Lógicas Modales

Algoritmos efectivos

Carlos Areces & Raul Fervari

1er cuatrimestre de 2017
Córdoba, Argentina

Repaso

Estuvimos viendo...

- Complejidad de distintas lógicas modales
- En particular, algoritmos óptimos, pero imprácticos!

Hoy vamos a ver

- Algoritmos con peor complejidad
- Pero buen comportamiento empírico en casos promedio

Satisfacibilidad modal ~~adivinando~~ buscando modelos

Algoritmo $\mathsf{NTIME}(f)$ para lógicas con modelos f-acotados

- Dada una fórmula φ:
  1. Adivinar un modelo $\mathcal{M}$ de tamaño a lo sumo $f(|\varphi|)$
  2. Adivinar un w en el dominio de $\mathcal{M}$
  3. Devolver 1 sii $\mathcal{M}, w \models \varphi$
- Obviamente, no sirve como algoritmo efectivo

Algoritmo de Tableaux

- Dada una fórmula φ
  1. Buscar ("backtracking") sistemáticamente un modelo de φ
  2. Devolver 1 sii se encuentra tal modelo
- Es la base de muchos razonadores para lógicas modales
- Varios tipos de tableaux, vamos a ver sólo *tableaux etiquetados*

¿Qué es un algoritmo de tableau (etiquetado)?

- *Reglas de tableaux:* definen un árbol de posibilidades.
- Los nodos de un árbol son, en general:
  - Fórmulas "etiquetadas" $w{:}\psi$, donde w es una etiqueta.
  - "Relaciones" Rwv donde w y v son etiquetas.
- Cada rama del árbol codifica de alguna manera un modelo.
- Las reglas nos dicen:
  1. Cómo *expandir* una rama.
  2. Cómo detectar que una rama no nos sirve (reglas de clash).
- *Algoritmo:* Dada φ, explorar (backtracking) el árbol de $w{:}\varphi$.

Ejemplo de tableaux etiquetado

Lógica modal con pasado

Reglas de expansión

$\wedge \quad \dfrac{w{:}\varphi\wedge\psi}{w{:}\varphi, w{:}\psi} \qquad \vee \quad \dfrac{w{:}\varphi\vee\psi}{w{:}\varphi \mid w{:}\psi}$

$\Diamond \quad \dfrac{w{:}\Diamond\varphi}{Rwv, v{:}\varphi}$ (nuevo v) $\qquad \Diamond^{-1} \quad \dfrac{w{:}\Diamond^{-1}\varphi}{Rvw, v{:}\varphi}$ (nuevo v)

$\Box \quad \dfrac{w{:}\Box\varphi, Rwv}{v{:}\varphi} \qquad \Box^{-1} \quad \dfrac{w{:}\Box^{-1}\varphi, Rvw}{v{:}\varphi}$

Regla de clash:

$$clash \quad \frac{w{:}p, w{:}\neg p}{\bot}$$

- Asumimos fórmulas en *negation normal form*.
- Las reglas $\Diamond$ y $\Diamond^{-1}$ se usan sólo si no existe tal v en la rama.

Ejemplo

Ejercicio.
Decidir si $\varphi = p_1 \wedge \Diamond p_2 \wedge \Diamond\Box^{-1}\Box(\neg p_2 \vee \Box^{-1}\neg p_1)$ es satisfacible.

Tableau etiquetado para la lógica modal con pasado

Completitud

Teorema (Completitud)

Si Γ es una rama abierta y saturada para φ, φ es satisfacible.

Demostración. Extraemos un modelo de Γ.

Sea $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma\rangle$ donde
$$\begin{aligned} W_\Gamma &= \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma &= \{(w,v) \mid Rwv \in \Gamma\} \\ V_\Gamma(p) &= \{w \mid w{:}p \in \Gamma\} \end{aligned}$$

Sea ψ la fórmula *más pequeña* t.q. $w{:}\psi \in \Gamma$ y $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (porque en ese caso $w \in V(p)$) y $\psi \neq \neg p$ (habría clash)
- $\psi \neq \psi_1 \vee \psi_2$ porque tendríamos $w{:}\psi_i \in \Gamma$ y no sería mínima
- $\psi \neq \psi_1 \wedge \psi_2$ por razones análogas
- $\psi \neq \Diamond\chi$ porque tendríamos $Rwv, v{:}\chi \in \Gamma$ y no sería mínima
- $\psi \neq \Box\chi$, $\psi \neq \Box^{-1}\chi$ y $\psi \neq \Diamond^{-1}\chi$ por razones análogas.

