

Lógica Modal Computacional

Carlos Areces

LED, INRIA Lorraine

`carlos.areces@loria.fr`

`http://www.loria.fr/ areces`

12-14, Enero, 2004

Introducción: El Tema Central del Curso

Introducción: El Tema Central del Curso

OK, obviamente durante el curso vamos a hablar todo el tiempo de **Lógicas Modales** (después de todo ese tema cubre las dos terceras partes del título, no?). Pero más allá de aprender que

- las lógicas modales son herramientas para describir y razonar sobre estructuras relacionales.

Introducción: El Tema Central del Curso

OK, obviamente durante el curso vamos a hablar todo el tiempo de **Lógicas Modales** (después de todo ese tema cubre las dos terceras partes del título, no?). Pero más allá de aprender que

- ▶ las lógicas modales son herramientas para describir y razonar sobre estructuras relacionales.

Quiero introducir una cierta forma de usar métodos formales de inferencia.

- ▶ Agregar **sólo lo necesario**
- ▶ Encontrar el **balance** adecuado entre poder expresivo y costo computacional.

Lógicas Modales Hoy

Lógicas Modales Hoy

Clásicamente, las lógicas modales fueron introducidas para capturar nociones “intensionales” como Creencia, Necesidad, Obligación, etc.

Lógicas Modales Hoy

Clásicamente, las lógicas modales fueron introducidas para capturar nociones “intensionales” como Creencia, Necesidad, Obligación, etc.

Aunque aún hoy existe trabajo interesante en esa dirección. Decir que las lógicas modales cubren sólo fenómenos intensionales, es como decir que las computadoras son máquinas de sumar, solo que más rápidas.

Lógicas Modales Hoy

Clásicamente, las lógicas modales fueron introducidas para capturar nociones “intensionales” como Creencia, Necesidad, Obligación, etc.

Aunque aún hoy existe trabajo interesante en esa dirección. Decir que las lógicas modales cubren sólo fenómenos intensionales, es como decir que las computadoras son máquinas de sumar, solo que más rápidas.

En su perspectiva más moderna, las lógicas modales son vistas como

- herramientas para indentificar fragmentos interesantes de lógicas de primer y alto orden,

Lógicas Modales Hoy

Clásicamente, las lógicas modales fueron introducidas para capturar nociones “intensionales” como Creencia, Necesidad, Obligación, etc.

Aunque aún hoy existe trabajo interesante en esa dirección. Decir que las lógicas modales cubren sólo fenómenos intensionales, es como decir que las computadoras son máquinas de sumar, solo que más rápidas.

En su perspectiva más moderna, las lógicas modales son vistas como

- herramientas para indentificar fragmentos **interesantes** de lógicas de primer y alto orden,
- para una determinada **tarea de modelaje o inferencia**.

“Interesante”

“Interesante” usualmente significa:

- ▶ de baja complejidad,
- ▶ buenas propiedades metalógicas (completitud, interpolación, etc.),
- ▶ poder expresivo adecuado,
- ▶ simple diseño de algoritmos de decisión y de herramientas automáticas de inferencia,
- ▶ simplicidad de uso,
- ▶ etc.

“Tareas de Modelaje o Inferencia”

Por el otro lado, las posibles “tareas de modelaje o inferencia” pueden ser extremadamente variadas,

- ▶ aplicaciones clásicas en filosofía y epistemología,
- ▶ usos en lingüística,
- ▶ representación del conocimiento,
- ▶ inteligencia artificial,
- ▶ verificación formal de software y hardware.

El Curso: Requisitos Básicos

El Curso: Requisitos Básicos

- ▶ El curso no requiere conocimientos previos de lógica modal,
- ▶ Sí asume conocimientos básicos sobre
 - lógica proposicional,
 - lógica de predicados,
 - algún conocimiento de complejidad algorítmica.

El Curso: Requisitos Básicos

- ▶ El curso no requiere conocimientos previos de lógica modal,
- ▶ Sí asume conocimientos básicos sobre
 - lógica proposicional,
 - lógica de predicados,
 - algún conocimiento de complejidad algorítmica.
- ▶ Pero de todas formas, si tienen dudas o preguntas, **interrumpen y pregunten.**

Plan del Curso

Plan del Curso

Dia 1: Lógicas clásicas. Introducción a la lógica modal.

Expresividad y Complejidad. Lógicas modales como fragmentos de lógicas clásicas. La traducción standard y otras traducciones.

Testing.

Plan del Curso

Día 1: Lógicas clásicas. Introducción a la lógica modal.

Expresividad y Complejidad. Lógicas modales como fragmentos de lógicas clásicas. La traducción standard y otras traducciones. Testing.

Día 2: El método de tableaux. Tableaux clásicos. Correctitud, completitud y terminación. Lógicas para la descripción. El demostrador automático RACER.

Plan del Curso

- Dia 1:** Lógicas clásicas. Introducción a la lógica modal. Expresividad y Complejidad. Lógicas modales como fragmentos de lógicas clásicas. La traducción standard y otras traducciones. Testing.
- Dia 2:** El método de tableaux. Tableaux clásicos. Correctitud, completitud y terminación. Lógicas para la descripción. El demostrador automático RACER.
- Dia 3:** El método de Resolución. Lógicas híbridas. El demostrador automático HyLoRes.

Qué Hacemos Hoy?

Qué Hacemos Hoy?

- ▶ Ventajas y desventajas de las lógicas clásicas.
 - Decidibilidad.
 - Expresividad.

Qué Hacemos Hoy?

- ▶ Ventajas y desventajas de las lógicas clásicas.
 - Decidibilidad.
 - Expresividad.
- ▶ Introducción a la lógica modal básica.
 - Modelos, satisfiabilidad y validez.
 - Decidibilidad/Indecidibilidad.

Qué Hacemos Hoy?

- ▶ Ventajas y desventajas de las lógicas clásicas.
 - Decidibilidad.
 - Expresividad.
- ▶ Introducción a la lógica modal básica.
 - Modelos, satisfiabilidad y validez.
 - Decidibilidad/Indecidibilidad.
- ▶ Lógicas Modales como Fragmentos de PO
 - La traducción standard
 - Resolución via traducción
 - Testing

Lógica, con L Mayúscula

Lógica, con L Mayúscula

Por muchos años, decir “Lógica” era decir Lógica Clásica, y principalmente Lógica de Primer Orden (LPO).

- ▶ Alto poder expresivo.
- ▶ Simplicidad.
- ▶ Buen comportamiento sintáctico y semántico.
- ▶ Teoría de modelos desarrollada en detalle.

Lógica, con L Mayúscula

Por muchos años, decir “Lógica” era decir Lógica Clásica, y principalmente Lógica de Primer Orden (LPO).

- ▶ Alto poder expresivo.
- ▶ Simplicidad.
- ▶ Buen comportamiento sintáctico y semántico.
- ▶ Teoría de modelos desarrollada en detalle.

Pero, es LPO la única opción?

Es la mejor en todos los casos?

Veamos . . .

Decidibilidad

Decidibilidad

Uno de los principales problemas de LPO (desde el punto de vista computacional) es que decidir si una formula de LPO es satisfacible es **indecidable**.

Decidibilidad

Uno de los principales problemas de LPO (desde el punto de vista computacional) es que decidir si una formula de LPO es satisfacible es **indecidable**.

Cómo se demuestra que un determinado problema X es indecidible?
Una manera:

1. pedirle a alguien más inteligente que nosotros que demuestre que algun problema Y es indecidible.

Decidibilidad

Uno de los principales problemas de LPO (desde el punto de vista computacional) es que decidir si una formula de LPO es satisfacible es **indecidable**.

Cómo se demuestra que un determinado problema X es indecidible?
Una manera:

1. pedirle a alguien más inteligente que nosotros que demuestre que algun problema Y es indecidible.
2. demostrar que si X fuera decidible entonces Y sería decidible, dando una codificación de Y en X .

Decidibilidad

Uno de los principales problemas de LPO (desde el punto de vista computacional) es que decidir si una formula de LPO es satisfacible es **indecidable**.

Cómo se demuestra que un determinado problema X es indecidible?
Una manera:

1. pedirle a alguien más inteligente que nosotros que demuestre que algun problema Y es indecidible.
2. demostrar que si X fuera decidible entonces Y sería decidible, dando una codificación de Y en X .

Ciertos problemas, llamados **Problemas de Tiling** son particularmente simples de codificar como problemas de satisfacción para ciertas lógicas.

Problemas de Tiling

Problemas de Tiling

Un problema de tiling es básicamente un rompecabezas

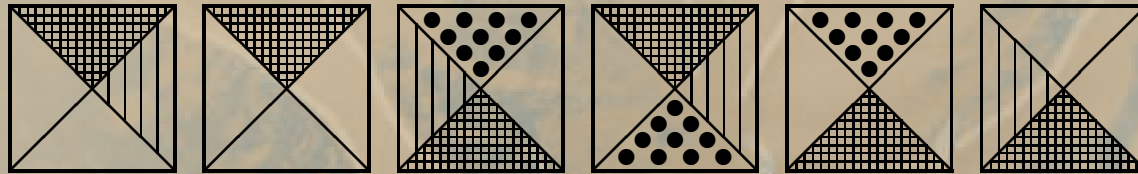
- Un **tile** T es un cuadrado de 1×1 , de orientación fija, con un determinado **color** en cada lado.

Problemas de Tiling

Un problema de tiling es básicamente un rompecabezas

- Un **tile** T es un cuadrado de 1×1 , de orientación fija, con un determinado **color** en cada lado.

