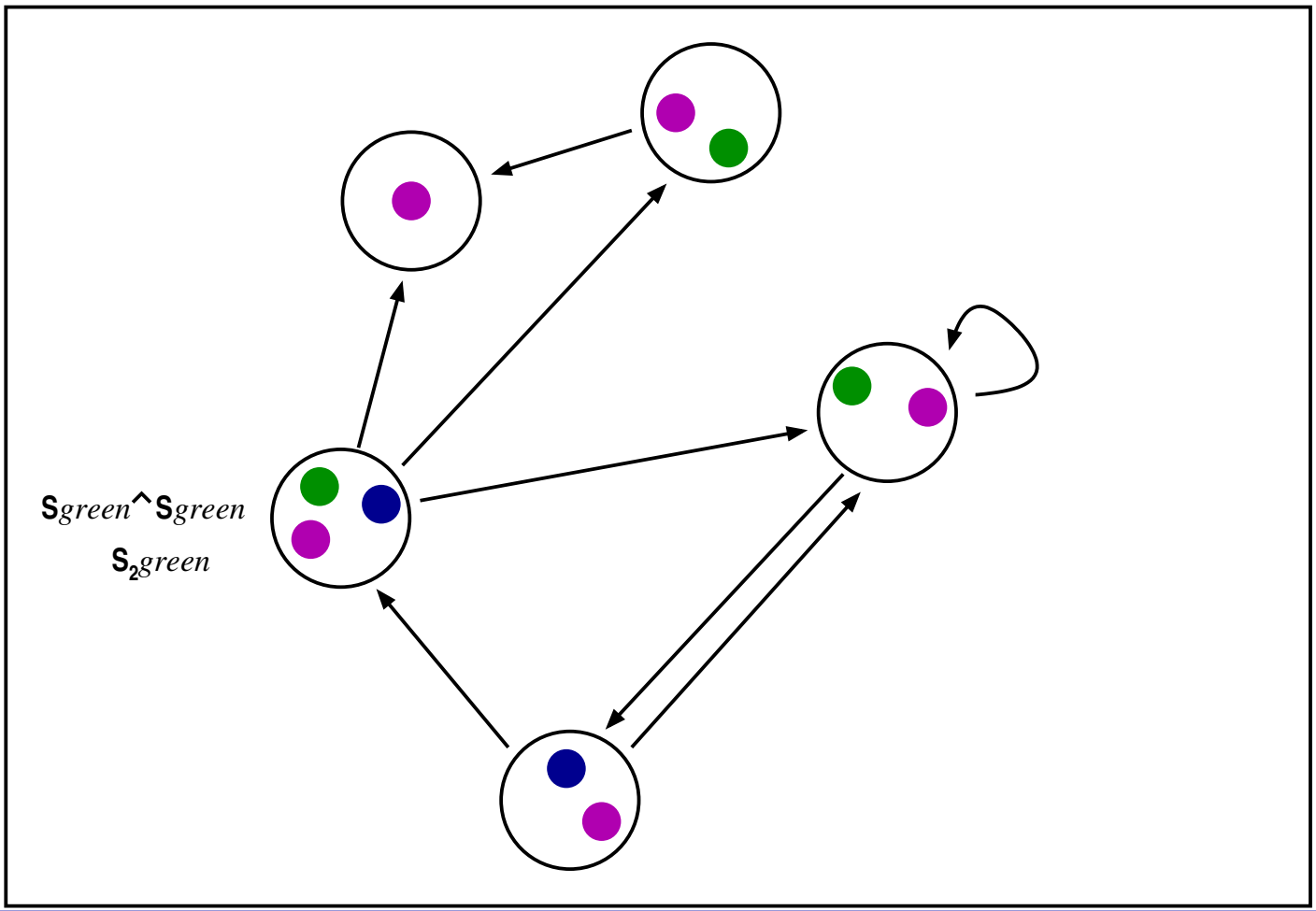
Otros Lenguajes

Otros Lenguajes

- A veces nos interesa definir lenguajes que nos permitan decir más cosas.