Luego, no existe tal fórmula; $w{:}\psi \in \Gamma$ implica $\mathcal{M}, w \models \psi$

Tableau etiquetado para la lógica modal con pasado

Terminación

Teorema

Toda rama saturada de un tableaux para φ es finita.

Demostración. Sea Γ una rama saturada y
$\mathsf{LABEL}(w) = \{\psi \mid w{:}\psi \in \Gamma\}$.

1. $\mathsf{LABEL}(w)$ es finito porque son todas subfórmulas de φ.
2. Luego, $\{v \mid Rwv \in \Gamma\}$ es finito (ver nota sobre $\Diamond$ y $\Diamond^{-1}$).
3. Entonces, Γ es infinito sii existe una cadena $w_1, w_2, \ldots$ tal que w_i *generó* a w_{i+1} usando la regla $\Diamond$ ó la regla $\Diamond^{-1}$.
4. Pero si w genera a v, $d(\mathsf{LABEL}(w)) > d(\mathsf{LABEL}(v))$ (sale por inducción en la derivación de Γ, d es profundidad modal).
5. Por lo tanto, para algún j, $d(\mathsf{LABEL}(w_j)) = 0$.

Detalles de implementación

¿Importa el orden en que aplicamos las reglas?

- No afecta la terminación del algoritmo.
- Sí afecta el tamaño del árbol generado!
  - Considerar: $(p_1 \vee p_2) \wedge ((p_3 \vee p_4) \wedge ((p_5 \vee p_6) \wedge (p \wedge \neg p)))$
  - ¿Qué sucede si siempre preferimos aplicar $\wedge$ antes que $\vee$?
  - ¿Qué sucede si siempre preferimos aplicar $\vee$ antes que $\wedge$?

Heurísticas básicas

- Usar reglas sin branching (e.g., $\wedge$) antes que aquellas con branching (como $\vee$)
- Usar reglas proposicionales (e.g., $\wedge$ y $\vee$) antes que reglas modales (como $\Diamond$ y $\Box$)

Optimizaciones

- Las reglas $\wedge$, $\vee$ y *clash* son un tableaux proposicional
- Pero es preferible DPLL para razonamiento proposicional
- Los demostradores basados en tableaux incorporan elementos de DPLL:
  - Branching semántico (una forma de splitting)
  - Backjumping
  - Caching

Terminación en casos más complejos

Un tableaux para K sobre la clase de modelos transitivos ($\mathsf{K4}$)

$$\wedge\ \frac{w{:}\varphi \wedge \psi}{w{:}\varphi, w{:}\psi} \qquad \vee\ \frac{w{:}\varphi \vee \psi}{w{:}\varphi \mid w{:}\psi} \qquad \textit{clash}\ \frac{w{:}p, w{:}\neg p}{\bot}$$

$$\Diamond\ \frac{w{:}\Diamond\varphi}{Rwv, v{:}\varphi}\ (\text{nuevo } v) \qquad \Box_4\ \frac{w{:}\Box\varphi, Rwv}{v{:}\varphi, v{:}\Box\varphi}$$

Teorema
Este tableaux es completo para $\mathsf{K4}$.

Demostración Ejercicio!

Terminación en casos más complejos
Blocking

- ¿Podemos repetir el argumento de terminación?
- ¿Qué sucede al ejecutar este tableaux sobre $w{:}(\Diamond p \wedge \Box\Diamond p)$?
- *Conclusión:* una rama abierta saturada puede no ser finita!

Técnicas de blocking

- Se usan para garantizar terminación en implementaciones
- *Idea:*
  - Algunas $w{:}\varphi$ pueden “bloquearse” o “desbloquearse”
  - Algunas reglas no se aplican sobre fórmulas bloqueadas
- Muchos tipos de blocking
  - subset blocking
  - dynamic blocking
  - ...
- En cada caso se debe probar terminación... y completitud!