Por ejemplo, acá vemos seis tipos diferentes de tiles:

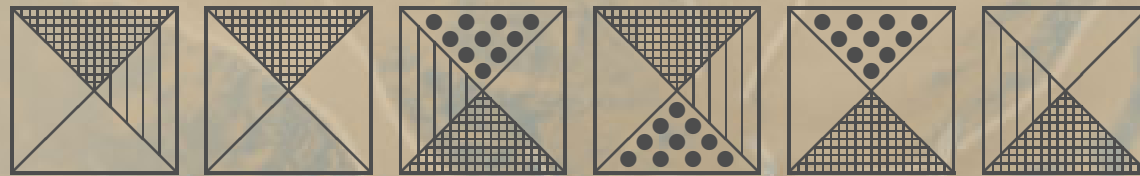


Problemas de Tiling

Un problema de tiling es básicamente un rompecabezas

- Un **tile** T es un cuadrado de 1×1 , de orientación fija, con un determinado **color** en cada lado.

Por ejemplo, acá vemos seis tipos diferentes de tiles:



- Un problema simple de tiling podría ser:

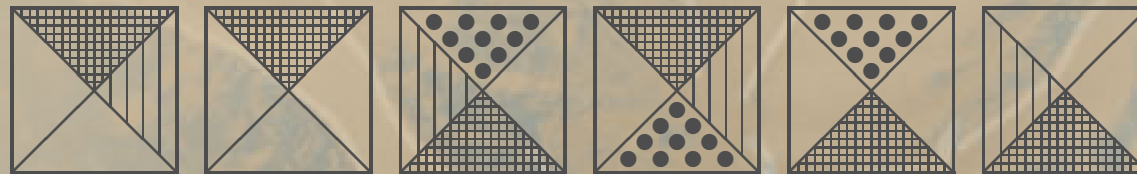
Es posible disponer tiles de los tipos que recién vimos, de forma de cubrir una grilla de 2×4 , y tal que tiles adyacentes tengan el mismo color en el lado compartido?

Problemas de Tiling

Un problema de tiling es básicamente un rompecabezas

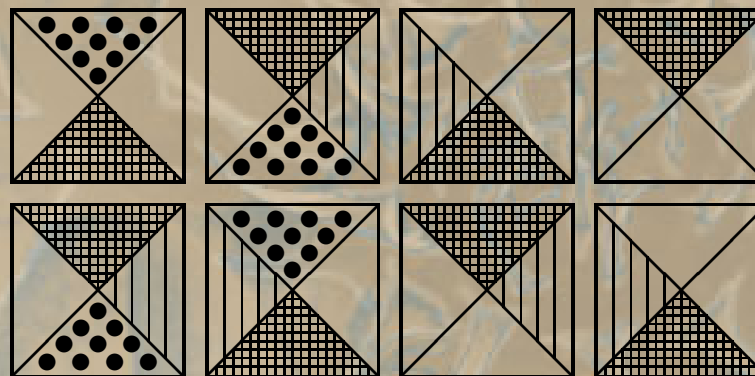
- Un **tile** T es un cuadrado de 1×1 , de orientación fija, con un determinado **color** en cada lado.

Por ejemplo, acá vemos seis tipos diferentes de tiles:



- Un problema simple de tiling podría ser:

Es posible disponer tiles de los tipos que recién vimos, de forma de cubrir una grilla de 2×4 , y tal que tiles adyacentes tengan el mismo color en el lado compartido?



Problemas de Tiling

Problemas de Tiling

- La forma general de un problema de tiling es la siguiente
 - Dado un número finito de tipos de tiles $\mathcal{T}$ representados como un conjunto T finito no vacío y relaciones H y V que indican que tiles puede ser adyacentes. Podemos cubrir una cierta parte de $\mathbb{Z} \times \mathbb{Z}$ de tal forma que tiles adyacentes respeten las relaciones H y V (“tengan el mismo color en el lado compartido”)?

Problemas de Tiling

- ▶ La forma general de un problema de tiling es la siguiente
 - Dado un número finito de tipos de tiles $\mathcal{T}$ representados como un conjunto T finito no vacío y relaciones H y V que indican que tiles puede ser adyacentes. Podemos cubrir una cierta parte de $\mathbb{Z} \times \mathbb{Z}$ de tal forma que tiles adyacentes respeten las relaciones H y V (“tengan el mismo color en el lado compartido”)?
 - En algunos casos, se imponen ciertas condiciones en lo que se considerará un cubrimiento (tiling) correcto.

Problemas de Tiling

- ▶ La forma general de un problema de tiling es la siguiente
 - Dado un número finito de tipos de tiles $\mathcal{T}$ representados como un conjunto T finito no vacío y relaciones H y V que indican que tiles puede ser adyacentes. Podemos cubrir una cierta parte de $\mathbb{Z} \times \mathbb{Z}$ de tal forma que tiles adyacentes respeten las relaciones H y V (“tengan el mismo color en el lado compartido”)?
 - En algunos casos, se imponen ciertas condiciones en lo que se considerará un cubrimiento (tiling) correcto.
- ▶ Cubriendo $\mathbb{N} \times \mathbb{N}$
 - **Tiling de $\mathbb{N} \times \mathbb{N}$** : Dado un conjunto finito de tiles $\mathcal{T}$, podemos $\mathcal{T}$ cubrir $\mathbb{N} \times \mathbb{N}$?
 - Este problema es indecidible (equivalente al problema de la parada de máquinas de Turing). Ver David Harel, **Algorithms. The Essence of Computing** para una explicación excelente.

Codificando el Tiling de $N \times N$ en LPO

Codificando el Tiling de $N \times N$ en LPO

T: El problema de determinar si una fórmula de LPO es satisfacible no es decidable.

D: [Gurevich, 1976] Para $h, v, t_1, \dots, t_n$ símbolos de función unaria. Sea φ la conjunción.

$$h(v(x)) = v(h(x)),$$

Codificando el Tiling de $N \times N$ en LPO

T: El problema de determinar si una fórmula de LPO es satisfacible no es decidable.

D: [Gurevich, 1976] Para $h, v, t_1, \dots, t_n$ símbolos de función unaria. Sea φ la conjunción.

$$\begin{aligned} h(v(x)) &= v(h(x)), \\ \bigvee \{t_i(x) = x \mid 1 \leq i \leq n\}, \end{aligned}$$

Codificando el Tiling de $N \times N$ en LPO

T: El problema de determinar si una fórmula de LPO es satisfacible no es decidable.

D: [Gurevich, 1976] Para $h, v, t_1, \dots, t_n$ símbolos de función unaria. Sea φ la conjunción.

$$\begin{aligned} & h(v(x)) = v(h(x)), \\ & \bigvee \{t_i(x) = x \mid 1 \leq i \leq n\}, \\ & \bigwedge \{t_i(x) = x \rightarrow \neg(t_j(x) = x) \mid 1 \leq i < j \leq n\}, \end{aligned}$$

Codificando el Tiling de $N \times N$ en LPO

T: El problema de determinar si una fórmula de LPO es satisfacible no es decidable.

D: [Gurevich, 1976] Para $h, v, t_1, \dots, t_n$ símbolos de función unaria. Sea φ la conjunción.

$$\begin{aligned} & h(v(x)) = v(h(x)), \\ & \bigvee \{t_i(x) = x \mid 1 \leq i \leq n\}, \\ & \bigwedge \{t_i(x) = x \rightarrow \neg(t_j(x) = x) \mid 1 \leq i < j \leq n\}, \\ & \bigvee \{t_i(x) = x \wedge t_j(h(x)) = h(x) \mid H(t_i, t_j)\}, \end{aligned}$$

Codificando el Tiling de $N \times N$ en LPO

T: El problema de determinar si una fórmula de LPO es satisfacible no es decidable.

D: [Gurevich, 1976] Para $h, v, t_1, \dots, t_n$ símbolos de función unaria. Sea φ la conjunción.

$$\begin{aligned} & h(v(x)) = v(h(x)), \\ & \bigvee \{t_i(x) = x \mid 1 \leq i \leq n\}, \\ & \bigwedge \{t_i(x) = x \rightarrow \neg(t_j(x) = x) \mid 1 \leq i < j \leq n\}, \\ & \bigvee \{t_i(x) = x \wedge t_j(h(x)) = h(x) \mid H(t_i, t_j)\}, \\ & \bigvee \{t_i(x) = x \wedge t_j(v(x)) = v(x) \mid V(t_i, t_j)\}. \end{aligned}$$

Codificando el Tiling de $\mathbb{N} \times \mathbb{N}$ en LPO

T: El problema de determinar si una fórmula de LPO es satisfacible no es decidable.

D: [Gurevich, 1976] Para $h, v, t_1, \dots, t_n$ símbolos de función unaria. Sea φ la conjunción.

$$\begin{aligned} & h(v(x)) = v(h(x)), \\ & \bigvee \{t_i(x) = x \mid 1 \leq i \leq n\}, \\ & \bigwedge \{t_i(x) = x \rightarrow \neg(t_j(x) = x) \mid 1 \leq i < j \leq n\}, \\ & \bigvee \{t_i(x) = x \wedge t_j(h(x)) = h(x) \mid H(t_i, t_j)\}, \\ & \bigvee \{t_i(x) = x \wedge t_j(v(x)) = v(x) \mid V(t_i, t_j)\}. \end{aligned}$$

Entonces $\forall x. \varphi(x)$ es satisfacible sii $\mathcal{T}$ cubre $\mathbb{N} \times \mathbb{N}$.

Poder Expresivo

Poder Expresivo

El resultado de indecidibilidad anterior puede interpretarse como diciendo que LPO es **demasiado expresiva**: permite codificar problemas indecidibles.

Poder Expresivo

El resultado de indecidibilidad anterior puede interpretarse como diciendo que LPO es **demasiado expresiva**: permite codificar problemas indecidibles.

Pero a la vez, LPO puede no tener el **tipo de expresividad** necesaria para una tarea determinada.

