Lógica prop.

Sistemas Inteligentes I

Tema 5. Lógica proposicional

José A. Montenegro Montes
monte@lcc.uma.es

Resumen

- Introducción
- Fundamentos
- Demostración Teoremas
- Conclusiones

Lógica prop.

Introducción

Agentes lógicos

- Los humanos conocen hechos, y lo que saben les ayuda a actuar
  - Los cerebros humanos llevan a cabo procesos de razonamiento que trabajan con representaciones internas del conocimiento
- La Inteligencia Artificial construye agentes basados en el conocimiento que también son capaces de razonar
  - En esta unidad veremos una lógica sencilla, la lógica proposicional
  - En la siguiente unidad estudiaremos la lógica de primer orden, que permite razonamientos más complejos

Fundamentos

Sintaxis

Sintaxis (I)

- La sintaxis de la lógica proposicional define las fórmulas bien formadas
- Las fórmulas atómicas (también llamadas átomos) consisten de un solo símbolo de proposición: P, Q, $Rains$, W_{35},…
  - Dos símbolos especiales: *True* y *False*
- Las fórmulas compuestas se obtienen de fórmulas más sencillas empleando los paréntesis y las conectivas lógicas
- A continuación se presentan las cinco conectivas que usaremos, en orden de precedencia

Sintaxis (II)

- ¬ (no). Un literal es, o bien una fórmula atómica (literal positivo), o bien su negación (literal negativo)
- ∧ (y). Una fórmula cuya conectiva de nivel más alto es ∧, se denomina conjunción
- ∨ (o). Una fórmula cuya conectiva de nivel más alto es ∨, se denomina disyunción
- ⇒ (implica). Una fórmula del tipo $\alpha \Rightarrow \beta$ se llama implicación, donde α es la premisa o antecedente, y β es la conclusión o consecuencia
- ⇔ (si y sólo si). Una fórmula cuya conectiva de nivel más alto es ⇔, se denomina bicondicional

Fundamentos

Semántica

Semántica (I)

- La semántica define las reglas para determinar la verdad de una fórmula con respecto a un modelo particular
- En la lógica proposicional, un modelo fija el valor de verdad (verdadero o falso) de todos los símbolos de proposición
  - *True* es verdadero en todo modelo, y *False* es falso en todo modelo
  - El valor de verdad de los demás símbolos lo especifica el modelo
- A continuación se presentan las reglas para calcular el valor de verdad de fórmulas compuestas en un modelo *m*

Semántica (II)

- $\neg P$ es verdadero sii P es falso en m
- $P \wedge Q$ es verdadero sii tanto P como Q son verdaderos en m
- $P \vee Q$ es verdadero sii P o bien Q son verdaderos en m
- $P \Rightarrow Q$ es verdadero a menos que P sea verdadero y Q sea falso en m
- $P \Leftrightarrow Q$ es verdadero sii P y Q son ambos verdaderos o ambos falsos en m

Semántica (III)

Tablas de Verdad

A	B	$A \wedge B$
V	V	V
V	F	F
F	V	F
F	F	F

A	B	$A \vee B$
V	V	V
V	F	V
F	V	V
F	F	F

A	$\neg A$
V	F
F	V

A	B	$A \to B$
V	V	V
V	F	F
F	V	V
F	F	V

A	B	$A \leftrightarrow B$
V	V	V
V	F	F
F	V	F
F	F	V

Semántica (IV)

- Si una fórmula α es verdadera en un modelo m, decimos que ***m* satisface α**
- Decimos que la fórmula **β se infiere de α** sii en todo modelo en el que α es verdadera, β también es verdadera. Lo notamos como $\alpha \models \beta$
- Una sentencia es válida sii es verdadera en todos los modelos; en tal caso decimos que es una tautología
- Una sentencia es **satisfacible** sii es verdadera en algún modelo
- Por último, se cumple que $\alpha \models \beta$ sii $(\alpha \wedge \neg \beta)$ es insatisfacible

Semántica (resumen)

- Una **interpretación** de una fórmula P es una asignación de valores verdad a todas las variables de P.
  - Entonces, una interpretación es una línea en la tabla de verdad.
- Un **modelo** de una formula P es una interpretación de P si P es V con respecto a esa interpretación.
- Una fórmula proposicional es **satisfacible** si toma el valor V para alguna interpretación
- Una formula proposicional es **insatisfacible** si no es satisfacible
- Una formula A es **tautología** (valida) si y solo si ¬A es insatisfacible

