Lógica

Semestre 2011-2

Lógica

"Está lloviendo."

Lógica

- En dos palabras:
  - Todos los estados en los que ha y puede llover.
- En vez de considerar todos los estados posibles, trabajamos con la **descripción** de un "conjunto de estados".
  - Abstracción de abstracción.
- Es un lenguaje en sí mismo.

Lógica

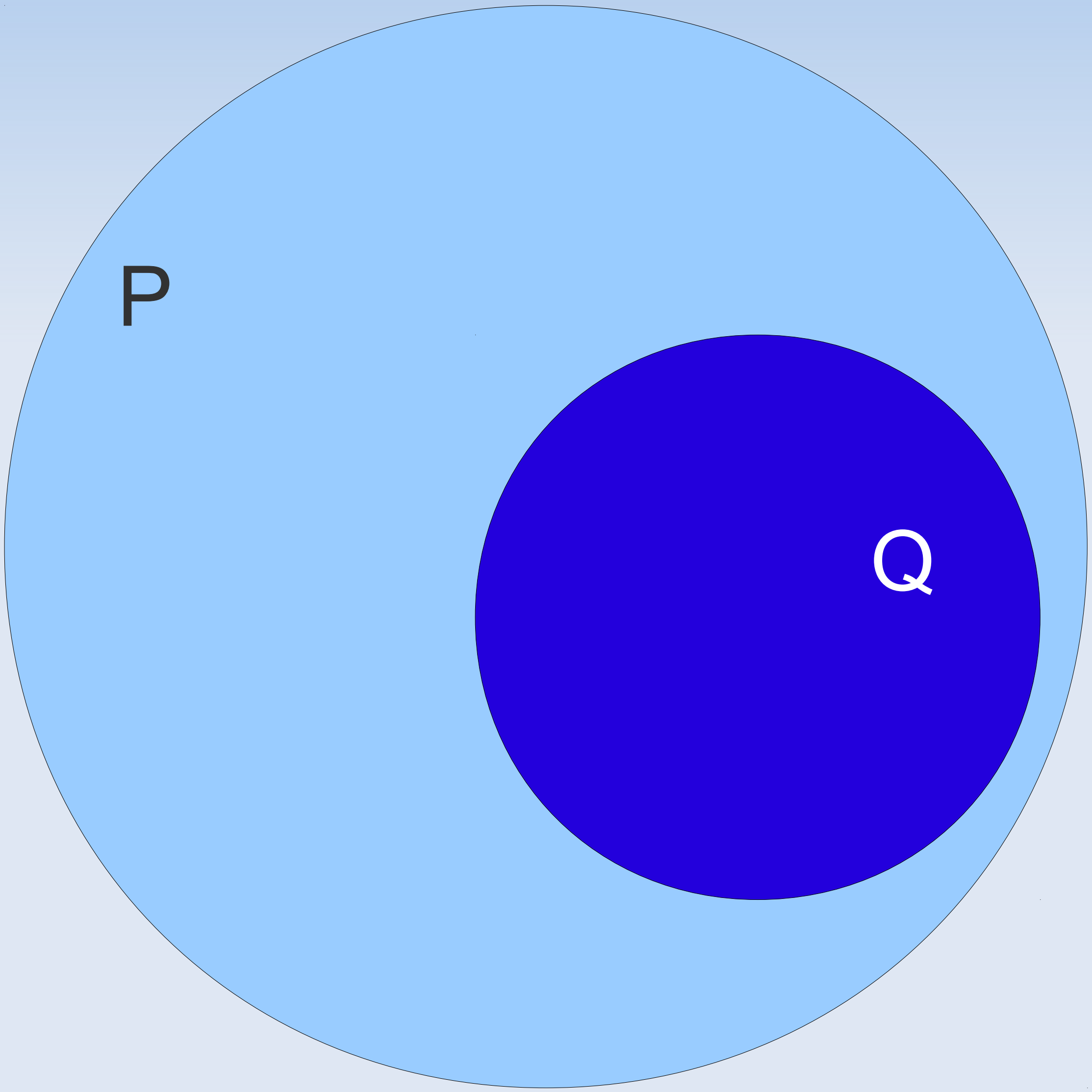
- Cada lógica es un lenguaje **formal**.
  - Alfabeto.
  - Sintáxis (gramática) => formar enunciados.
  - Semántica => significado a los enunciados.
- Lógica proposicional:
  - Significados posibles: true y false.

Pero Antes...

- Modelos:
  - x+y=4
    - true cuando (x=1,y=3),(x=-2,y=4),(x=2,y=2)
    - false cuando (x=2,y=-2),(x=1,y=7),(x=8,y=0)
- Vinculación:
  - $P \models Q$
    - Q es verdadero en todos los modelos en los que P es verdadero.
    - Los modelos que hacen true a Q están contenidos en el conjunto de modelos que hacen true a P.