- ▶ Dominio finito,
- ▶ Dominio conexo,
- ▶ clausura transitiva,
- ▶ etc.

Poder Expresivo

El resultado de indecidibilidad anterior puede interpretarse como diciendo que LPO es **demasiado expresiva**: permite codificar problemas indecidibles.

Pero a la vez, LPO puede no tener el **tipo de expresividad** necesaria para una tarea determinada.

- ▶ Dominio finito,
- ▶ Dominio conexo,
- ▶ clausura transitiva,
- ▶ etc.

De todas formas, LPO **es** una lógica extremadamente útil y de buen comportamiento.

La Lógica Modal Básica

La Lógica Modal Básica

► El Language:

- empezamos con un conjunto de símbolos proposicionales PROP
- $\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid \Box\varphi$
- 'diamond φ ' y 'box φ ' ?!!

La Lógica Modal Básica

► El Language:

- empezamos con un conjunto de símbolos proposicionales PROP
- $\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid \Box\varphi$
- 'diamond φ ' y 'box φ ' ?!!

► Modelos:

- Una **estructura relacional** es un $\mathcal{M} = \langle W, \{R_a \mid a \in A\} \rangle$ donde
 - $W \neq \emptyset$ es el universo o dominio.
 - $\{R_a \mid a \in A\}$ es un conjunto de relaciones sobre W .

La Lógica Modal Básica

► El Language:

- empezamos con un conjunto de símbolos proposicionales PROP
- $\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid \Box\varphi$
- 'diamond φ ' y 'box φ ' ?!!

► Modelos:

- Una **estructura relacional** es un $\mathcal{M} = \langle W, \{R_a \mid a \in A\} \rangle$ donde
 - $W \neq \emptyset$ es el universo o dominio.
 - $\{R_a \mid a \in A\}$ es un conjunto de relaciones sobre W .
- un **modelo de Kripke** es una tripla $\mathcal{M} = \langle W, R, V \rangle$, donde $\langle W, R \rangle$ es una estructura relacional y V es una **valuación** que asigna subconjuntos de W a los elementos de φ (un grafo decorado)

La Lógica Modal Básica

► El Language:

- empezamos con un conjunto de símbolos proposicionales PROP
- $\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid \Box\varphi$
- 'diamond φ ' y 'box φ ' ?!!

► Modelos:

- Una **estructura relacional** es un $\mathcal{M} = \langle W, \{R_a \mid a \in A\} \rangle$ donde
 - $W \neq \emptyset$ es el universo o dominio.
 - $\{R_a \mid a \in A\}$ es un conjunto de relaciones sobre W .
- un **modelo de Kripke** es una tripla $\mathcal{M} = \langle W, R, V \rangle$, donde $\langle W, R \rangle$ es una estructura relacional y V es una **valuación** que asigna subconjuntos de W a los elementos de φ (un grafo decorado)
- el par $\langle W, R \rangle$ se llama usualmente un **frame**.

Relación de Satisfiabilidad

Relación de Satisfiabilidad

- Interpretando formulas:

$$M, w \models p \quad \text{sii} \quad w \in V(p)$$

Relación de Satisfiabilidad

- Interpretando formulas:

$$M, w \models p \quad \text{sii} \quad w \in V(p)$$

$$M, w \models \neg\varphi \quad \text{sii} \quad M, w \not\models \varphi$$

Relación de Satisfiabilidad

► Interpretando formulas:

$$M, w \models p \quad \text{sii} \quad w \in V(p)$$

$$M, w \models \neg\varphi \quad \text{sii} \quad M, w \not\models \varphi$$

$$M, w \models \varphi \wedge \psi \quad \text{sii} \quad M, w \models \varphi \text{ y } M, w \models \psi$$

Relación de Satisfiabilidad

► Interpretando formulas:

$$M, w \models p \quad \text{sii} \quad w \in V(p)$$

$$M, w \models \neg\varphi \quad \text{sii} \quad M, w \not\models \varphi$$

$$M, w \models \varphi \wedge \psi \quad \text{sii} \quad M, w \models \varphi \text{ y } M, w \models \psi$$

$$M, w \models \Diamond\varphi \quad \text{sii} \quad \text{para algún } v \text{ tal que } R w v, M, v \models \varphi$$

Relación de Satisfiabilidad

► Interpretando formulas:

$$M, w \models p \quad \text{sii} \quad w \in V(p)$$

$$M, w \models \neg\varphi \quad \text{sii} \quad M, w \not\models \varphi$$

$$M, w \models \varphi \wedge \psi \quad \text{sii} \quad M, w \models \varphi \text{ y } M, w \models \psi$$

$$M, w \models \Diamond\varphi \quad \text{sii} \quad \text{para algún } v \text{ tal que } R w v, M, v \models \varphi$$

$$M, w \models \Box\varphi \quad \text{sii} \quad \text{para todo } v \text{ tal que } R w v, M, v \models \varphi$$

Qué es lo que Hace a una Lógica Modal?

Qué es lo que Hace a una Lógica Modal?

- Semántica en términos de modelos relacionales.

Qué es lo que Hace a una Lógica Modal?

- ▶ Semántica en términos de modelos relacionales.
- ▶ Una perspectiva interna
 - las formulas se evalúan respecto de un punto/estado/momento/. . . dado en el grafo.

Qué es lo que Hace a una Lógica Modal?

- ▶ Semántica en términos de modelos relacionales.
- ▶ Una perspectiva interna
 - las formulas se evalúan respecto de un punto/estado/momento/. . . dado en el grafo.
- ▶ Quantificación restringida
 - sólo ciertos patrones de cuantificación están permitidos.
 - Podemos investigar sólo puntos “relacionados”.

Tareas de Inferencia: Satisfiabilidad y Validez

Tareas de Inferencia: Satisfiabilidad y Validez

- Una fórmula modal φ es **satisfacible en un modelo** M si existe w en M tal que $M, w \models \varphi$

Tareas de Inferencia: Satisfiabilidad y Validez

- ▶ Una fórmula modal φ es **satisfacible en un modelo** M si existe w en M tal que $M, w \models \varphi$
- ▶ Una fórmula es **satisfacible** si es satisfacible en algún modelo M

Tareas de Inferencia: Satisfiabilidad y Validez

- ▶ Una fórmula modal φ es **satisfacible en un modelo** M si existe w en M tal que $M, w \models \varphi$
- ▶ Una fórmula es **satisfacible** si es satisfacible en algún modelo M
- ▶ φ es **válida en un modelo** M , notación: $M \models \varphi$, si $M, w \models \varphi$ para todo w in M

Tareas de Inferencia: Satisfiabilidad y Validez

- ▶ Una fórmula modal φ es **satisfacible en un modelo** M si existe w en M tal que $M, w \models \varphi$
- ▶ Una fórmula es **satisfacible** si es satisfacible en algún modelo M
- ▶ φ es **válida en un modelo** M , notación: $M \models \varphi$, si $M, w \models \varphi$ para todo w in M
- ▶ φ es **válida** si es válida en todo modelo

Tareas de Inferencia

Tareas de Inferencia

Dado un language $\mathcal{L}$ y una determinada clase de modelos $\mathcal{K}$.

► **comparación de modelos:**

- Input: dos modelos $\mathcal{M}_1$ y $\mathcal{M}_2$ en $\mathcal{K}$;
- Pregunta: son $\mathcal{M}_1$ y $\mathcal{M}_2$ “el mismo” respecto de $\mathcal{L}$?
- Output: Si/No.

Tareas de Inferencia

Dado un language $\mathcal{L}$ y una determinada clase de modelos $\mathcal{K}$.

► **comparación de modelos:**

- Input: dos modelos $\mathcal{M}_1$ y $\mathcal{M}_2$ en $\mathcal{K}$;
- Pregunta: son $\mathcal{M}_1$ y $\mathcal{M}_2$ “el mismo” respecto de $\mathcal{L}$?
- Output: Si/No.

► **model checking:**

- Input: un modelo $\mathcal{M}$ en $\mathcal{K}$ y una formula φ en $\mathcal{L}$;
- Pregunta: satisface $\mathcal{M}$ la formula φ ?
- Output: Si/No.

Tareas de Inferencia

Dado un language $\mathcal{L}$ y una determinada clase de modelos $\mathcal{K}$.

- ▶ **comparación de modelos:**
 - Input: dos modelos $\mathcal{M}_1$ y $\mathcal{M}_2$ en $\mathcal{K}$;
 - Pregunta: son $\mathcal{M}_1$ y $\mathcal{M}_2$ “el mismo” respecto de $\mathcal{L}$?
 - Output: Si/No.
- ▶ **model checking:**
 - Input: un modelo $\mathcal{M}$ en $\mathcal{K}$ y una formula φ en $\mathcal{L}$;
 - Pregunta: satisface $\mathcal{M}$ la formula φ ?
 - Output: Si/No.
- ▶ **satisfiabilidad:**
 - Input: una formula φ en $\mathcal{L}$
 - Pregunta: existe un modelo en $\mathcal{K}$ que satisfaga φ ?
 - Output: Yes/No.

Tareas de Inferencia

Dado un language $\mathcal{L}$ y una determinada clase de modelos $\mathcal{K}$.

► **comparación de modelos:**

- Input: dos modelos $\mathcal{M}_1$ y $\mathcal{M}_2$ en $\mathcal{K}$;
- Pregunta: son $\mathcal{M}_1$ y $\mathcal{M}_2$ “el mismo” respecto de $\mathcal{L}$?
- Output: Si/No.

► **model checking:**

- Input: un modelo $\mathcal{M}$ en $\mathcal{K}$ y una formula φ en $\mathcal{L}$;
- Pregunta: satisface $\mathcal{M}$ la formula φ ?
- Output: Si/No.