Ejemplo (Ejercicio 1)

- Demuestra que las siguientes fórmulas bien formadas son tautologías:
  - $[P \wedge (P \Rightarrow Q)] \Rightarrow Q$
  - $(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P)$

P	Q	$P \Rightarrow Q$	$P \wedge (P \Rightarrow Q)$	$[P \wedge (P \Rightarrow Q)] \Rightarrow Q$
V	V	V	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	V

P	Q	$\neg Q$	$\neg P$	$P \Rightarrow Q$	$\neg Q \Rightarrow \neg P$	$(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P)$
V	V	F	F	V	V	V
V	F	V	F	F	F	V
F	V	F	V	V	V	V
F	F	V	V	V	V	V

Lógica prop.

Fundamentos

Base de conocimiento

Bases de conocimiento (I)

- La base de conocimiento (*knowledge base*, KB) de un agente lógico es un conjunto de fórmulas que incluye:
  - Las reglas del mundo que el agente conoce
  - Los hechos que el agente conoce acerca del mundo, que también se llaman percepciones
- El agente le dice a la KB lo que percibe, es decir, inserta hechos en la KB
- El agente también consulta a la KB acerca de hechos
  - Las respuestas ayudan al agente a tomar decisiones

Bases de conocimiento (II)

- Una base de conocimiento sencilla:
  - Regla_1: $Wet \Leftrightarrow (Rain \vee Flooding)$
  - Regla_2: $Hot \Leftrightarrow (Summer \vee Sunny \vee Fire)$
  - Hecho_1: $\neg Summer$
  - Hecho_2: $\neg Wet$
  - Hecho_3: Hot

Demostración Teoremas

Demostración por resolución

- Una regla de inferencia toma varias fórmulas y produce otra fórmula que puede inferirse de ellas
- Una demostración es una secuencia de fórmulas obtenida por aplicación de reglas de inferencia a partir de una KB
- Sólo consideraremos una regla de inferencia, la regla de resolución
  - La resolución es correcta, es decir, nunca produce una fórmula que no se infiera de la KB
  - También es completa, es decir, cuando se combina con cualquier algoritmo de búsqueda completo, es capaz de alcanzar cualquier fórmula que pueda deducirse de la KB

La regla de inferencia de resolución

- La resolución toma dos cláusulas (disyunciones de literales) tales que hay un literal l_i en la primera cláusula que es la negación de un literal m_j de la segunda cláusula, o sea, l_i y m_j son literales complementarios.

C1. $\neg Q$

C2. $\neg R \vee Q$

C3. $\neg R$ Resolver C1 con C2

La regla de inferencia de resolución

- Produce una cláusula con todos los literales de las dos cláusulas originales excepto los dos literales complementarios.

$$\frac{l_1 \vee \ldots \vee l_k, \quad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

- Las apariciones repetidas de literales son también eliminadas de la cláusula resultante.

C1. $\neg P \vee Q$

C2. $P \vee Q$

C3. Q Resolver C1 con C2

Demostración Teoremas

Forma normal conjuntiva

Forma normal conjuntiva

- La resolución sólo se puede aplicar a cláusulas
- Una sentencia que es una conjunción de cláusulas se dice que está en forma normal conjuntiva (*conjunctive normal form*, CNF)
- Toda fórmula de la lógica proposicional es lógicamente equivalente a una conjunción de cláusulas
  - A continuación se da un algoritmo para convertir a CNF

Conversión a CNF

- Eliminar $\Leftrightarrow$ reemplazando
  - $\alpha \Leftrightarrow \beta$ por $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
- Eliminar $\Rightarrow$ reemplazando
  - $\alpha \Rightarrow \beta$ por $\neg\alpha \vee \beta$
- Mover $\neg$ hacia dentro aplicando repetidamente:
  - $\neg(\neg\alpha) \equiv \alpha$
  - $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$
  - $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$
- Aplicar la distributividad de $\vee$ respecto a $\wedge$:
  - $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Ejemplos

$p \Rightarrow ((q \Rightarrow r) \vee \neg s)$;