Vinculación

$P \models Q$



Vinculación

- Vinculación se utiliza para derivar conclusiones (**inferencia lógica**).
- Ejemplo del Mundo del Wumpus.
  - El agente se pone en un laberinto, que se divide en secciones.
  - Hay un Wumpus, monstruo apestoso.
  - Abismos (PITs) con corrientes de viento.
  - El agente tiene una flecha para matar al Wumpus.

Mundo del Wumpus

	1	2	3	4
4	Stench	**GOLD**	Breeze	PIT
3	**WUMPUS**	Breeze	PIT	Breeze
2	Stench		Breeze	
1	*Start*	Breeze	PIT	Breeze

Mundo del Wumpus

	1	2	3	4
4	Stench	**GOLD**	Breeze	PIT
3	**WUMPUS**	Breeze	PIT	Breeze
2	Stench		Breeze	
1	*Start*	Breeze	PIT	Breeze

Cuadros (1,2) y (2,1) están OK.

Mundo del Wumpus

	1	2	3	4
4	Stench	**GOLD**	Breeze	PIT
3	**WUMPUS**	Breeze	PIT	Breeze
2	Stench		Breeze	
1	Start	*Breeze*	PIT	Breeze

Mundo del Wumpus

	1	2	3	4
4	Stench	**GOLD**	Breeze	PIT
3	**WUMPUS**	Breeze	PIT	Breeze
2	Stench		Breeze	
1	Start	*Breeze*	PIT	Breeze

Cuadros (2,2) y (1,3) están en duda.

Vinculación

- KB es el conjunto de modelos obtenidos con el conocimiento de las reglas y de lo observado, que dan un significado de true (conocimiento presente, o Knowledge-Base).
- Modelos posibles de KB:
  - (1,1) (1,2) y (2,1) Ok.
    - (2,2) OK. (1,3) PIT.
    - (2,2) PIT. (1,3) OK.
    - (2,2) PIT. (1,3) PIT.

Vinculación

- P = “No hay PIT en (1,2).”
  - (1,1) (1,2) y (2,1) Ok.
    - (2,2) OK. (1,3) PIT. P = true
    - (2,2) PIT. (1,3) OK. P = true
    - (2,2) PIT. (1,3) PIT. P = true
- Por lo tanto, KB ⊨ P

Vinculación

- Q = “No hay PIT en (2,2).”
  - (1,1) (1,2) y (2,1) Ok.
    - (2,2) OK. (1,3) PIT. Q = true
    - (2,2) PIT. (1,3) OK. Q = false
    - (2,2) PIT. (1,3) PIT. Q = false
- Por lo tanto, KB $\not\models$ Q

Inferencia

- Derivar conclusiones (“nueva” información) a partir del conocimiento presente.
  - O, saber si un enunciado nuevo (no presente en KB) *P* se puede vincular con KB.
- Denotación:
  - KB $\vdash_i$ P donde *i* es el algoritmo utilizado.
  - Dicho: P es derivado de KB por medio de *i*
- Algoritmo utilizado: **verificación de modelos**.

Inferencia

- Un algoritmo que sólo deriva enunciados vinculados del conocimiento presente se dice que “preserva la verdad” o que es **sensato** ('sound', en inglés).
  - No se inventa cosas.
- Un algoritmo que deriva *todos* los enunciados vinculados posibles se dice que es **completo.**

Lógica Proposicional

- También conocida como Booleana.
- La semántica más simple.
  - Significados posibles: true y false.
- De ésta, se puede extender a otras lógicas:
  - Lógica de Primer Orden
- Sirve como “introducción”.

Sintaxis

- Define el alfabeto y una gramática con la que podemos deducir si un **enunciado** es:
  - Bien formado.
  - Mal formado.
- Sintaxis de una ecuación matemática:
  - x+y=4 bien formado
  - x4y=+ mal formado

Sintaxis de Lógica Proposicional

- Bien formados:
  - True
  - False
  - P
  - Q
  - ¬P
  - P∧Q

- Proposición
  - Constante
  - Constante
  - Simbólica
  - Simbólica

    Atómicas
  - Compleja
  - Compleja

Sintaxis de Lógica Proposicional

- Conectores lógicos:
  -
  - $\neg$
  - $\wedge$
  - $\vee$
  - $\Rightarrow$
  - $\Leftrightarrow$
  -
  - $\equiv$

- En orden de precedencia
  - Negación
  - And
  - Or
  - Implicación
  - Bicondicional
  -
  - Equivalencia
  - (No es un conector lógico en sí)

Equivalencia

- $P \equiv Q$ significa “logicamente equivalentes).
- Se cumple si P y Q terminan siendo true con el mismo conjunto de modelos.
- En varios textos, lo equiparan con bidireccionalidad: $\Leftrightarrow$
- O con bi-vinculación:
  - $P \equiv Q$, si $P \models Q$ y $Q \models P$