► **satisfiabilidad:**

- Input: una formula φ en $\mathcal{L}$
- Pregunta: existe un modelo en $\mathcal{K}$ que satisfaga φ ?
- Output: Yes/No.

► **generación de modelos:**

- Input: una formula φ en $\mathcal{L}$
- Pregunta: existe un modelo $\mathcal{M}$ en $\mathcal{K}$ que satisfaga φ ?
- Output: una representación de $\mathcal{M}$.

Validez

Validez

- “Demostrabilidad” en términos axiomáticos. La lógica **K**

A1 Todas las tautologías de lógica proposicional.

A2 $\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$

R1 De φ y $\varphi \rightarrow \psi$, inferir ψ

R2 De φ inferir $\Box\varphi$

- **K** (Kripke) es un sistema axiomático correcto y completo.

Modelos Finitos y Decidibilidad

Modelos Finitos y Decidibilidad

- Podemos decidir efectivamente si una formula de la lógica modal básica es satisfacible o no?

Modelos Finitos y Decidibilidad

- ▶ Podemos decidir efectivamente si una formula de la lógica modal básica es satisfacible o no?
- ▶ La **propiedad de modelo finito** dice que φ es válida en todo modelo sii es válida en todos los modelos finitos. El hecho de que la lógica básica (**K**) tenga la propiedad de modelo finito implica decidibilidad. Por qué?

Modelos Finitos y Decidibilidad

- ▶ Podemos decidir efectivamente si una formula de la lógica modal básica es satisfacible o no?
- ▶ La **propiedad de modelo finito** dice que φ es válida en todo modelo sii es válida en todos los modelos finitos. El hecho de que la lógica básica (**K**) tenga la propiedad de modelo finito implica decidibilidad. Por qué?
- ▶ Decidibilidad “sin completitud”: si una formula es satisfacible, es satisfacible en una estructura finita de **tamaño limitado** (bounded model property)

Modelos Finitos y Decidibilidad

- ▶ Podemos decidir efectivamente si una formula de la lógica modal básica es satisfacible o no?
- ▶ La **propiedad de modelo finito** dice que φ es válida en todo modelo sii es válida en todos los modelos finitos. El hecho de que la lógica básica (**K**) tenga la propiedad de modelo finito implica decidibilidad. Por qué?
- ▶ Decidibilidad “sin completitud”: si una formula es satisfacible, es satisfacible en una estructura finita de **tamaño limitado** (bounded model property)
- ▶ En particular, la logica modal básica posee la bounded model property: si φ es satisfacible, entonces es satisfacible en un modelo con a lo sumo $2^{|\varphi|}$ elementos.

Decidibilidad y Complejidad

Decidibilidad y Complejidad

Nos interesa determinar aspectos computacionales del **problema de satisfiabilidad**: dados una fórmula modal φ y una clase de modelos M , es el problema de determinar si φ es satisfacible en un modelo de M computable?

- Si la respuesta es **SI**, podemos seguir preguntando: que tan complejo es este problema?

Decidibilidad y Complejidad

Nos interesa determinar aspectos computacionales del **problema de satisfiabilidad**: dados una fórmula modal φ y una clase de modelos M , es el problema de determinar si φ es satisfacible en un modelo de M computable?

- ▶ Si la respuesta es **SI**, podemos seguir preguntando: que tan complejo es este problema?
- ▶ Si la respuesta es **NO**, en que nivel de no computabilidad se ubica?

Métodos para Demostrar In/Decidibilidad

Métodos para Demostrar In/Decidibilidad

- ▶ Decidibilidad
 - Via modelos finitos
 - Via interpretacion en la logica modadica de segundo orden sobre arboles
 - Quasi-modelos y Mosaicos
- ▶ Indecidibilidad
 - Via problemas de tiling
 - Via codificacion de una lógica indecidible

Lógicas Modales como Fragmentos de LPO

Lógicas Modales como Fragmentos de LPO

- Los modelos $\langle W, R, V \rangle$ pueden ser vistos como modelos para un lenguaje de primer orden.

Lógicas Modales como Fragmentos de LPO

- ▶ Los modelos $\langle W, R, V \rangle$ pueden ser vistos como modelos para un lenguaje de primer orden.
- ▶ Como?
 - Consideremos un lenguaje de primer orden con
 - un símbolo de relación binario R (la relación de accesibilidad)
 - símbolos de relación unarios $P_1, P_2, \dots$ (para los símbolos de proposición).

Lógicas Modales como Fragmentos de LPO

- ▶ Los modelos $\langle W, R, V \rangle$ pueden ser vistos como modelos para un lenguaje de primer orden.
- ▶ Como?
 - Consideremos un lenguaje de primer orden con
 - un simbolo de relación binario R (la relacion de accesibilidad)
 - simbolos de relación unarios $P_1, P_2, \dots$ (para los símbolos de proposición).
- ▶ Entonces
 - Usamos R (en el modelo) para interpretar R
 - Usamos $V(p_i)$ para interpretar las simbolos de predicado P_i

Lógicas Modales como Fragmentos de LPO

Lógicas Modales como Fragmentos de LPO

- Semánticamente entonces la relación es fuerte. El vínculo sintáctico lo da la traducción **standard** o **relational**:

$$ST_x(p_i) = P_i(x)$$

Lógicas Modales como Fragmentos de LPO

- Semánticamente entonces la relación es fuerte. El vínculo sintáctico lo da la traducción **standard** o **relational**:

$$\begin{aligned} ST_x(p_i) &= P_i(x) \\ ST_x(\neg\varphi) &= \neg ST_x(\varphi) \end{aligned}$$

Lógicas Modales como Fragmentos de LPO

- Semánticamente entonces la relación es fuerte. El vínculo sintáctico lo da la traducción **standard** o **relational**:

$$ST_x(p_i) = P_i(x)$$

$$ST_x(\neg\varphi) = \neg ST_x(\varphi)$$

$$ST_x(\varphi \wedge \psi) = ST_x(\varphi) \wedge ST_x(\psi)$$

Lógicas Modales como Fragmentos de LPO

- Semánticamente entonces la relación es fuerte. El vínculo sintáctico lo da la traducción **standard** o **relational**:

$$ST_x(p_i) = P_i(x)$$

$$ST_x(\neg\varphi) = \neg ST_x(\varphi)$$

$$ST_x(\varphi \wedge \psi) = ST_x(\varphi) \wedge ST_x(\psi)$$

$$ST_x(\Diamond\varphi) = \exists y (Rxy \wedge ST_y(\varphi))$$

Lógicas Modales como Fragmentos de LPO

- Semánticamente entonces la relación es fuerte. El vínculo sintáctico lo da la traducción **standard** o **relational**:

$$ST_x(p_i) = P_i(x)$$

$$ST_x(\neg\varphi) = \neg ST_x(\varphi)$$

$$ST_x(\varphi \wedge \psi) = ST_x(\varphi) \wedge ST_x(\psi)$$

$$ST_x(\Diamond\varphi) = \exists y (Rxy \wedge ST_y(\varphi))$$

$$ST_x(\Box\varphi) = \forall y (Rxy \rightarrow ST_y(\varphi))$$

Lógicas Modales como Fragmentos de LPO

- Semánticamente entonces la relación es fuerte. El vínculo sintáctico lo da la traducción **standard** o **relational**:

$$ST_x(p_i) = P_i(x)$$

$$ST_x(\neg\varphi) = \neg ST_x(\varphi)$$

$$ST_x(\varphi \wedge \psi) = ST_x(\varphi) \wedge ST_x(\psi)$$

$$ST_x(\Diamond\varphi) = \exists y (Rxy \wedge ST_y(\varphi))$$

$$ST_x(\Box\varphi) = \forall y (Rxy \rightarrow ST_y(\varphi))$$

donde y es una variable no usada hasta el momento.

Lógicas Modales como Fragmentos de LPO

Lógicas Modales como Fragmentos de LPO

Teorema: Para toda formula modal φ , todo modelo M y todo estado w en M ,

$$M, w \models \varphi \text{ sii } M \models ST_x(\varphi)[w]$$

Lógicas Modales como Fragmentos de LPO

Teorema: Para toda formula modal φ , todo modelo M y todo estado w en M ,

$$M, w \models \varphi \text{ sii } M \models ST_x(\varphi)[w]$$

$$M \models \varphi \text{ sii } M \models \forall x.ST_x(\varphi).$$

Lógicas Modales como Fragmentos de LPO

Teorema: Para toda formula modal φ , todo modelo M y todo estado w en M ,

$$M, w \models \varphi \text{ sii } M \models ST_x(\varphi)[w]$$

$$M \models \varphi \text{ sii } M \models \forall x.ST_x(\varphi).$$

En otras palabras, cuando las interpretamos sobre modelos, las formulas modales terminan siendo un **fragmento** de la logica de primer orden!

- Por que los fragmentos modales son **especiales**?

Fragmentos de LPO y Extensiones

Fragmentos de LPO y Extensiones

- ▶ Fragmentos en forma normal prenexa (Prenex Normal Form Fragments).
- ▶ Fragmentos con numero finito de variables (Finite Variable Fragments).
- ▶ Fragmentos de Guarda (Guarded Fragments)

Prenex Normal Form Fragments

Prenex Normal Form Fragments

- Definidos mediante strings sobre $\{\exists, \forall, \exists^*, \forall^*\}$

Prenex Normal Form Fragments

- Definidos mediante strings sobre $\{\exists, \forall, \exists^*, \forall^*\}$
- Area de investigacion muy activa desde el comienzo del siglo XX.