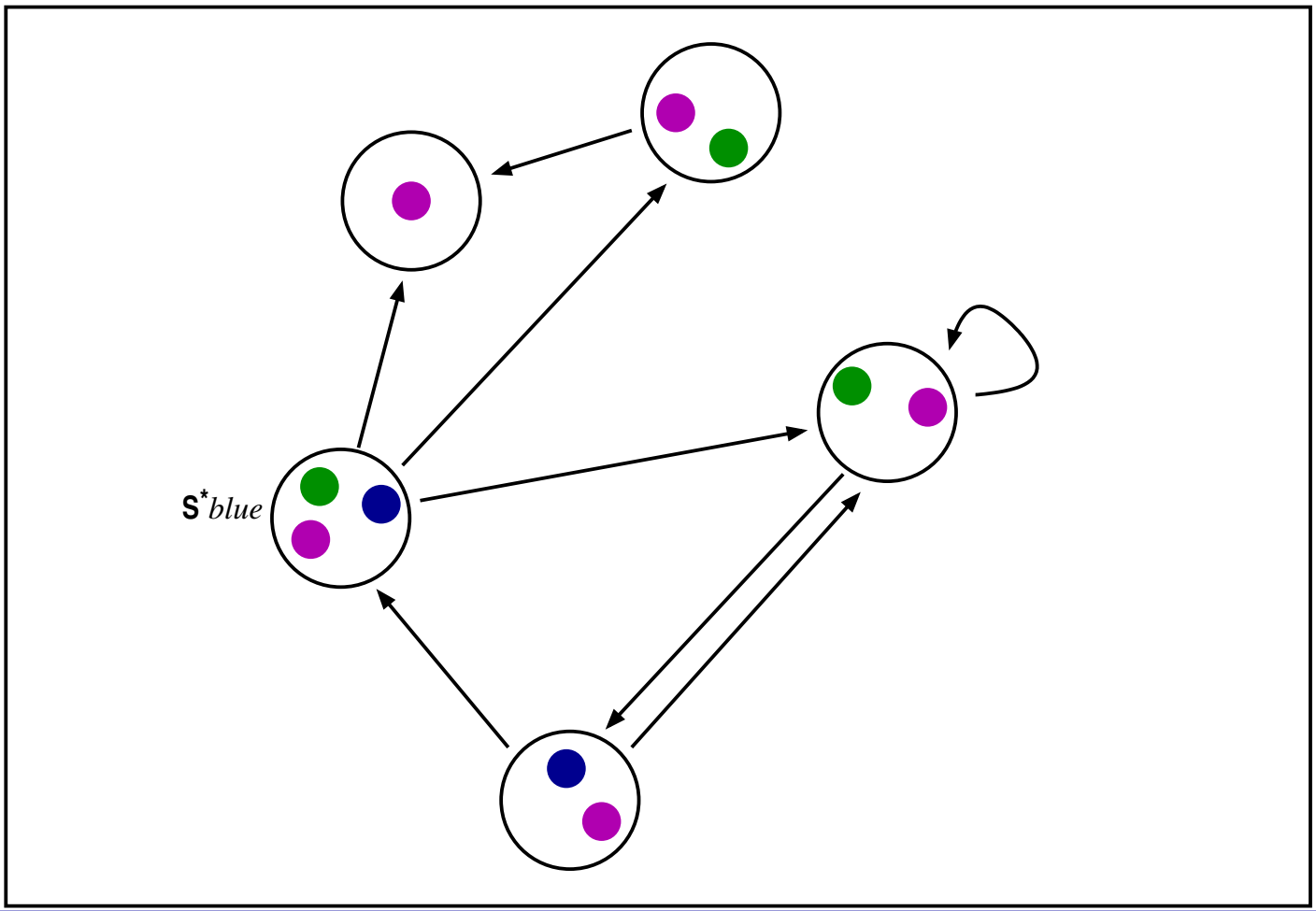
Otros Lenguajes

- A veces nos interesa definir lenguajes que nos permitan decir más cosas.



Otros Lenguajes

- A veces nos interesa definir lenguajes que nos permitan decir más cosas.



Otros Lenguajes

- A veces nos interesa definir lenguajes que nos permitan decir más cosas.
- En nuestro área, investigamos todos aquellos lenguajes que nos sirven para describir estructuras relacionales.

Otros Lenguajes

- A veces nos interesa definir lenguajes que nos permitan decir más cosas.
- En nuestro área, investigamos todos aquellos lenguajes que nos sirven para describir estructuras relacionales.
- La preguntas más importantes son:
  - Podemos definir algoritmos de inferencia para estos lenguajes?
  - Cual es la complejidad de estos algoritmos?
  - Cuáles son los límites de expresividad de estos lenguajes?

Otros Lenguajes

- A veces nos interesa definir lenguajes que nos permitan decir más cosas.
- En nuestro área, investigamos todos aquellos lenguajes que nos sirven para describir estructuras relacionales.
- La preguntas más importantes son:
  - Podemos definir algoritmos de inferencia para estos lenguajes?
  - Cual es la complejidad de estos algoritmos?
  - Cuáles son los límites de expresividad de estos lenguajes?
- La tarea no es fácil, porque los distintos operadores del lenguaje pueden interactuar de formas inesperadas.
  Por ejemplo, agregar el operador S^* transforma el problema de satisfacibilidad del lenguaje en EXPTIME-complete!

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.
  - Criptografía.
  - Inteligencia Artificial.
  - Filosofía.
  - Linguística Computacional.
  - 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.
  - Criptografía.
  - Inteligencia Artificial.
  - Filosofía.
  - Linguística Computacional.
  - 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.
  - Criptografía.
  - Inteligencia Artificial.
  - Filosofía.
  - Linguística Computacional.
  - Epistemología.
  - ...
- **Por qué?** Muchas cosas pueden ser representadas como grafos!! (o sea, 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.
  - Criptografía.
  - Inteligencia Artificial.
  - Filosofía.
  - Linguística Computacional.
  - Epistemología.
  - ...
- **Por qué?** Muchas cosas pueden ser representadas como grafos!! (o sea, estructuras relacionales).
  Y antes dijimos que los lenguajes modales están diseñados especialmente para razonar y describir propiedades de grafos.

