Lógica Modal Computacional

Teórica 1

Carlos Areces

Computational and Applied Logic Group
Institute for Logic, Language and Computation
University of Amsterdam

September 4, 2001

□. 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 alla 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.

$\Box$. 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 perspective 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 extremandamente variadas,

- aplicaciones clásicas en filosofía y epistemología,
- usos en linguística,
- representación del conocimiento,
- verificación formal de software y hardware.

□. El Curso

Teóricas: El principal énfasis teórico del curso será en los aspectos computacionales de lenguages modales (incluyendo lenguajes modales no standard, como lógicas híbridas y lógicas para descripción). En particular, discutiremos diferentes métodos de inferencia y decisión (directos e indirectos), su complejidad y posibles heurísticas y optimizaciones para problemas de satisfiabilidad.

Prácticas En las clases prácticas presentaremos una selección de herramientas para deduccin autom'atica (RACER, Bliksem, HyLoRes), complementando las nociones teóricas con experiencia de laboratorio.

□. 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.

□. Plan del Curso

Semana 1: Lógicas clásicas. Introducción a la lógica modal. Expresividad y Complejidad. Ejemplos de applicacin.

Semana 2: El método de tableux. Tableux clásicos. Correctitud, completitud y terminación. Lógicas para la descripción. El demostrador automático RACER.

Semana 3: Lógicas modales como fragmentos de lógicas clásicas. La traduccin standard y otras traducciones. El método de resolución clásico. El demostrador automático Bliksem.

Semana 4: Lógicas híbridas. Internalización de tableaux y resolución. El demostrador automático HyLoRes (with all the nitty-gritty details :)).

□. Qué Hacemos Hoy?

- Ventajas y desventajas de las lógicas clásicas.
  - Decidibilidad.
  - Expresividad.
- Introducción a la lógica modal básica.
- Modelos, satisfiabilidad y validez.
- Un menú de lógicas modales.
  - Diferentes operadores modales.
  - Diferentes clases de modelos.
- Ejemplos de applicación.

□. 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

Uno de los principales problemas de LPO (desde nuestro punto de vista) es que decidir si una formula de LPO es satisfacible es indecidible.

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.

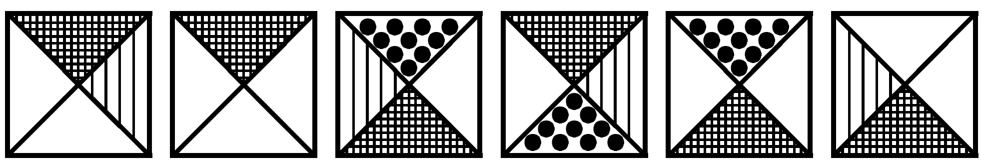
2. demostrar que si X fuera decidible entonces Y sería decidible, dando una codificacion de Y en X.

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

□. Problemas de Tiling

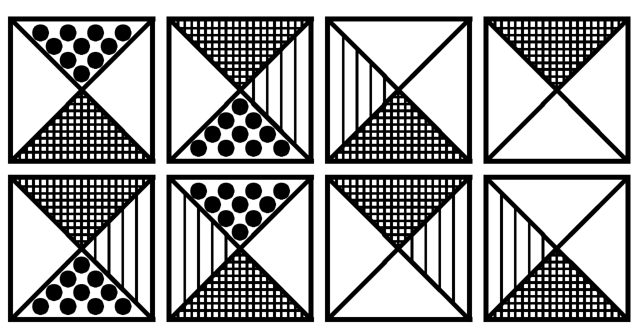
Un problema de tiling es básicamente una especie de rompecabezas

- 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

- La forma general de un problema de tiling es la siguiente

  > Dado un número finito de tipos de tiles $\mathcal{T}$, podemos cubrir una cierta parte de $\mathbb{Z} \times \mathbb{Z}$ de tal forma que tiles adyacentes 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 (puede demostrarse 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 $\mathbb{N} \times \mathbb{N}$ en LPO

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

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

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

$$\bigvee\{t_i(x) = x \mid 1 \leq i \leq n\},$$

$$\bigwedge\{t_i(x) = x \rightarrow \neg(t_j(x) = x) \mid 1 \leq i < j \leq n\},$$

$$\bigvee\{t_i(x) = x \wedge t_j(h(x)) = h(x) \mid H(t_i, t_j)\},$$

$$\bigvee\{t_i(x) = x \wedge t_j(v(x)) = v(x) \mid V(t_i, t_j)\}.$$

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

□. Poder Expresivo

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

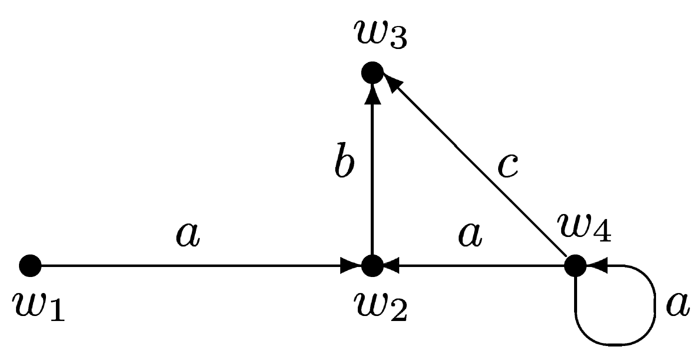
Pero a la vez, LPO puede no tener el **tipo de expresividad** necesaria para una tarea determinada.