Prenex Normal Form Fragments

- Definidos mediante strings sobre $\{\exists, \forall, \exists^*, \forall^*\}$
- Area de investigacion muy activa desde el comienzo del siglo XX.
- La decidibilidad/indecidibilidad de todos los fragmentos ha sido totalmente mapeada (Teorema de Clasificación. [Gurevich, 1969]). En muchos casos tambien se conoce la complejidad.

Prenex Normal Form Fragments

- Definidos mediante **strings** sobre $\{\exists, \forall, \exists^*, \forall^*\}$
- Area de investigacion muy activa desde el comienzo del siglo XX.
- La decidibilidad/indecidibilidad de **todos** los fragmentos ha sido totalmente mapeada (Teorema de Clasificación. [Gurevich, 1969]). En muchos casos tambien se conoce la complejidad.
 - Decidibles: $\exists^*\forall\exists^*$, $\exists^*\forall^2\exists^*$, ...
 - Indecidibles: $\forall^*\exists^*$, $\forall^3\exists^*$, $\forall\exists\forall$, ...

Prenex Normal Form Fragments

- Definidos mediante strings sobre $\{\exists, \forall, \exists^*, \forall^*\}$
- Area de investigacion muy activa desde el comienzo del siglo XX.
- La decidibilidad/indecidibilidad de todos los fragmentos ha sido totalmente mapeada (Teorema de Clasificación. [Gurevich, 1969]). En muchos casos tambien se conoce la complejidad.
 - Decidibles: $\exists^*\forall\exists^*$, $\exists^*\forall^2\exists^*$, ...
 - Indecidibles: $\forall^*\exists^*$, $\forall^3\exists^*$, $\forall\exists\forall$, ...
- Classical Decision Problem de Börger, Grädel y Gurevich.

Prenex Normal Form Fragments

- Definidos mediante **strings sobre** $\{\exists, \forall, \exists^*, \forall^*\}$
- Area de investigacion muy activa desde el comienzo del siglo XX.
- La decidibilidad/indecidibilidad de **todos** los fragmentos ha sido totalmente mapeada (Teorema de Clasificación. [Gurevich, 1969]). En muchos casos tambien se conoce la complejidad.
 - Decidibles: $\exists^*\forall\exists^*$, $\exists^*\forall^2\exists^*$, ...
 - Indecidibles: $\forall^*\exists^*$, $\forall^3\exists^*$, $\forall\exists\forall$, ...
- **Classical Decision Problem** de Börger, Grädel y Gurevich.

Puede el fragmento modal mapearse (directamente) en alguno de estos fragmentos?

Finite Variable Fragments

Finite Variable Fragments

- Permiten un número finito (fijo) de variables, con reuso.

Finite Variable Fragments

- ▶ Permiten un **número finito (fijo) de variables**, con reuso.
- ▶ Ejemplo: Se necesitan $n + 1$ variables para caracterizar un camino de longitud n en un grafo G , si pedimos que la formula este en forma normal prenexa

$$\exists x_0 \dots x_n \left(\bigwedge_{0 \leq i < n} E(x_i, x_{i+1}) \right),$$

pero una formula en LPO^2 es suficiente si permitimos reuso:

$$\exists x_0 x_1 (E(x_0, x_1) \wedge \exists x_0 (E(x_1, x_0) \wedge \dots)).$$

Finite Variable Fragments

- ▶ Permiten un **número finito (fijo) de variables**, con reuso.
- ▶ Ejemplo: Se necesitan $n + 1$ variables para caracterizar un camino de longitud n en un grafo G , si pedimos que la formula este en forma normal prenexa

$$\exists x_0 \dots x_n \left(\bigwedge_{0 \leq i < n} E(x_i, x_{i+1}) \right),$$

pero una formula en LPO^2 es suficiente si permitimos reuso:

$$\exists x_0 x_1 (E(x_0, x_1) \wedge \exists x_0 (E(x_1, x_0) \wedge \dots)).$$

- ▶ la traduccion relacional puede ser modificada para mapear formulas modales en LPO^2

Finite Variable Fragments

Finite Variable Fragments

- ▶ LPO^2 Decidable [Scott, 1962; Mortimer, 1975]
- ▶ LPO^k , $k \geq 3$ Indecidable.
- ▶ LPO^k , $k \geq 2$ Falla Interpolación.
- ▶ LPO^k , Caracterizables vía resultados de teoría de modelos (k -back-and-forth systems).

Finite Variable Fragments

- ▶ LPO^2 Decidable [Scott, 1962; Mortimer, 1975]
- ▶ LPO^k , $k \geq 3$ Indecidable.
- ▶ LPO^k , $k \geq 2$ Falla Interpolación.
- ▶ LPO^k , Caracterizables vía resultados de teoría de modelos (k -back-and-forth systems).
- ▶ LPO^2
 - Model checking: $O(||M||^2 \cdot |\varphi|)$

Finite Variable Fragments

- ▶ LPO^2 Decidable [Scott, 1962; Mortimer, 1975]
- ▶ LPO^k , $k \geq 3$ Indecidable.
- ▶ LPO^k , $k \geq 2$ Falla Interpolación.
- ▶ LPO^k , Caracterizables vía resultados de teoría de modelos (k -back-and-forth systems).
- ▶ LPO^2
 - Model checking: $O(||M||^2 \cdot |\varphi|)$
 - Validez: LPO^2 tiene la strong finite model property y por lo tanto es decidable. Mas específicamente, LPO^2 es NEXPTIME-complete

Finite Variable Fragments

- ▶ LPO^2 Decidable [Scott, 1962; Mortimer, 1975]
- ▶ LPO^k , $k \geq 3$ Indecidable.
- ▶ LPO^k , $k \geq 2$ Falla Interpolación.
- ▶ LPO^k , Caracterizables vía resultados de teoría de modelos (k -back-and-forth systems).
- ▶ LPO^2
 - Model checking: $O(||M||^2 \cdot |\varphi|)$
 - Validez: LPO^2 tiene la strong finite model property y por lo tanto es decidable. Mas específicamente, LPO^2 es NEXPTIME-complete
 - El resultado puede ser extendido para incluir constantes.

Finite Variable Fragments

- ▶ LPO^2 Decidable [Scott, 1962; Mortimer, 1975]
- ▶ LPO^k , $k \geq 3$ Indecidable.
- ▶ LPO^k , $k \geq 2$ Falla Interpolación.
- ▶ LPO^k , Caracterizables vía resultados de teoría de modelos (k -back-and-forth systems).
- ▶ LPO^2
 - Model checking: $O(||M||^2 \cdot |\varphi|)$
 - Validez: LPO^2 tiene la strong finite model property y por lo tanto es decidable. Mas específicamente, LPO^2 es NEXPTIME-complete
 - El resultado puede ser extendido para incluir constantes.
 - Podemos incluir tambien simbolos de funcion?

Guarded Fragments

Guarded Fragments

- El fragmento modal solo permite una forma muy restringida de cuantificación

$$\forall y (Rxy \wedge ST_y(\varphi))$$

Guarded Fragments

- El fragmento modal solo permite una forma muy restringida de cuantificación

$$\forall y (Rxy \wedge ST_y(\varphi))$$

- Que pasa si liberamos la restricción sobre el numero de variables y restringimos a cambio los patrones de cuantificación?

Guarded Fragments

- El fragmento modal solo permite una forma muy restringida de cuantificación

$$\forall y (Rxy \wedge ST_y(\varphi))$$

- Que pasa si liberamos la restricción sobre el numero de variables y restringimos a cambio los patrones de cuantificación?
- Esto nos lleva al **guarded fragment** de LPO en donde los cuantificadores pueden aparecer sólo como

$$\exists \bar{y} (G(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y}))$$

en donde $G(\bar{x}, \bar{y})$ es una formula atómica llamada la **guarda**.

Toda variable libre de ψ debe tambien aparecer libre en la guarda.

Métodos de Inferencia para Satisfiabilidad y Validez

Métodos de Inferencia para Satisfiabilidad y Validez

- ▶ **Métodos directos:** trabajan directamente sobre las formulas modales
 - axiomáticos, tableau, resolución

Métodos de Inferencia para Satisfiabilidad y Validez

- ▶ **Métodos directos:** trabajan directamente sobre las formulas modales
 - axiomáticos, tableau, resolución
- ▶ **Métodos indirectos:** traducen el input modal en otro formalismo, y resuelven sat modal mediante algun método de inferencia en dicho formaliso
 - lógica de primer orden, MSO sobre arboles, autómata

Métodos de Inferencia para Satisfiabilidad y Validez

- ▶ **Métodos directos:** trabajan directamente sobre las formulas modales
 - axiomáticos, tableau, resolución
- ▶ **Métodos indirectos:** traducen el input modal en otro formalismo, y resuelven sat modal mediante algún método de inferencia en dicho formalismo
 - lógica de primer orden, MSO sobre arboles, autómatas
- ▶ Ventajas/desventajas . . .

Resolución para Lógica Proposicional

Resolución para Lógica Proposicional

► **Forma clausal:**

- Escribir φ en forma normal conjuntiva: $\varphi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)}$,

Resolución para Lógica Proposicional

► Forma clausal:

- Escribir φ en forma normal conjuntiva: $\varphi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)}$,
- y sea el clause set asociado a φ $ClSet(\varphi) = \{\{\psi_{(l,m)} \mid m \in M\} \mid l \in L\}$.

Resolución para Lógica Proposicional

► Forma clausal:

- Escribir φ en forma normal conjuntiva: $\varphi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi(l, m)$,
- y sea el clause set asociado a φ $ClSet(\varphi) = \{\{\psi(l, m) \mid m \in M\} \mid l \in L\}$.