Lógica Modal Básica

Lógica Modal Básica

Las fórmulas del lenguaje modal básico se contruyen a partir de:

- El lenguaje proposicional: p, q, $\wedge$, $\neg$, etc.
- Los operadores modales: $\Diamond$, $\Box$.

Lógica Modal Básica

Las fórmulas del lenguaje modal básico se contruyen a partir de:

- El lenguaje proposicional: p, q, $\wedge$, $\neg$, etc.
- Los operadores modales: $\Diamond$, $\Box$.

Las fórmulas son interpretadas sobre modelos relacionales:

$$\mathcal{M} = \langle W, R, V \rangle$$

- W es un conjunto no vacío de elementos.
- R es una relación binaria sobre W.
- $V : \mathrm{PROP} \to \wp(W)$ el conjunto de proposiciones que son verdaderas en cada estado.

Lógica Modal Básica

Las fórmulas del lenguaje modal básico se contruyen a partir de:

- El lenguaje proposicional: p, q, $\wedge$, $\neg$, etc.
- Los operadores modales: $\Diamond$, $\Box$.

Las fórmulas son interpretadas sobre modelos relacionales:

$$\mathcal{M} = \langle W, R, V \rangle$$

- W es un conjunto no vacío de elementos.
- R es una relación binaria sobre W.
- $V : \mathrm{PROP} \to \wp(W)$ el conjunto de proposiciones que son verdaderas en cada estado.

Intuitivamente, un modelo es un grafo dirigido etiquetado.

La Relación de Satisfacción

La Relación de Satisfacción

- Dado un modelo $\mathcal{M}$, un estado w en $\mathcal{M}$ y una fórmula φ escribimos $\mathcal{M}, w \models \varphi$ para decir que w satisface φ en $\mathcal{M}$.

La Relación de Satisfacción

- Dado un modelo $\mathcal{M}$, un estado w en $\mathcal{M}$ y una fórmula φ escribimos $\mathcal{M}, w \models \varphi$ para decir que w satisface φ en $\mathcal{M}$.
- Los operadores proposicionales se interpretan como siempre:

$$\mathcal{M}, w \models \varphi \wedge \varphi' \text{: si } w \text{ satisface ambas } \varphi \text{ y } \varphi' \quad .$$

La Relación de Satisfacción

- Dado un modelo $\mathcal{M}$, un estado w en $\mathcal{M}$ y una fórmula φ escribimos $\mathcal{M}, w \models \varphi$ para decir que w satisface φ en $\mathcal{M}$.
- Los operadores proposicionales se interpretan como siempre:

  $\mathcal{M}, w \models \varphi \wedge \varphi'$: si w satisface ambas φ y φ' .
- $\mathcal{M}, w \models \Diamond\varphi$: φ se satisface en algún sucesor de w.

La Relación de Satisfacción

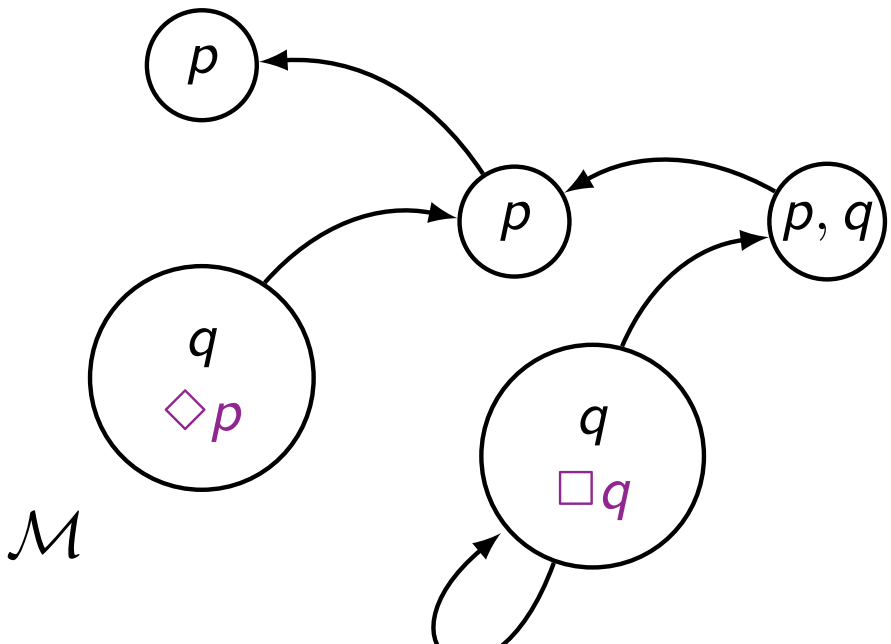
- Dado un modelo $\mathcal{M}$, un estado w en $\mathcal{M}$ y una fórmula φ escribimos $\mathcal{M}, w \models \varphi$ para decir que w satisface φ en $\mathcal{M}$.
- Los operadores proposicionales se interpretan como siempre:

  $\mathcal{M}, w \models \varphi \wedge \varphi'$: si w satisface ambas φ y φ' .
- $\mathcal{M}, w \models \Diamond\varphi$: φ se satisface en algún sucesor de w.
- $\mathcal{M}, w \models \Box\varphi$: φ se satisface en todos los sucesores de w.

La Relación de Satisfacción

- Dado un modelo $\mathcal{M}$, un estado w en $\mathcal{M}$ y una fórmula φ escribimos $\mathcal{M}, w \models \varphi$ para decir que w satisface φ en $\mathcal{M}$.
- Los operadores proposicionales se interpretan como siempre:

  $\mathcal{M}, w \models \varphi \wedge \varphi'$: si w satisface ambas φ y φ' .
- $\mathcal{M}, w \models \Diamond\varphi$: φ se satisface en algún sucesor de w.
- $\mathcal{M}, w \models \Box\varphi$: φ se satisface en todos los sucesores de w.



Qué sabemos hasta ahora?

Qué sabemos hasta ahora?

- La lógica modal básica nos da una perspectiva de las características internas del grafo.
- Las fórmulas nos permiten, a partir de un estado dado, explorar el modelo y sus propiedades.
- Puede ser interesante no solo explorar el modelo, sino ver como reacciona ante cambios.

Qué sabemos hasta ahora?

- La lógica modal básica nos da una perspectiva de las características internas del grafo.
- Las fórmulas nos permiten, a partir de un estado dado, explorar el modelo y sus propiedades.
- Puede ser interesante no solo explorar el modelo, sino ver como reacciona ante cambios.
- Existen diversos lenguajes que explotan estas ideas:
  - Memory Logic.
  - Swap Logic.

Memory Logics

Memory Logics

- Son lógicas que pueden memorizar y luego consultar información acerca de los estados del modelo.

Memory Logics

- Son lógicas que pueden memorizar y luego consultar información acerca de los estados del modelo.
- Extendamos un modelo $\mathcal{M} = \langle W, R, V \rangle$ con una memoria M: $\mathcal{M} = \langle W, R, V, M \rangle$, $M \subseteq W$.

Memory Logics

- Son lógicas que pueden memorizar y luego consultar información acerca de los estados del modelo.
- Extendamos un modelo $\mathcal{M} = \langle W, R, V \rangle$ con una memoria M: $\mathcal{M} = \langle W, R, V, M \rangle$, $M \subseteq W$.
- Sea $\mathcal{M}[w] = \langle W, R, V, M \cup \{w\} \rangle$ definimos los siguientes operadores:

  remember $\quad \mathcal{M}, w \models ⓡ\varphi$ sii $\mathcal{M}[w], w \models \varphi$.

Memory Logics

- Son lógicas que pueden memorizar y luego consultar información acerca de los estados del modelo.
- Extendamos un modelo $\mathcal{M} = \langle W, R, V \rangle$ con una memoria M: $\mathcal{M} = \langle W, R, V, M \rangle$, $M \subseteq W$.
- Sea $\mathcal{M}[w] = \langle W, R, V, M \cup \{w\} \rangle$ definimos los siguientes operadores:

remember	$\mathcal{M}, w \models ⓡ\varphi$ sii $\mathcal{M}[w], w \models \varphi$.
known	$\mathcal{M}, w \models ⓚ$ sii $w \in M$.

Memory Logics

Memory Logics

- El operador unario ⓡ (remember), marca el estado corriente como visitado, y lo almacena en la memoria.
- El operador cero-ario ⓚ (known), verifica si el estado corriente ha sido guardado en memoria.

Memory Logics

- El operador unario $\textcircled{\texttt{r}}$ (remember), marca el estado corriente como visitado, y lo almacena en la memoria.
- El operador cero-ario $\textcircled{\texttt{k}}$ (known), verifica si el estado corriente ha sido guardado en memoria.
- Consideremos la fórmula

$$\textcircled{\texttt{r}}\Diamond\textcircled{\texttt{k}}.$$

Memory Logics

- El operador unario ⓡ (remember), marca el estado corriente como visitado, y lo almacena en la memoria.
- El operador cero-ario ⓚ (known), verifica si el estado corriente ha sido guardado en memoria.
- Consideremos la fórmula

$$ⓡ \Diamond ⓚ.$$

La fórmula es satisfecha por un estado w del modelo $\langle W, R, V, \emptyset \rangle$ sii $R(w, w)$.

