

Lógicas Modales

Algoritmos efectivos

Carlos Areces

1er cuatrimestre de 2012

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

Tableaux modales

Algoritmo $\text{NTIME}(f)$ 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

Satisfacibilidad modal ~~adivinando~~ buscando modelos

Tableaux modales

Algoritmo 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.

¿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.

¿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.

¿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).

¿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

$$\wedge \frac{w:\varphi \wedge \psi}{w:\varphi, w:\psi}$$

$$\vee \frac{w:\varphi \vee \psi}{w:\varphi \mid w:\psi}$$

$$\diamond \frac{w:\diamond\varphi}{Rwv, v:\varphi} \quad (\text{nuevo } v)$$

$$\diamond^{-1} \frac{w:\diamond^{-1}\varphi}{Rvw, v:\varphi} \quad (\text{nuevo } v)$$

$$\square \frac{w:\square\varphi, Rvw}{v:\varphi}$$

$$\square^{-1} \frac{w:\square^{-1}\varphi, Rvw}{v:\varphi}$$

Reglas de clash

$$\text{clash} \frac{w:p, w:\neg p}{\perp}$$

- 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 $\varphi = p_1 \wedge \diamond p_2 \wedge \diamond \square^{-1} \square (\neg p_2 \vee \square^{-1} \neg p_1)$ es satisfacible.

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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

$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$

$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$

$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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

$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$

$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$

$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$

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

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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

$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$

$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$

$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$

- 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)

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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

$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$

$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$

$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$

- 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

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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
$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$
$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$
$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$
- 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

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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
$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$
$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$
$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$
- 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 $R w v, v:\chi \in \Gamma$ y no sería mínima

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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
$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$
$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$
$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$
- 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 $R w v, v:\chi \in \Gamma$ y no sería mínima
 - $\psi \neq \square \chi, \psi \neq \square^{-1} \chi$ y $\psi \neq \diamond^{-1} \chi$ por razones análogas.

Tableau etiquetado para la lógica modal con pasado

Compleitud

Teorema (Compleitud)

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
$$W_\Gamma = \{w \mid w:\varphi \in \Gamma\}$$
$$R_\Gamma = \{(w, v) \mid R w v \in \Gamma\}$$
$$V_\Gamma(p) = \{w \mid w:p \in \Gamma\}$$
- 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 $R w v, v:\chi \in \Gamma$ 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.

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 $\text{LABEL}(w) = \{\psi \mid w:\psi \in \Gamma\}$.

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 $\text{LABEL}(w) = \{\psi \mid w:\psi \in \Gamma\}$.

- 1 $\text{LABEL}(w)$ es finito porque son todas subfórmulas de φ .

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 $\text{LABEL}(w) = \{\psi \mid w:\psi \in \Gamma\}$.

- 1 $\text{LABEL}(w)$ es finito porque son todas subfórmulas de φ .
- 2 Luego, $\{v \mid R w v \in \Gamma\}$ es finito (ver nota sobre \diamond y \diamond^{-1}).

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 $\text{LABEL}(w) = \{\psi \mid w:\psi \in \Gamma\}$.

- 1 $\text{LABEL}(w)$ es finito porque son todas subfórmulas de φ .
- 2 Luego, $\{v \mid R w v \in \Gamma\}$ es finito (ver nota sobre \diamond y \diamond^{-1}).
- 3 Entonces, Γ es infinito sii existe una cadena w_1, w_2, \dots tal que w_i generó a w_{i+1} usando la regla \diamond ó la regla \diamond^{-1} .

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 $\text{LABEL}(w) = \{\psi \mid w:\psi \in \Gamma\}$.

- 1 $\text{LABEL}(w)$ es finito porque son todas subfórmulas de φ .
- 2 Luego, $\{v \mid R w v \in \Gamma\}$ es finito (ver nota sobre \diamond y \diamond^{-1}).
- 3 Entonces, Γ es infinito sii existe una cadena w_1, w_2, \dots 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 , $d(\text{LABEL}(w)) > d(\text{LABEL}(v))$ (sale por inducción en la derivación de Γ , d es profundidad modal).

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 $\text{LABEL}(w) = \{\psi \mid w:\psi \in \Gamma\}$.

- 1 $\text{LABEL}(w)$ es finito porque son todas subfórmulas de φ .
- 2 Luego, $\{v \mid R w v \in \Gamma\}$ es finito (ver nota sobre \diamond y \diamond^{-1}).
- 3 Entonces, Γ es infinito sii existe una cadena w_1, w_2, \dots 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 , $d(\text{LABEL}(w)) > d(\text{LABEL}(v))$ (sale por inducción en la derivación de Γ , d es profundidad modal).
- 5 Por lo tanto, para algún j , $d(\text{LABEL}(w_j)) = 0$.

Detalles de implementación

¿Importa el orden en que aplicamos las reglas?

Detalles de implementación

¿Importa el orden en que aplicamos las reglas?

- No afecta la terminación del algoritmo.

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 ?

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 \square)

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 (K4)

$$\wedge \frac{w:\varphi \wedge \psi}{w:\varphi, w:\psi}$$

$$\vee \frac{w:\varphi \vee \psi}{w:\varphi \mid w:\psi}$$

$$\text{clash} \frac{w:p, w:\neg p}{\perp}$$

$$\diamond \frac{w:\diamond\varphi}{Rwv, v:\varphi} \quad (\text{nuevo } v)$$

$$\Box_4 \frac{w:\Box\varphi, Rwv}{v:\varphi, v:\Box\varphi}$$

Terminación en casos más complejos

Un tableaux para K sobre la clase de modelos transitivos (K4)

$$\wedge \frac{w:\varphi \wedge \psi}{w:\varphi, w:\psi}$$

$$\vee \frac{w:\varphi \vee \psi}{w:\varphi \mid w:\psi}$$

$$\text{clash} \frac{w:p, w:\neg p}{\perp}$$

$$\diamond \frac{w:\diamond\varphi}{Rwv, v:\varphi} \quad (\text{nuevo } v)$$

$$\Box_4 \frac{w:\Box\varphi, Rww}{v:\varphi, v:\Box\varphi}$$

Teorema

Este tableaux es completo para K4.

Terminación en casos más complejos

Un tableaux para K sobre la clase de modelos transitivos (K4)

$$\wedge \frac{w:\varphi \wedge \psi}{w:\varphi, w:\psi}$$

$$\vee \frac{w:\varphi \vee \psi}{w:\varphi \mid w:\psi}$$

$$\text{clash} \frac{w:p, w:\neg p}{\perp}$$

$$\diamond \frac{w:\diamond\varphi}{Rwv, v:\varphi} \quad (\text{nuevo } v)$$

$$\Box_4 \frac{w:\Box\varphi, Rwv}{v:\varphi, v:\Box\varphi}$$

Teorema

Este tableaux es completo para K4.

Demostración

Ejercicio!

Terminación en casos más complejos

Blocking

- ¿Podemos repetir el argumento de terminación?

Terminación en casos más complejos

Blocking

- ¿Podemos repetir el argumento de terminación?
- ¿Qué sucede al ejecutar este tableaux sobre $w:(\diamond p \wedge \square \diamond p)$?

Terminación en casos más complejos

Blocking

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

Terminación en casos más complejos

Blocking

- ¿Podemos repetir el argumento de terminación?
- ¿Qué sucede al ejecutar este tableaux sobre $w:(\diamond p \wedge \square \diamond p)$?
- *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!