- ## ► Sea $ClSet^*(\varphi)$ be el conjunto mas pequeño que contiene a $ClSet(\varphi)$ y esta clausurado bajo la regla (RES):

$$\frac{Cl_1 \cup \{N\} \in ClSet^*(\varphi) \quad Cl_2 \cup \{\neg N\} \in ClSet^*(\varphi)}{Cl_1 \cup Cl_2 \in ClSet^*(\varphi)}$$

Resolución para Lógica Proposicional

► Forma clausal:

- Escribir φ en forma normal conjuntiva: $\varphi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi(l, m)$,
- y sea el clause set asociado a φ $ClSet(\varphi) = \{\{\psi(l, m) \mid m \in M\} \mid l \in L\}$.

- ## ► Sea $ClSet^*(\varphi)$ be el conjunto mas pequeño que contiene a $ClSet(\varphi)$ y esta clausurado bajo la regla (RES):

$$\frac{Cl_1 \cup \{N\} \in ClSet^*(\varphi) \quad Cl_2 \cup \{\neg N\} \in ClSet^*(\varphi)}{Cl_1 \cup Cl_2 \in ClSet^*(\varphi)}$$

- ## ► Teorema: $\forall \varphi$, $ClSet^*(\varphi)$ es finito. φ es inconsistente sii $\{\} \in ClSet^*(\varphi)$.

Resolución para Lógica de Primer Orden

Resolución para Lógica de Primer Orden

- **Forma Clausal.** Escribir φ en forma prenexa, con la matriz en normal conjuntiva

$$\varphi = Q.\psi$$

$$\psi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)}.$$

Resolución para Lógica de Primer Orden

- **Forma Clausal**. Escribir φ en forma prenexa, con la matriz en normal conjuntiva

$$\begin{aligned}\varphi &= \mathcal{Q}.\psi \\ \psi &= \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)}.\end{aligned}$$

Sea $Sko(\varphi)$ la **skolemización** de φ :

- Mientras exista un cuantificador existencial en $\mathcal{Q}$, sea $\bar{x}$ la lista de variables universalmente cuantificadas en $\mathcal{Q}$ que ocurren delante del primer cuantificador existencial $\exists x_i$.

Resolución para Lógica de Primer Orden

- **Forma Clausal.** Escribir φ en forma prenexa, con la matriz en normal conjuntiva

$$\begin{aligned}\varphi &= \mathcal{Q}.\psi \\ \psi &= \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)}.\end{aligned}$$

Sea $Sko(\varphi)$ la **skolemización** de φ :

- Mientras exista un cuantificador existencial en $\mathcal{Q}$, sea $\bar{x}$ la lista de variables universalmente cuantificadas en $\mathcal{Q}$ que ocurren delante del primer cuantificador existencial $\exists x_i$.
- Eliminar $\exists x_i$ de $\mathcal{Q}$ y reemplazar ψ por $\psi[f(\bar{x})/x_i]$, donde f es una función $|\bar{x}|$ -aria no usada hasta el momento.

Resolución para Lógica de Primer Orden

- **Forma Clausal**. Escribir φ en forma prenexa, con la matriz en normal conjuntiva

$$\begin{aligned}\varphi &= Q.\psi \\ \psi &= \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)}.\end{aligned}$$

Sea $Sko(\varphi)$ la **skolemización** de φ :

- Mientras exista un cuantificador existencial en Q , sea $\bar{x}$ la lista de variables universalmente cuantificadas en Q que ocurren delante del primer cuantificador existencial $\exists x_i$.
- Eliminar $\exists x_i$ de Q y reemplazar ψ por $\psi[f(\bar{x})/x_i]$, donde f es una función $|\bar{x}|$ -aria no usada hasta el momento.
- Eliminar todos los cuantificadores existenciales, olvidar Q , considerar la matriz obtenida como una fórmula proposicional en forma normal conjuntiva y definir C/Set como antes.

Resolución para Lógica de Primer Orden

Resolución para Lógica de Primer Orden

- Sea $ClSet^*(\varphi)$ be el conjunto mas pequeño que contiene a $ClSet(\varphi)$ y esta clausurado bajo las reglas (RES) y (FAC):

$$\frac{Cl_1 \cup \{N\} \in ClSet^*(\varphi) \quad Cl_2 \cup \{\neg M\} \in ClSet^*(\varphi)}{(Cl_1 \cup Cl_2)\theta \in ClSet^*(\varphi)}$$

$$\frac{Cl \cup \{N, M\}}{(Cl \cup \{N\})\theta}$$

donde θ es el unificador más general de M y N .

Resolución para Lógica de Primer Orden

- Sea $ClSet^*(\varphi)$ be el conjunto mas pequeño que contiene a $ClSet(\varphi)$ y esta clausurado bajo las reglas (RES) y (FAC):

$$\frac{Cl_1 \cup \{N\} \in ClSet^*(\varphi) \quad Cl_2 \cup \{\neg M\} \in ClSet^*(\varphi)}{(Cl_1 \cup Cl_2)\theta \in ClSet^*(\varphi)}$$

$$\frac{Cl \cup \{N, M\}}{(Cl \cup \{N\})\theta}$$

donde θ es el unificador más general de M y N .

- **Teorema:** $\forall \varphi, \varphi$ es inconsistente sii $\{\} \in ClSet^*(\varphi)$.

Un ejemplo sin problemas

Un ejemplo sin problemas

La traduccion standard de $\Box p \wedge \Diamond \neg p$

1. $\{\neg R(c, y), P(y)\},$
2. $\{R(c, d)\},$
3. $\{\neg P(d)\}$

Un ejemplo sin problemas

La traduccion standard de $\Box p \wedge \Diamond \neg p$

1. $\{\neg R(c, y), P(y)\},$
2. $\{R(c, d)\},$
3. $\{\neg P(d)\}$

Por resolución obtenemos dos clausulas nuevas:

4. $\{P(d)\}$
5. $\{\neg R(c, d)\}$

Un ejemplo sin problemas

La traduccion standard de $\Box p \wedge \Diamond \neg p$

1. $\{\neg R(c, y), P(y)\},$
2. $\{R(c, d)\},$
3. $\{\neg P(d)\}$

Por resolución obtenemos dos clausulas nuevas:

4. $\{P(d)\}$
5. $\{\neg R(c, d)\}$

Clausulas 3 y 4, (y clausulas 2 y 5 también) resuelven para dar $\{\}$

Un ejemplo con problemas

Un ejemplo con problemas

La traducción standard de $\Box(p \rightarrow \Diamond p)$

1. $\{\neg R(c, y), \neg P(y), R(y, f(y))\}$
2. $\{\neg R(c, z), \neg P(z), P(f(z))\}$

Un ejemplo con problemas

La traducción standard de $\Box(p \rightarrow \Diamond p)$

1. $\{\neg R(c, y), \neg P(y), R(y, f(y))\}$
2. $\{\neg R(c, z), \neg P(z), P(f(z))\}$

Por resolución obtenemos dos clausulas nuevas (f^n es f aplicada n veces):

3. $\{\neg R(c, c), \neg P(c), \neg P(f(c)), P(f^2(c))\}$
4. $\{\neg R(c, f(z)), R(f(z), f^2(z)), \neg R(c, z), \neg P(z)\}.$

Un ejemplo con problemas

La traducción standard de $\Box(p \rightarrow \Diamond p)$

1. $\{\neg R(c, y), \neg P(y), R(y, f(y))\}$
2. $\{\neg R(c, z), \neg P(z), P(f(z))\}$

Por resolución obtenemos dos clausulas nuevas (f^n es f aplicada n veces):

3. $\{\neg R(c, c), \neg P(c), \neg P(f(c)), P(f^2(c))\}$
4. $\{\neg R(c, f(z)), R(f(z), f^2(z)), \neg R(c, z), \neg P(z)\}$.

Clausulas 2 y 4 resuelven para producir

5. $\{\neg R(c, f^2(z)), R(f^2(z), f^3(z)), \neg R(c, f(z)), \neg R(c, z), \neg P(z)\}$.

Cual es el problema?

Cual es el problema?



Layering

Layering

- ▶ **Hardwire** la idea de layering

Layering

- ▶ **Hardwire** la idea de layering
- ▶ **Como?** haciendo las subformulas en distintos layers sintacticamente diferentes:

$$Tr(p, n) := p_n$$

$$Tr(\neg\psi, n) := \neg Tr(\psi, n)$$

$$Tr(\psi \wedge \chi, n) := Tr(\psi, n) \wedge Tr(\chi, n)$$

$$Tr(\Diamond\psi, n) := \Diamond_{n+1} Tr(\psi, n + 1)$$

Layering

- ▶ **Hardwire** la idea de layering
- ▶ **Como?** haciendo las subformulas en distintos layers sintacticamente diferentes:

$$Tr(p, n) := p_n$$

$$Tr(\neg\psi, n) := \neg Tr(\psi, n)$$

$$Tr(\psi \wedge \chi, n) := Tr(\psi, n) \wedge Tr(\chi, n)$$

$$Tr(\Diamond\psi, n) := \Diamond_{n+1} Tr(\psi, n+1)$$

- ▶ Ahora, traducir dentro de LPO usando ST (la composicion de Tr y ST es lo que llamamos la layered translation)

Layering

- ▶ **Hardwire** la idea de layering
- ▶ **Como?** haciendo las subformulas en distintos layers sintacticamente diferentes:

$$Tr(p, n) := p_n$$

$$Tr(\neg\psi, n) := \neg Tr(\psi, n)$$

$$Tr(\psi \wedge \chi, n) := Tr(\psi, n) \wedge Tr(\chi, n)$$

$$Tr(\Diamond\psi, n) := \Diamond_{n+1} Tr(\psi, n + 1)$$

- ▶ Ahora, traducir dentro de LPO usando ST (la composicion de Tr y ST es lo que llamamos la layered translation)
- ▶ **Teorema:** φ es satisfacible sii la layered translation de φ lo es.