Memory Logics

- El operador unario Ⓡ (remember), marca el estado corriente como visitado, y lo almacena en la memoria.
- El operador cero-ario Ⓚ (known), verifica si el estado corriente ha sido guardado en memoria.
- Consideremos la fórmula

$$\textcircled{r}\Diamond\textcircled{k}.$$

La fórmula es satisfecha por un estado w del modelo $\langle W, R, V, \emptyset\rangle$ sii $R(w, w)$.
No existe una fórmula equivalente en lógica modal básica (las memory logics son estrictamente más expresivas que la lógica modal básica).

Swap Logic

Swap Logic

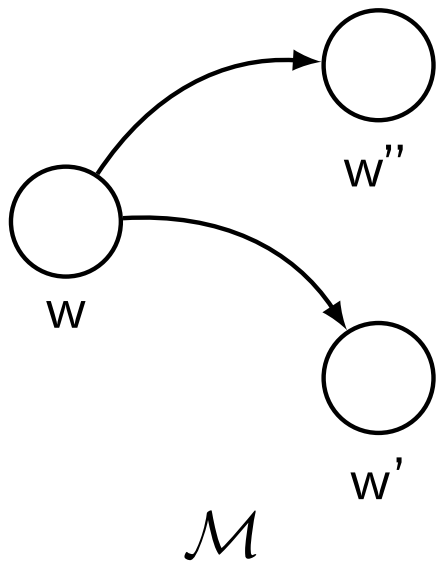
- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.

Swap Logic

- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.
- Sea $\mathcal{M} = \langle W, R, V \rangle$ y w, w' elementos de W tal que $R(w, w')$. Denotaremos $\mathcal{M}[w' \mapsto w]$ el modelo idéntico a $\mathcal{M}$, cambiando a R por $(R \setminus (w, w')) \cup \{(w', w)\}$.

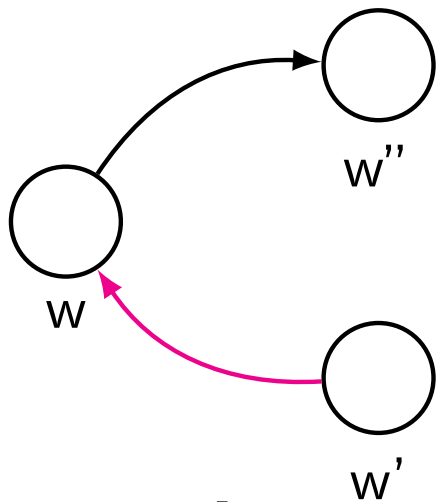
Swap Logic

- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.
- Sea $\mathcal{M} = \langle W, R, V \rangle$ y w, w' elementos de W tal que $R(w, w')$. Denotaremos $\mathcal{M}[w' \mapsto w]$ el modelo idéntico a $\mathcal{M}$, cambiando a R por $(R \setminus (w, w')) \cup \{(w', w)\}$.



Swap Logic

- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.
- Sea $\mathcal{M} = \langle W, R, V \rangle$ y w, w' elementos de W tal que $R(w, w')$. Denotaremos $\mathcal{M}[w' \mapsto w]$ el modelo idéntico a $\mathcal{M}$, cambiando a R por $(R \setminus (w, w')) \cup \{(w', w)\}$.



$\mathcal{M}[w' \mapsto w]$

Swap Logic

- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.
- Sea $\mathcal{M} = \langle W, R, V \rangle$ y w, w' elementos de W tal que $R(w, w')$. Denotaremos $\mathcal{M}[w' \mapsto w]$ el modelo idéntico a $\mathcal{M}$, cambiando a R por $(R \setminus (w, w')) \cup \{(w', w)\}$.
- Extendemos la sintaxis del lenguaje modal básico con dos nuevos operadores ($\langle s \rangle$, $[s]$), con la siguiente semántica:

Swap Logic

- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.
- Sea $\mathcal{M} = \langle W, R, V \rangle$ y w, w' elementos de W tal que $R(w, w')$. Denotaremos $\mathcal{M}[w' \mapsto w]$ el modelo idéntico a $\mathcal{M}$, cambiando a R por $(R \setminus (w, w')) \cup \{(w', w)\}$.
- Extendemos la sintaxis del lenguaje modal básico con dos nuevos operadores ($\langle s \rangle$, $[s]$), con la siguiente semántica:
  - $\mathcal{M}, w \models \langle s \rangle \varphi$ sii existe un sucesor w' t.q. $\mathcal{M}[w' \mapsto w], w' \models \varphi$.