$\equiv \quad p \Rightarrow ((\neg q \vee r) \vee \neg s)$;

$\equiv \quad \neg p \vee ((\neg q \vee r) \vee \neg s)$

$\neg(\neg p \wedge (q \wedge \neg (r \wedge s)))$;

$\equiv \quad \neg\neg p \vee \neg (q \wedge \neg (r \wedge s))$;

$\equiv \quad p \vee (\neg q \vee \neg\neg (r \wedge s))$;

$\equiv \quad p \vee (\neg q \vee (r \wedge s))$;

$\equiv \quad p \vee ((\neg q \vee r) \wedge (\neg q \vee s))$;

$\equiv \quad (p \vee \neg q \vee r) \wedge (p \vee \neg q \vee s))$;

Ejemplos

$(p \land q) \lor (p \land \neg q)$

$\equiv \quad ((p \land q) \lor p) \land ((p \land q) \lor \neg q)$

$\equiv \quad ((p \lor p) \land (q \lor p)) \land ((p \lor \neg q) \land (q \lor \neg q))$

$((p \land q) \lor (r \land s)) \lor (\neg q \land (p \lor t))$

Ejercicio

Lógica prop.

Demostración Teoremas

Un algoritmo de resolución

Un algoritmo de resolución

- Nuestro objetivo es demostrar que KB $\models \alpha$. Lo haremos por reducción al absurdo, o sea, demostraremos que KB$\wedge\neg\alpha$ es insatisfacible
  - Primero convertimos KB$\wedge\neg\alpha$ a CNF
  - Después aplicamos la regla de resolución repetidamente
- Hay dos posibles resultados:
  - No se pueden añadir más cláusulas, lo que significa que α no se infiere de KB
  - Se produce la cláusula vacía, lo que significa que α se infiere de KB

Ejemplo (I)

- Restringimos nuestra KB a Regla_1 y Hecho_2
  - Regla_1: $Wet \Leftrightarrow (Rain \lor Flooding)$
  - Hecho_2: $\neg Wet$
- Queremos demostrar $\alpha = \neg Rain$
- Primero mostramos la conversión de KB a CNF

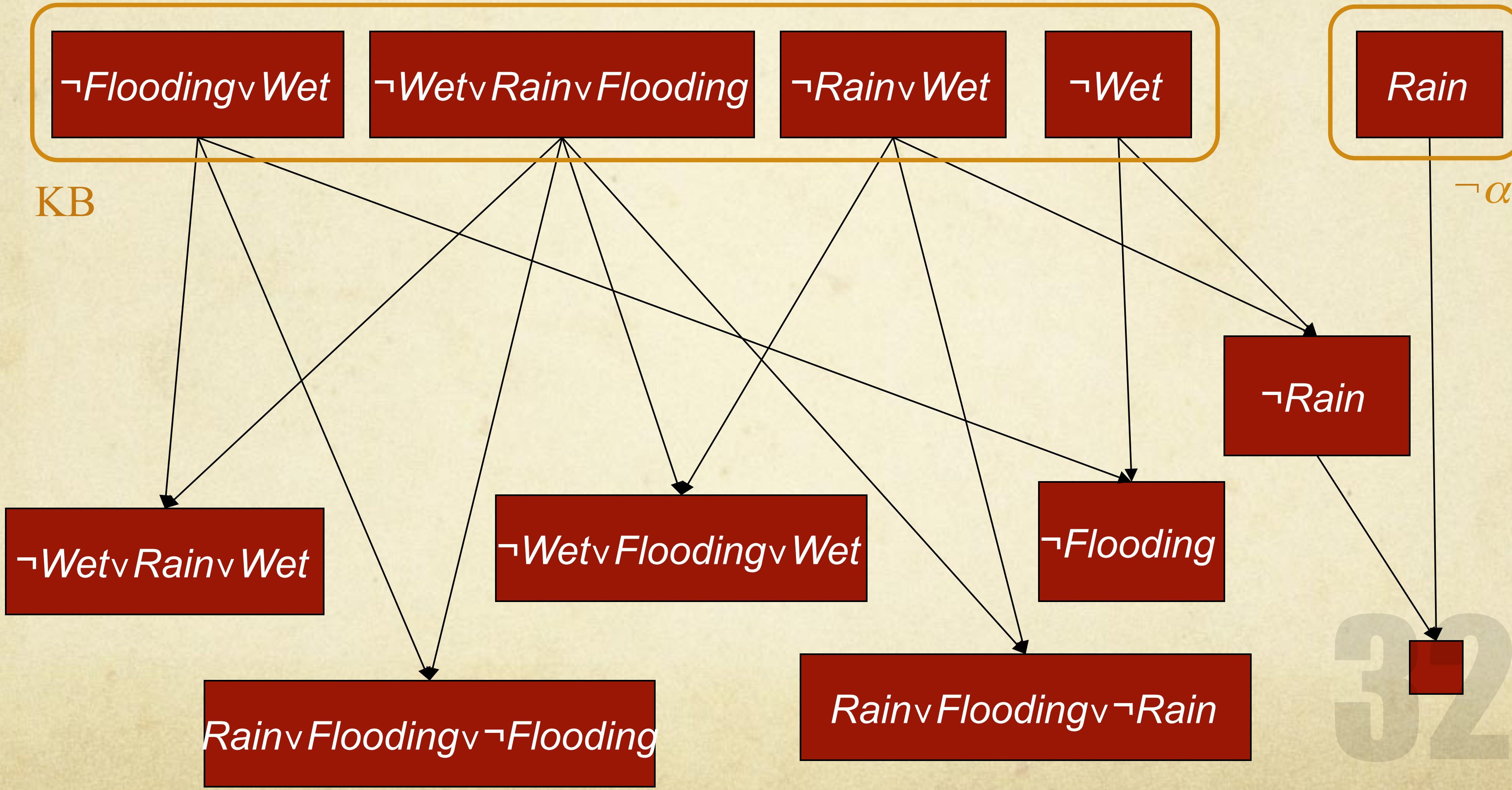
Ejemplo (I)