Volvemos al ejemplo

Volvemos al ejemplo

La layered translation de $\Box(p \rightarrow \Diamond p)$

- ▶ $\{\neg R_1(c, y), \neg P_1(y), R_2(y, f(y))\}$
- ▶ $\{\neg R_1(c, z), \neg P_1(z), P_2(f(z))\}$

No podemos generar ninguna nueva clausula y el clause set esta saturado. La clausula original es satisfacible.

Un metodo de decision?

Un metodo de decision?

- La layered translation más resolución **no es** un método de decisión para la lógica modal básica.

1. $\Box(q_0 \vee q_1 \vee \Box\perp)$;
2. $\Box(\neg q_0 \vee q_1 \vee \Box\perp)$;
3. $\Box(q_0 \vee \neg q_1 \vee \Box\perp)$;
4. $\Diamond\top$.

Más problemas

Más problemas

La fórmula es satisfacible

1. $\neg R_1(a, x_1), Q_{01}(x_1), Q_{11}(x_1), \neg R_2(x_1, x_2)$
2. $\neg R_1(a, x_3), \neg Q_{01}(x_3), Q_{11}(x_3), \neg R_2(x_3, x_4)$
3. $\neg R_1(a, x_5), Q_{01}(x_5), \neg Q_{11}(x_5), \neg R_2(x_5, x_6)$
4. $R_1(a, b)$

Simple resolución genera clausulas con literales redundantes.

- $\neg R_2(x_3, x_4), \neg R_2(x_5, x_6), \dots$

Entonces?

Entonces?

- ▶ La layered translation más resolución **no es** un método de decisión para la lógica modal básica.
- ▶ **Pero:** La layered translation mas ordered resolution, y una forma restringida de condensamiento y renombramiento, si es método de decisión.

De todas formas la layered translation puede ser implementada como una **optimización** sobre otros métodos de decisión.

Testing

Testing

- Muchas veces, un simple estudio teórico, no da indicaciones sobre la efectividad real de sistemas complejos como los demostradores automaticos que estamos usando.

Testing

- ▶ Muchas veces, un simple estudio teórico, no da indicaciones sobre la efectividad real de sistemas complejos como los demostradores automaticos que estamos usando.
- ▶ En estos casos, necesitamos evaluación empírica de su performance.

Testing

- ▶ Muchas veces, un simple estudio teórico, no da indicaciones sobre la efectividad real de sistemas complejos como los demostradores automaticos que estamos usando.
- ▶ En estos casos, necesitamos evaluación empírica de su performance.
- ▶ En Sat proposicional, tenemos un conocimiento detallado sobre los posibles experimentos (el fenómeno de phase transition).

Testing

- ▶ Muchas veces, un simple estudio teórico, no da indicaciones sobre la efectividad real de sistemas complejos como los demostradores automaticos que estamos usando.
- ▶ En estos casos, necesitamos evaluación empírica de su performance.
- ▶ En Sat proposicional, tenemos un conocimiento detallado sobre los posibles experimentos (el fenómeno de phase transition).
- ▶ En Sat modal recién estamos en los comienzos.

Testing

- ▶ Muchas veces, un simple estudio teórico, no da indicaciones sobre la efectividad real de sistemas complejos como los demostradores automaticos que estamos usando.
- ▶ En estos casos, necesitamos evaluación empírica de su performance.
- ▶ En Sat proposicional, tenemos un conocimiento detallado sobre los posibles experimentos (el fenómeno de phase transition).
- ▶ En Sat modal recién estamos en los comienzos.
- ▶ Cómo podemos testear empíricamente, en forma ordenada?

Eligiendo el input

Eligiendo el input

- ▶ Ejemplos de la **vida real o aplicaciones** previas de modelado, y variantes sobre estas.
 - Son pocos y no pueden ser generados (o existen o no existen).
 - Pueden mostrar patrones interesantes para optimizaciones en casos particulares. Proveen mediciones reales de performance.

Eligiendo el input

- ▶ Ejemplos de la **vida real o aplicaciones** previas de modelado, y variantes sobre estas.
 - Son pocos y no pueden ser generados (o existen o no existen).
 - Pueden mostrar patrones interesantes para optimizaciones en casos particulares. Proveen mediciones reales de performance.
- ▶ inputs generados **a mano** que intentan presentar problemas difíciles al demostrador.
 - Son difíciles de generar. No son exhaustivos.
 - Su estructura es bien conocida. Permiten gradualmente incrementar la complejidad del problema.

Eligiendo el input

- ▶ Ejemplos de la **vida real o aplicaciones** previas de modelado, y variantes sobre estas.
 - Son pocos y no pueden ser generados (o existen o no existen).
 - Pueden mostrar patrones interesantes para optimizaciones en casos particulares. Proveen mediciones reales de performance.
- ▶ inputs generados **a mano** que intentan presentar problemas difíciles al demostrador.
 - Son difíciles de generar. No son exhaustivos.
 - Su estructura es bien conocida. Permiten gradualmente incrementar la complejidad del problema.
- ▶ inputs generados sistemáticamente **en forma random**
 - Pueden estar condicionados a perturbaciones de los parametros.
 - Permiten generar facilmente muchas instancias, cubriendo ampliamente el espacio de posibilidades. Pueden proveer casos atipicos.

Testing

Testing

Actualmente, las metodologías de testing más importantes se basan en la generación random de formulas.

Testing

Actualmente, las metodologías de testing más importantes se basan en la generación random de formulas.

- Cuales son las propiedades que un buen set de testeo debe tener?
 - Reproducibilidad.

Testing

Actualmente, las metodologías de testing más importantes se basan en la generación random de formulas.

- ▶ Cuales son las propiedades que un buen set de testeo debe tener?
 - Reproducibilidad.
 - Representatividad
 - modal vs. proposicional,
 - Valid vs. not-valid balance
 - satisfacibles en modelos de estructuras diferentes.

Testing

Actualmente, las metodologías de testing más importantes se basan en la generación random de formulas.

- ▶ Cuales son las propiedades que un buen set de testeo debe tener?
 - Reproducibilidad.
 - Representatividad
 - modal vs. proposicional,
 - Valid vs. not-valid balance
 - satisfacibles en modelos de estructuras diferentes.
 - Dificultad controlada
 - . . .

Testing

Testing

- Heuerding and Schwendimann (**HS**) crearon un test set de formulas parametrizadas disenadas a mano. Fueron usadas en la comparacion TANCS de 1998, pero son en la actualidad demasiado faciles.

Testing

- ▶ Heuerding and Schwendimann (**HS**) crearon un test set de formulas parametrizadas disenadas a mano. Fueron usadas en la comparacion TANCS de 1998, pero son en la actualidad demasiado faciles.
- ▶ Guinchiglia et al., Hustadt and Schmidt, Horrocks et al. definieron (y luego redefinieron) un generador random **CNF** para logicas modales basado en el generador CNF para logica proposicional.

Testing

- ▶ Heuerding and Schwendimann (**HS**) crearon un test set de formulas parametrizadas disenadas a mano. Fueron usadas en la comparacion TANCS de 1998, pero son en la actualidad demasiado faciles.
- ▶ Guinchiglia et al., Hustadt and Schmidt, Horrocks et al. definieron (y luego redefinieron) un generador random **CNF** para logicas modales basado en el generador CNF para logica proposicional.
- ▶ Massacci introdujo un generador random de formulas modales por traduccion de **QBF**, que fue usado en TANCS 1999 y 2000.

De vuelta a la idea de Layering

De vuelta a la idea de Layering

- Veamos si layering-sobre-un-metodo-de-decision-preexistente funciona.

De vuelta a la idea de Layering

- ▶ Veamos si layering-sobre-un-metodo-de-decision-preexistente funciona.
- ▶ Tests realizados en una Sun Ultra II (300 Mhz), con 1Gb RAM, Solaris 5.2.5

De vuelta a la idea de Layering

- ▶ Veamos si layering-sobre-un-metodo-de-decision-preexistente funciona.
- ▶ Tests realizados en una Sun Ultra II (300 Mhz), con 1Gb RAM, Solaris 5.2.5
- ▶ Demostradores usados (ambos demostradores para logica de primer orden basados en resolucion)
 - Bliksem 1.10a fijando el procedimiento de decision para el GF.
 - SPASS 1.0.3 en modo auto.

20 problemas no satisfacibles en Bliksem

20 problemas no satisfacibles en Bliksem

Prob #	Average Time		M	Average Clauses		M
	Layered	Standard		Layered	Standard	
1	0.238	0.277	0	192	676	1
2	0.381	0.423	0	336	787	0
3	0.545	0.582	0	401	852	0
4	0.799	12.010	1	970	6634	1
5	1.360	38.043	1	1470	10689	1
6	1.922	6.119	1	763	5743	1
7	2.418	37.281	1	949	11558	1
8	5.140	27294.000	4	4117	94581	1
9	7.319	23625.000	4	5070	90663	1
10	10.087	9536.800	3	5948	72433	1
11	6.887	1132.800	2	1815	11346	1
12	9.186	1615.000	2	2198	18224	1
13	10.716	190.870	1	1992	21093	1
14	0.608	0.757	0	341	1220	1
15	1.138	3.225	0	689	4518	1
16	3.973	8044.600	3	2936	67882	1
17	5.781	27107.000	4	3823	109900	1
18	7.271	31258.000	4	4457	103250	1
19	6.951	2237.400	3	2245	19483	1
20	10.586	154.810	1	1989	20623	1

Testeando con Random QBF

Testeando con Random QBF

- Traducción modal de fórmulas de QBF generadas en forma random.