Swap Logic

- Consideremos ahora un lenguaje en el cual podemos modificar las relaciones de un grafo.
- Sea $\mathcal{M} = \langle W, R, V \rangle$ y w, w' elementos de W tal que $R(w, w')$. Denotaremos $\mathcal{M}[w' \mapsto w]$ el modelo idéntico a $\mathcal{M}$, cambiando a R por $(R \setminus (w, w')) \cup \{(w', w)\}$.
- Extendemos la sintaxis del lenguaje modal básico con dos nuevos operadores ($\langle s \rangle$, $[s]$), con la siguiente semántica:
  - $\mathcal{M}, w \models \langle s \rangle \varphi$ sii existe un sucesor w' t.q. $\mathcal{M}[w' \mapsto w], w' \models \varphi$.
  - $\mathcal{M}, w \models [s] \varphi$ sii para todo sucesor w', $\mathcal{M}[w' \mapsto w], w' \models \varphi$.

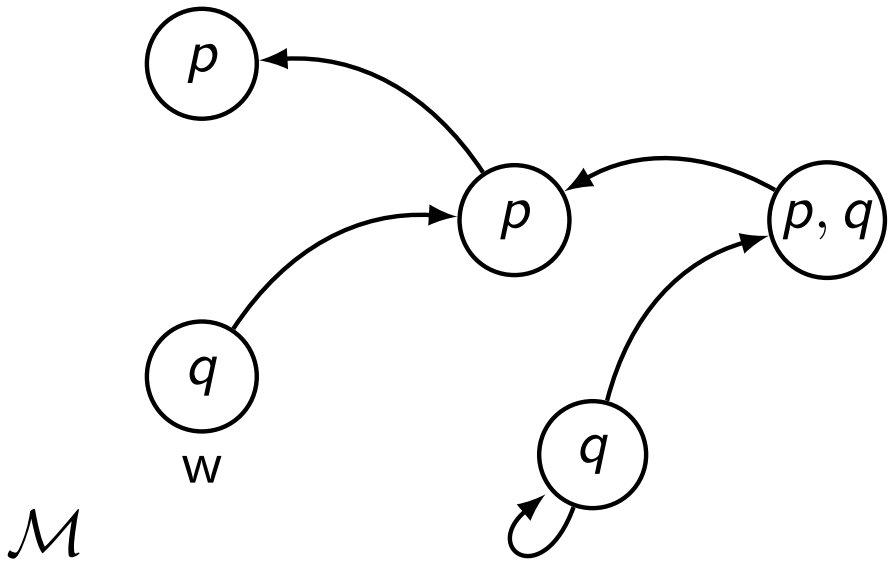
Swap Logic

Swap Logic

Consideremos el siguiente modelo:

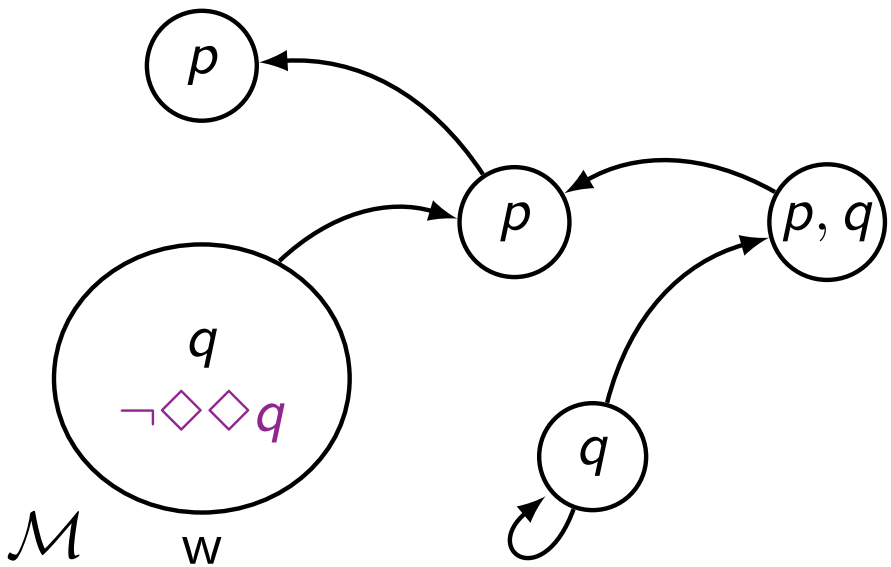
Swap Logic

Consideremos el siguiente modelo:



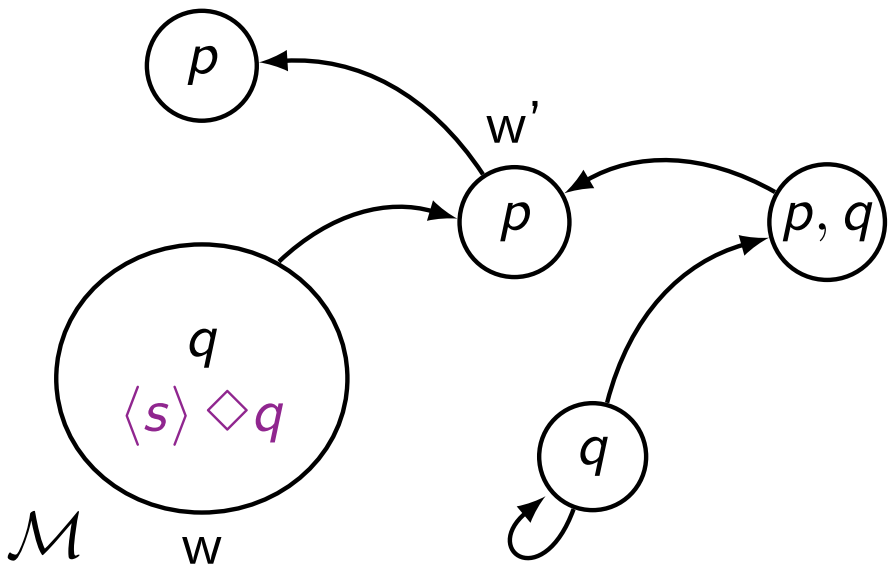
Swap Logic

Consideremos el siguiente modelo:



Swap Logic

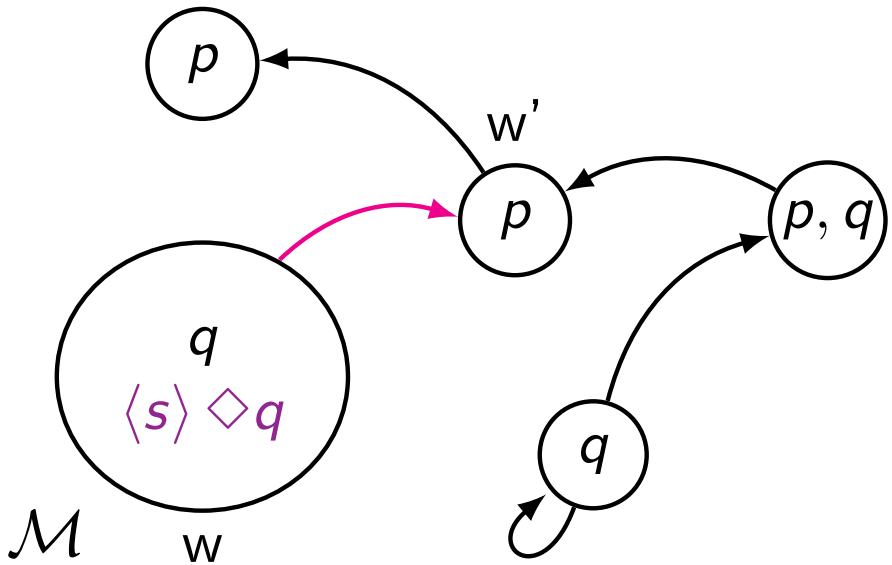
Consideremos el siguiente modelo:



Swap Logic

Consideremos el siguiente modelo:

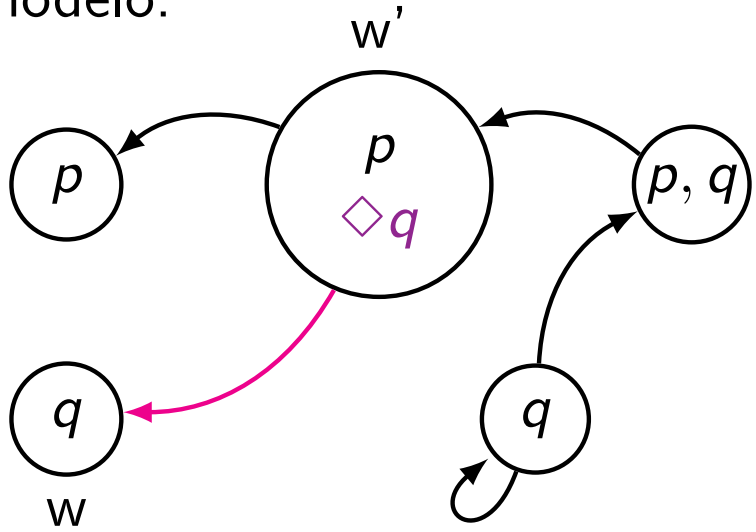
$\mathcal{M}, w \models \langle s \rangle \Diamond q$ iff



Swap Logic

Consideremos el siguiente modelo:

$\mathcal{M}[w' \mapsto w], w' \models \Diamond q$ iff



$\mathcal{M}[w' \mapsto w]$

Swap Logic - Objetivos

Swap Logic - Objetivos

Algunas de las cosas que nos interesan acerca de este lenguaje:

Swap Logic - Objetivos

Algunas de las cosas que nos interesan acerca de este lenguaje:

- Es swap logic equivalente a un fragmento de *FOL*?

Swap Logic - Objetivos

Algunas de las cosas que nos interesan acerca de este lenguaje:

- Es swap logic equivalente a un fragmento de *FOL*?
- Construir una noción de (bi)simulación adecuada, y estudiar la relación entre ésta y equivalencia entre modelos.