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

□. Que son las Lógicas Modales?

Una familia de lenguages con poder expresivo limitado.

Lenguajes especialmente diseñados para hablar sobre propiedades especificas de estructuras relacionales.



□. Que Podemos Describir?

- Condiciones sobre grafos etiquetados
  - Condiciones Globales
  - Condiciones Estructurales
  - Condiciones de Sharing
  - Condiciones Safety
  - Condiciones Liveness

- La estructura de un website

□. Qué es lo que Hace a una Lógica Modal?

- 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".

□. Estructuras Relacionales

Una *estructura relacional* es una tuple $\mathcal{M} = \langle W, \{R_a \mid a \in A\}\rangle$ donde

- $W \neq \emptyset$ es el universo o dominio.
- $\{R_a \mid a \in A\}$ es un conjunto de relaciones sobre W.

$\Box$. La Lógica Modal Básica

- El Lenguage:
  - empezamos con un conjunto de símbolos proposicionales Φ
  - $\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond\phi \mid \Box\phi$
  - 'diamond ϕ' y 'box ϕ' ?!!
- Modelos:
  - 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

- Interterpretando formulas:

$$
\begin{array}{rcl}
M, w \models p & \text{sii} & w \in V(p) \\
M, w \models \neg\phi & \text{sii} & M, w \not\models \phi \\
M, w \models \phi \wedge \psi & \text{sii} & M, w \models \phi \text{ y } M, w \models \psi \\
M, w \models \Diamond\phi & \text{sii} & \text{para algún } v \text{ tal que } Rwv,\ M, v \models \phi \\
M, w \models \Box\phi & \text{sii} & \text{para todo } v \text{ tal que } Rwv,\ M, v \models \phi
\end{array}
$$

$\square$. Satisfiabilidad y Validez

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

□. Tareas de Inferencia

Dado un lenguage $\mathcal{L}$ 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 gM.

$\Box$. **Validez**

- "Demostrabilidad" en términos axiomáticos. La lógica **K**

  A1 Todas las tautologías de lógica proposicional.

  A2 $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$

  R1 De ϕ y $\phi \to \psi$, inferir ψ

  R2 De ϕ inferir $\Box\phi$

- (Kripke) **K** is a sound and complete axiom system

$\Box$. Validez

- 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 básica posee la bounded model property: si ϕ es satisfacible, entonces es satisfacible en un modelo con a lo sumo $2^{|\phi|}$ elementos.
- Vamos a ver más adelante que la complejidad del problema de validez para **K** es en realidad PSPACE-complete.

□. Lógica Modal y Representación de Conocimiento

'And there is one among them that might have been foaled in the morning of the world. The horses of the Nine cannot vie with him; tireless, swift as the flowing wind. Shadowfax they called him. By day his coat glistens like silver; and by night it is like a shade, and he passes unseen.'

from "The Lord of the Rings," J. R. R. Tolkien

$$\texttt{shadowfax} \rightarrow \left(\begin{array}{c} \langle \mathit{foaled} \rangle \text{ things-existing-in-the-morning-of-the-world} \wedge \\ \text{tireless-things} \ \wedge \ \text{swift-as-the-wind-things} \wedge \\ [h_{\mathit{as\text{-}coat\text{-}like}}] \text{ things-glistening-like-silver-by-day} \wedge \\ [\mathit{has\text{-}coat\text{-}like}] \text{ things-like-shades-by-night} \ \wedge [\mathit{pass\text{-}like}] \text{ unseen-things} \end{array} \right)$$

$$\text{horses-of-the-Nine} \rightarrow \neg \langle \mathit{vie} \rangle \ \texttt{shadowfax}.$$

□. □. Lógica Modal y Verificación de Sistemas

Lógicas modales con operadores temporales (F “en el futuro,” G “simpre en el futuro,” P “en el pasado,” H “siempre en el pasado,” etc.) pueden usarse para modelar la evolución de un sistema y verificar sus propiedades.

Propiedades SAFETY	Propiedades LIVENESS
• “Nada malo pasa”	• “Algo bueno va a pasar”
• Tienen la forma $\mathsf{G}\neg\varphi$	• Classificación
$\mathsf{G}\neg(Esc_1 \wedge Esc_2)$	– Guarantee: $\mathsf{F}\varphi$
• Son generalmente una invariante	– Persistence: $\mathsf{FG}\varphi$
• Permiten “inactividad”	– Responce: $\mathsf{GG}\varphi$
	...

□. Que Hicimos Hoy?

- Presentar el curso.
- Por qué sí/no Lógicas Clásicas?
- Qué es la Lógica Modal.
- Sintaxis, Semántica.
- Problemas de Inferencia, Satisfiabilidad, Validez.

$\square$. Tarea

- Leer
  - Modal Logic, Blackburn, de Rijke y Venema. Ch 1 y 2.
  - Logic Engineering. Ch 1.
- Jugar con Modelos y Formulas Modales.
  - Ejercicios 1.3.1, 1.3.2, 1.3.3, 1.3.4 de Modal Logic.

□. La Próxima Clase . . .

- Lógicas Modales y Fragmentos de Lógicas Clásicas.
- Bisimulación.
- Más sobre Complejidad.