Testeando con Random QBF

- ▶ Traducción modal de fórmulas de QBF generadas en forma random.
- ▶ Parametros
 - C : numero de clausulas in la matriz proposicional
 - V : maximo numero de variables por alternacion
 - D : profundidad de alternacion en la formula QBF

Testeando con Random QBF

- ▶ Traducción modal de fórmulas de QBF generadas en forma random.
- ▶ Parametros
 - C : numero de clausulas in la matriz proposicional
 - V : maximo numero de variables por alternacion
 - D : profundidad de alternacion en la formula QBF
- ▶ El Test TANCS $(C, V, D) = (20, 2, 2)$

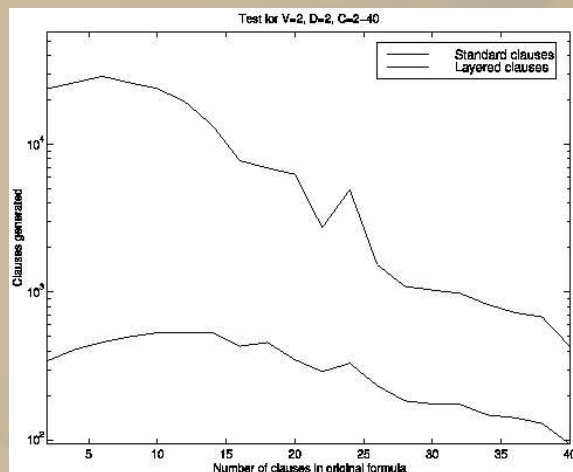
Testeando con Random QBF

- ▶ Traducción modal de fórmulas de QBF generadas en forma random.
- ▶ Parametros
 - C : numero de clausulas in la matriz proposicional
 - V : maximo numero de variables por alternacion
 - D : profundidad de alternacion en la formula QBF
- ▶ El Test TANCS $(C, V, D) = (20, 2, 2)$
- ▶ corridas de 64 instancias por configuración.

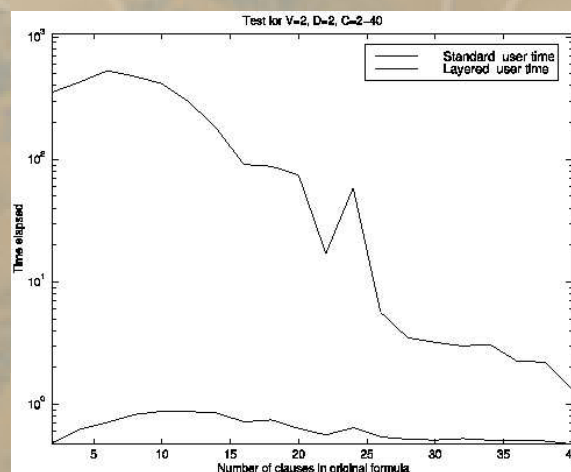
Modal QBF en SPASS

Modal QBF en SPASS

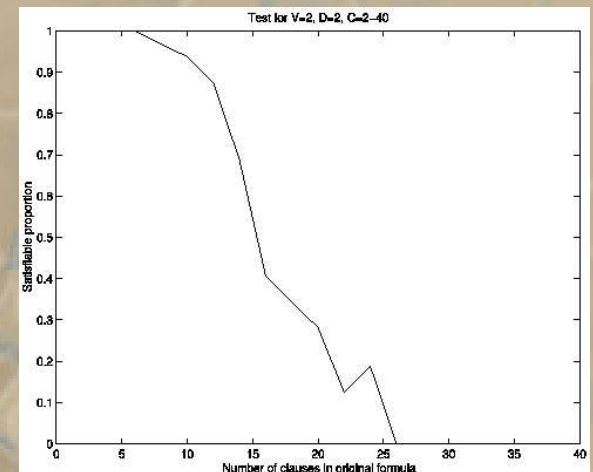
- ▶ Un barrido ($V = D = 2$, $C = 2-40$)
- ▶ Tiempo de CPU y numero de clausulas generadas, relacional vs. layered
- ▶ Layering no cambia la naturaleza del problema



clausulas generadas



tiempo CPU

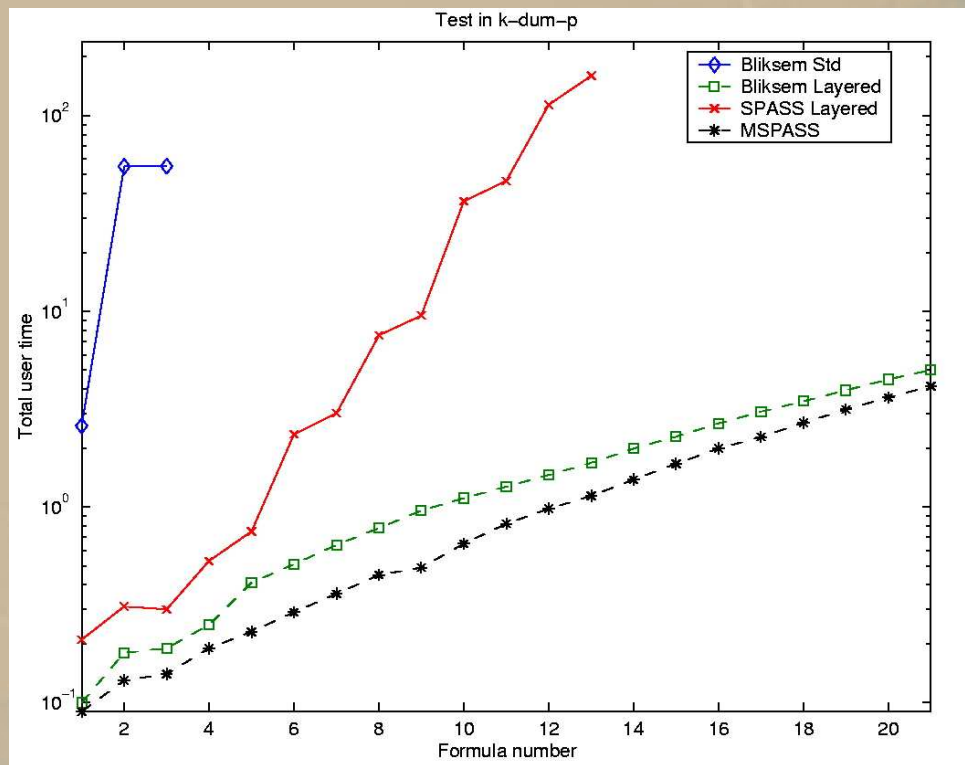


proporcion satisfacible

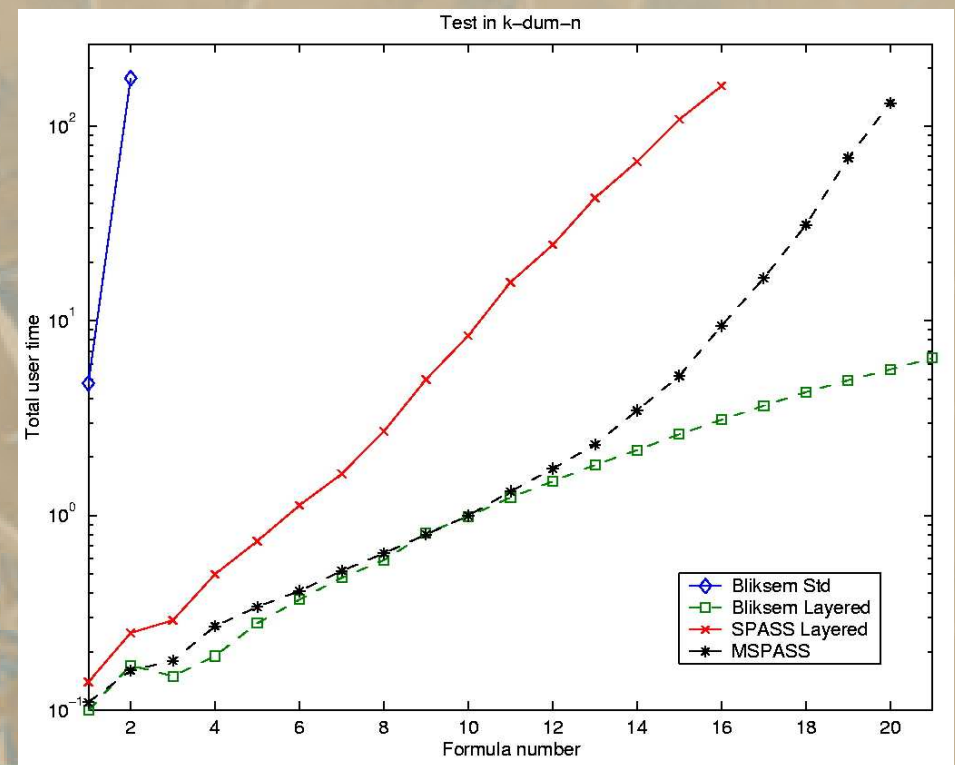
Demostradores diferentes, curvas Diferentes?

Demostradores diferentes, curvas Diferentes?

Bliksem, Bliksem + Layered, SPASS + Layered, MSPASS



k-dum-p



k-dum-n

Que hicimos hoy

Que hicimos hoy

- ▶ Indecidibilidad de LPO
- ▶ Intro a Lógica Modal
- ▶ Fragmentos de LPO
- ▶ Lógica Modal como fragmento de LPO
- ▶ Métodos Indirectos: Layered Translation
- ▶ Testing

Que hicimos hoy

- ▶ Indecidibilidad de LPO
- ▶ Intro a Lógica Modal
- ▶ Fragmentos de LPO
- ▶ Lógica Modal como fragmento de LPO
- ▶ Métodos Indirectos: Layered Translation
- ▶ Testing

Mañana:

- ▶ Método de Tableaux
- ▶ Lógicas para la Descripción
- ▶ Demo: RACER