Swap Logic - Objetivos

Algunas de las cosas que nos interesan acerca de este lenguaje:

- Es swap logic equivalente a un fragmento de *FOL*?
- Construir una noción de (bi)simulación adecuada, y estudiar la relación entre ésta y equivalencia entre modelos.
- Establecer relaciones entre los modelos de swap logic, y modelos con múltiples modalidades.

Swap Logic - Objetivos

Algunas de las cosas que nos interesan acerca de este lenguaje:

- Es swap logic equivalente a un fragmento de *FOL*?
- Construir una noción de (bi)simulación adecuada, y estudiar la relación entre ésta y equivalencia entre modelos.
- Establecer relaciones entre los modelos de swap logic, y modelos con múltiples modalidades.
- Probar decidibilidad del lenguaje.

Swap Logic - Objetivos

Algunas de las cosas que nos interesan acerca de este lenguaje:

- Es swap logic equivalente a un fragmento de *FOL*?
- Construir una noción de (bi)simulación adecuada, y estudiar la relación entre ésta y equivalencia entre modelos.
- Establecer relaciones entre los modelos de swap logic, y modelos con múltiples modalidades.
- Probar decidibilidad del lenguaje.
- Dar una axiomatización completa para la lógica.

Swap Logic - Traducción a Primer Orden

Swap Logic - Traducción a Primer Orden

- Un resultado conocido dentro de lógica modal, es que toda fórmula modal básica es equivalente a una fórmula de FO^2.

Swap Logic - Traducción a Primer Orden

- Un resultado conocido dentro de lógica modal, es que toda fórmula modal básica es equivalente a una fórmula de FO^2.

Teorema: *Existe una función computable ST que asigna a cada fórmula de swap logic una fórmula equivalente de FOL.*

Swap Logic - Traducción a Primer Orden

- Un resultado conocido dentro de lógica modal, es que toda fórmula modal básica es equivalente a una fórmula de FO^2.

Teorema: *Existe una función computable ST que asigna a cada fórmula de swap logic una fórmula equivalente de FOL.*

- La traducción usa un número no limitado de variables.

Swap Logic - Traducción a Primer Orden

- Un resultado conocido dentro de lógica modal, es que toda fórmula modal básica es equivalente a una fórmula de FO^2.

Teorema: *Existe una función computable ST que asigna a cada fórmula de swap logic una fórmula equivalente de FOL.*

- La traducción usa un número no limitado de variables.
- TODO: Mostrar que esto es necesario o dar una traducción que use solo un número finito.

Swap Logic - Bisimulación

Swap Logic - Bisimulación

- Dentro de lógica modal la manera de definir equivalencia entre modelos es a través de bisimulación.

Swap Logic - Bisimulación

- Dentro de lógica modal la manera de definir equivalencia entre modelos es a través de bisimulación.
- Ya hemos definido la noción adecuada de swap-bisimulación.

Swap Logic - Bisimulación

- Dentro de lógica modal la manera de definir equivalencia entre modelos es a través de bisimulación.
- Ya hemos definido la noción adecuada de swap-bisimulación.

Teorema: *Si dos modelos son swap bisimilares entonces ambos satisfacen las mismas fórmulas de swap logic.*

Swap Logic - Bisimulación

- Dentro de lógica modal la manera de definir equivalencia entre modelos es a través de bisimulación.
- Ya hemos definido la noción adecuada de swap-bisimulación.

Teorema: *Si dos modelos son swap bisimilares entonces ambos satisfacen las mismas fórmulas de swap logic.*

- TODO:
  - Estudiar el cómputo de swap bisimulaciones desde el punto de vista algorítmico.
  - Probar la equivalencia en modelos *"image finite"*.

Swap Logic - Axiomatización

Swap Logic - Axiomatización

- Existe una axiomatización completa para lógica modal básica.

Swap Logic - Axiomatización

- Existe una axiomatización completa para lógica modal básica.

Teorema: *Las siguientes swap fórmulas son válidas en la clase de todos los modelos.*

K $\vdash [s](\varphi \to \psi) \to ([s]\varphi \to [s]\psi)$

S1 $\vdash \Diamond p \leftrightarrow \langle s \rangle p$

S2 $\vdash p \to [s]\Diamond p$

S3 $\vdash \Diamond\varphi \to \langle s \rangle \top$

S4 $\vdash \langle s \rangle \varphi \to \Diamond\top$

Swap Logic - Axiomatización

- Existe una axiomatización completa para lógica modal básica.

Teorema: *Las siguientes swap fórmulas son válidas en la clase de todos los modelos.*

K $\vdash [s](\varphi \to \psi) \to ([s]\varphi \to [s]\psi)$

S1 $\vdash \Diamond p \leftrightarrow \langle s \rangle p$

S2 $\vdash p \to [s]\Diamond p$

S3 $\vdash \Diamond\varphi \to \langle s \rangle \top$

S4 $\vdash \langle s \rangle \varphi \to \Diamond\top$

- TODO: Demostrar completitud con respecto a este conjunto de axiomas, o agregar los necesarios para obtener un sistema completo.