Negación, And, y Or Bien Formados

- And, Or
  - $P \wedge Q$
  - $Q \vee P$
- Negación
  - $\neg P$
  - $\neg Q$

Implicación y Bicondicional Bien Formados

- Implicación (o condicional, o regla).
  - $P \land Q \Rightarrow Q \lor P$
    - $P \land Q$ es la **premisa**
    - $Q \lor P$ es la **conclusión**
- Bicondicional (si, y sólo si)
  - $\neg P \Leftrightarrow Q \lor P$

Semántica

P	*Q*	¬P	P ∧ Q	P ∨ Q	P ⇒ Q	P ⇔ Q
FALSE	***FALSE***	TRUE	FALSE	FALSE	TRUE	TRUE
FALSE	***TRUE***	TRUE	FALSE	TRUE	TRUE	FALSE
TRUE	***FALSE***	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	***TRUE***	FALSE	TRUE	TRUE	TRUE	TRUE

Tabla de Verdad

Implicación

- No hay una relación de consecuencia entre premisa y conclusión.
- Se describe como:
  - Si P es true, entonces puedo decir que Q es true; si no, no puedo decir nada.
  - Por lo que si P es falso, $P \Rightarrow Q$ es verdadero.
- Es más intuitivo considerarlo como:
  - $P \Rightarrow Q \quad \equiv \quad \neg P \vee Q$

Equivalencias Lógicas

- $P \wedge Q \equiv Q \wedge P$ [conmutatividad]
- $P \vee Q \equiv Q \vee P$
- $(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$ [associatividad]
- $(P \vee Q) \vee R \equiv P \vee (Q \vee R)$
- $\neg(\neg P) \equiv P$ [negación de negación]
- $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Equivalencias Lógicas

- $P \Rightarrow Q \equiv \neg Q \Rightarrow \neg P$
- $P \Rightarrow Q \equiv \neg P \lor Q$ [elim. implicación]
- $P \Leftrightarrow Q \equiv (P \Rightarrow Q) \land (Q \Rightarrow P)$ [elim. bidirec.]

  $(\neg P \lor Q) \land (P \lor \neg Q)$
- $\neg(P \land Q) \equiv \neg P \lor \neg Q$ [de Morgan]
- $\neg(P \lor Q) \equiv \neg P \land \neg Q$

Validez

- Cuando un enunciado es siempre true para todo modelo.
  - true
  - ¬false
  - $P \vee \neg P$
- ¿Para qué? Teorema de Deducción.
  - Para cualquier enunciado P y Q, $P \models Q$ si y solo si el enunciado $(P \Rightarrow Q)$ es válido.

Teorema de Deducción (En Español)

- Si tenemos una forma de deducir que un enunciado es válido, también podemos deducir si un enunciado es vinculado a otro.
- Dícese, si la validez de un enunciado **requiere** la validez de otro.
- Esto está relacionado con técnicas de *comprobación* ('proof', en inglés), que básicamente son técnicas para probar la validez de un enunciado.

Satisfactibilidad

- Cuando un enunciado es true con tan siquiera un modelo.
  - Se dice que el modelo *m* satisface al enunciado P.

No Satisfactibilidad

- Que un enunciado sea no satisfactible, significa que no hay un modelo que lo haga true.
- O, lo inverso de válido: es siempre false para todo modelo.
  - P es válido $\Leftrightarrow$ $\neg$P es no satisfactible.
- Por lo tanto:
  - $P \models Q \Leftrightarrow (P \wedge \neg Q)$ es no satisfactible
    - *Reductio ad absurdum*
    - Prueba por refutación o por contradicción
      - Asume Q como false, y encuentra contradicciones en axiomas conocidos como true en P.

Satisfactibilidad

- ¿Es satisfactible? ¿Es válido?
  - $Q \wedge \neg Q$
  - $P \wedge Q$
  - $P \vee Q \vee (P \Rightarrow Q)$
  - $P \Rightarrow Q \Rightarrow (\neg Q \Rightarrow \neg P)$ [contra-positivo]
  - $\neg(\neg P \vee Q) \wedge (\neg Q)$
  - $(R \vee P) \wedge \neg(R \vee Q) \wedge (\neg P \wedge Q)$

Satisfactibilidad

- ¿Es satisfactible?
  - $p_1 \wedge (p_1 \vee p_2) \wedge (p_1 \Rightarrow p_3) \wedge (p_1 \wedge p_3 \Rightarrow p_4)$
    $\wedge (p_5 \vee p_6) \wedge (p_5 \Rightarrow p_7) \wedge (\neg p_5 \vee p_8) \wedge (\neg p_7 \vee \neg p_8)$

¿Cómo lo Resolverían Eficientemente?

Siguiente Clase

- Repaso de sección 7.5:
  - Patrones de Razonamiento en Lógica Proposicional
- Enfoque mayor en 7.6:
  - Inferencia Proposicional Efectiva