- En la última transparencia, las cláusulas de la primera fila producen las cláusulas inferiores por resolución
- Si la regla de resolución produce una cláusula en la que aparecen dos literales complementarios, la descartamos porque es lógicamente equivalente a *True*
- Al final se llega a producir la cláusula vacía (recuadro pequeño), lo que quiere decir que α se infiere de KB

Ejemplo (II)

- $[\ Wet \Leftrightarrow (Rain \lor Flooding)\ ] \land \neg Wet$
- $[\ Wet \Rightarrow (Rain \lor Flooding)\ ] \land [\ (Rain \lor Flooding) \Rightarrow Wet\ ] \land \neg Wet$
- $[\ \neg Wet \lor Rain \lor Flooding\ ] \land [\neg(Rain \lor Flooding) \lor Wet\ ] \land \neg Wet$
- $[\ \neg Wet \lor Rain \lor Flooding\ ] \land [\ (\neg Rain \land \neg Flooding) \lor Wet\ ] \land \neg Wet$
- $(\ \neg Wet \lor Rain \lor Flooding\ ) \land (\neg Rain \lor Wet\ ) \land (\neg Flooding \lor Wet\ ) \land \neg Wet$

Ejemplo (III)



¬Flooding∨Wet
¬Wet∨Rain∨Flooding
¬Rain∨Wet
¬Wet
Rain
KB
¬α
¬Rain
¬Wet∨Rain∨Wet
¬Wet∨Flooding∨Wet
¬Flooding
Rain∨Flooding∨¬Flooding
Rain∨Flooding∨¬Rain

Ejercicio 7

$\neg Q \wedge (R \Rightarrow Q)$

$\neg R \Rightarrow P$

$\models$

$\neg R$

C1. ¬ Q

C2. ¬ R ∨ Q

C3. R ∨ P

C4. R

C5. ¬R   Resolver C1 con C2

C6. False   Resolver C4 con C5

Ejercicio 7

$\neg Q \wedge (R \Rightarrow Q)$

$\neg R \Rightarrow P$

$|=$

$\neg R$

QRP	$\neg Q \wedge (R \Rightarrow Q)$	$\neg R \Rightarrow P$	R	$\wedge$
000	1	0	0	0
001	1	1	0	0
010	0	1	1	0
011	0	1	1	0
100	0	0	0	0
101	0	1	0	0
110	0	1	1	0
111	0	1	1	0

Ejercicio 7

$\neg Q \wedge (R \Rightarrow Q)$

$\neg R \Rightarrow P$

$|=$

$\neg R$

QRP	$\neg Q$	$\neg R \vee Q$	$R \vee P$	R	$\wedge$
000	1	1	0	0	0
001	1	1	1	0	0
010	1	0	1	1	0
011	1	0	1	1	0
100	0	1	0	0	0
101	0	1	1	0	0
110	0	1	1	1	0
111	0	1	1	1	0

Ejercicio 8

$P \Leftrightarrow T$

$(T \Rightarrow \neg S) \Leftrightarrow Q$

$\neg P$

$\models$

Q

C1. $\neg P \lor T$
C2. $P \lor \neg T$
C3. $T \lor Q$
C4. $S \lor Q$
C5. $\neg T \lor \neg S \lor \neg Q$
C6. $\neg P$
C7. $\neg Q$

C8. $\neg T$	Resuelvo C2 con C6
C9. Q	Resuelvo C3 con C8
C10. *False*	Resuelvo C7 con C9

$P \Leftrightarrow T$

$(T \Rightarrow \neg S) \Leftrightarrow Q$

$\neg P$

$\models$

Q

PTSQ	$P \Leftrightarrow T$	$(T \Rightarrow \neg S) \Leftrightarrow Q$	$\neg P$	$\neg Q$	$\wedge$
0000	1	0	1	1	0
0001	1	-	1	0	0
0010	1	0	1	1	0
0011	1	-	1	0	0
0100	0	-	1	1	0
0101	0	-	1	0	0
0110	0	-	1	1	0
0111	0	-	1	0	0
1000	0	-	0	1	0
1001	0	-	0	0	0
1010	0	-	0	1	0
1011	0	-	0	0	0
1100	1	-	0	1	0
1101	1	-	0	0	0
1110	1	-	0	1	0
1111	1	-	0	0	0

Conclusión

Sumario

- Los agentes inteligentes necesitan conocimiento acerca de su mundo a fin de tomar buenas decisiones
- El conocimiento se representa en los agentes mediante fórmulas que se almacenan en una base de conocimiento
- La inferencia es el proceso de derivar nuevas fórmulas a partir de las ya conocidas
- La regla de resolución da lugar a un algoritmo de inferencia correcto y completo para la lógica proposicional

Sistemas Inteligentes

José A. Montenegro Montes

monte@lcc.uma.es

Ejemplos

$((p \land q) \lor (r \land s)) \lor (\neg q \land (p \lor t))$

$\equiv\ (((p \land q) \lor r) \land ((p \land q) \lor s)) \lor (\neg q \land (p \lor t))$

$\equiv\ ((p \lor r) \land (q \lor r) \land (p \lor s) \land (q \lor s)) \lor (\neg q \land (p \lor t))$

$\equiv\ ((p \lor r) \lor (\neg q \land (p \lor t)) \land$

$((q \lor r) \lor (\neg q \land (p \lor t)) \land$

$((p \lor s) \lor (\neg q \land (p \lor t)) \land$

$((q \lor s) \lor (\neg q \land (p \lor t))$

$\equiv\ (p \lor r \lor \neg q) \land (p \lor r \lor p \lor t)) \land$

$(q \lor r \lor \neg q) \land (q \lor r \lor p \lor t) \land$

$(p \lor s \lor \neg q) \land (p \lor s \lor p \lor t) \land$

$(q \lor s \lor \neg q) \land (q \lor s \lor p \lor t)$

Ejemplos

$((p \wedge q) \vee (r \wedge s)) \vee (\neg q \wedge (p \vee t))$

$\equiv \quad (p \vee r \vee \neg q) \wedge (p \vee r \vee p \vee t)) \wedge$

$(q \vee r \vee \neg q) \wedge (q \vee r \vee p \vee t) \wedge$

$(p \vee s \vee \neg q) \wedge (p \vee s \vee p \vee t) \wedge$

$(q \vee s \vee \neg q) \wedge (q \vee s \vee p \vee t)$

$\equiv \quad (p \vee r \vee \neg q) \wedge (p \vee r \vee t) \wedge (q \vee r \vee p \vee t) \wedge (p \vee s \vee \neg q)$
$\wedge (s \vee p \vee t) \wedge (q \vee s \vee p \vee t)$