

Algoritmos y Estructuras de Datos

Lógica de predicados

2024

Repaso - Lógica de Propocisional (PROP)

Lógica de Primer Orden (PRED)

Sintaxis de PRED

Semántica de PRED

Semántica trivaluada

Lógica proposicional (PROP) - sintaxis

- ▶ símbolos

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)$

- ▶ variables proposicionales (infinitas)

p, q, r, \dots

- ▶ fórmulas

- ▶ combinaciones **apropiadas** de símbolos y variables proposicionales
- ▶ Ejemplo de combinación inapropiada: $(\wedge p(($

Semántica clásica

- ▶ Consiste en asignarle **valores de verdad** a las fórmulas
- ▶ El conjunto de valores de verdad es

$$\{T, F\}$$

- ▶ Dos enfoques para darle semántica a las fórmulas de PROP
 1. Tablas de verdad
 2. Valuaciones
- ▶ Son equivalentes

Tablas de verdad

Conociendo el valor de las variables proposicionales de una fórmula, conocemos el valor de verdad de la fórmula

ϕ	$(\neg\phi)$
T	F
F	T

ϕ	ψ	$(\phi \wedge \psi)$
T	T	T
T	F	F
F	T	F
F	F	F

ϕ	ψ	$(\phi \vee \psi)$
T	T	T
T	F	T
F	T	T
F	F	F

ϕ	ψ	$(\phi \rightarrow \psi)$
T	T	T
T	F	F
F	T	T
F	F	T

ϕ	ψ	$(\phi \leftrightarrow \psi)$
T	T	T
T	F	F
F	T	F
F	F	T

Repaso - Lógica de Propocisional (PROP)

Lógica de Primer Orden (PRED)

Sintaxis de PRED

Semántica de PRED

Semántica trivaluada

Predicados

Considerar la frase

Todo estudiante es más joven que algún profesor

- ▶ En PROP lo representaríamos con una variable proposicional p
- ▶ Se pierde información sobre la estructura lógica de la frase
 - ▶ ser estudiante
 - ▶ ser profesor
 - ▶ ser más joven que

Individuos, predicados, variables y cuantificadores

Todo estudiante es más joven que algún profesor

$$(\forall x)(E(x) \rightarrow (\exists y(P(y) \wedge J(x, y))))$$

Individuos, predicados, variables y cuantificadores

Todo estudiante es más joven que algún profesor

$$(\forall x)(E(x) \rightarrow (\exists y(P(y) \wedge J(x, y))))$$

- ▶ Individuos (entidad distintiva e indivisible):
 - ▶ Estudiantes y profesores
 - ▶ Denotados por las variables x e y
- ▶ Predicados (predican sobre individuos):
 - ▶ $E(x)$: x es estudiante
 - ▶ $P(x)$: x es profesor
 - ▶ $J(x, y)$: x es más joven que y
- ▶ Cuantificadores
 - ▶ \forall : Para todo
 - ▶ \exists : Existe (para algún)

Funciones

Toda persona es menor que su madre biológica

- ▶ $G(x)$: x es persona
- ▶ $M(y, x)$: y es la madre de x

$$(\forall x)(\forall y)(G(x) \wedge M(y, x) \rightarrow J(x, y))$$

Funciones

Toda persona es menor que su madre biológica

- ▶ $G(x)$: x es persona
- ▶ $M(y, x)$: y es la madre de x

$$(\forall x)(\forall y)(G(x) \wedge M(y, x) \rightarrow J(x, y))$$

Funciones:

- ▶ Permiten representar objetos de manera más directa
- ▶ En lugar de escribir $M(y, x)$ podemos denotar a y con $m(x)$

$$(\forall x)(G(x) \rightarrow J(x, m(x)))$$

Otro ejemplo: *Andrea y Pedro tienen la misma abuela materna*

$$m(m(\text{andrea})) = m(m(\text{pedro}))$$

Términos y fórmulas

La lógica de predicados habla sobre dos clases de cosas:

- ▶ Individuos: Personas, números, colores, bolitas, grafos, árboles, etc.
 - ▶ Las expresiones que denotan individuos se llaman **términos**
 - ▶ Ej: las variables
 - ▶ Ej: Constantes como *andrea* y expresiones como $m(\textit{andrea})$
- ▶ Valores de verdad
 - ▶ Las expresiones que denotan valores de verdad se llaman **fórmulas**
 - ▶ Ej: $J(x, y)$

Cuantificadores

- ▶ $(\forall x) P(x)$: Fórmula lógica. Afirma que **todos** los elementos cumplen la propiedad P .
 - ▶ Se lee “Para todo x se cumple $P(x)$ ”
- ▶ $(\exists x) P(x)$: Fórmula lógica. Afirma que **al menos un** elemento cumple la propiedad P .
 - ▶ Se lee “Existe al menos un x que cumple $P(x)$ ”

Cuantificadores tipados

Syntax sugar para nuestro lenguaje de especificación (más en la próxima clase) vamos a aplicar cuantificadores a elementos de un tipo de datos.

- ▶ $(\forall x : T) P(x)$: Fórmula lógica. Afirma que **todos** los elementos de tipo T cumplen la propiedad P .
 - ▶ Se lee “Para todo x de tipo T se cumple $P(x)$ ”
- ▶ $(\exists x : T) P(x)$: Fórmula lógica. Afirma que **al menos un** elemento de tipo T cumple la propiedad P .
 - ▶ Se lee “Existe al menos un x de tipo T que cumple $P(x)$ ”

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se define si $y = 0$

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se define si $y \neq 0$

Rta (literal): “Dado un n mayor a 1, Para todos los n' enteros se cumple que si son mayores a 1 y menores a n , entonces n no es múltiplo de n' ”

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se define si $y \neq 0$

Rta (literal): “Dado un n mayor a 1, Para todos los n' enteros se cumple que si son mayores a 1 y menores a n , entonces n no es múltiplo de n' ”

Rta (corta): “Ningún entero positivo es múltiplo de otro entero positivo menor”

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se define si $y \neq 0$

Rta (literal): “Dado un n mayor a 1, Para todos los n' enteros se cumple que si son mayores a 1 y menores a n , entonces n no es múltiplo de n' ”

Rta (corta): “Ningún entero positivo es múltiplo de otro entero positivo menor”

- Todos los enteros entre 1 y 10 son pares:

$$(\forall n : \mathbb{Z})(1 \leq n \leq 10 \rightarrow n \bmod 2 = 0).$$

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se define si $y \neq 0$

Rta (literal): “Dado un n mayor a 1, Para todos los n' enteros se cumple que si son mayores a 1 y menores a n , entonces n no es múltiplo de n' ”

Rta (corta): “Ningún entero positivo es múltiplo de otro entero positivo menor”

- Todos los enteros entre 1 y 10 son pares:

$$(\forall n : \mathbb{Z})(1 \leq n \leq 10 \rightarrow n \bmod 2 = 0).$$

- Existe un entero entre 1 y 10 que es par:

$$(\exists n : \mathbb{Z})(1 \leq n \leq 10 \wedge n \bmod 2 = 0).$$

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se indefinir si $y = 0$

Rta (literal): “Dado un n mayor a 1, Para todos los n' enteros se cumple que si son mayores a 1 y menores a n , entonces n no es múltiplo de n' ”

Rta (corta): “Ningún entero positivo es múltiplo de otro entero positivo menor”

- Todos los enteros entre 1 y 10 son pares:

$$(\forall n : \mathbb{Z})(1 \leq n \leq 10 \rightarrow n \bmod 2 = 0).$$

- Existe un entero entre 1 y 10 que es par:

$$(\exists n : \mathbb{Z})(1 \leq n \leq 10 \wedge n \bmod 2 = 0).$$

- En general, si queremos decir que todos los enteros x que cumplen $P(x)$ también cumplen $Q(x)$, decimos:

$$(\forall x : \mathbb{Z})(P(x) \rightarrow Q(x)).$$

Ejemplos

- ¿qué dice el siguiente predicado?

$$n > 1 \wedge (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \bmod n' \neq 0)$$

Observación: $x \bmod y$ se define si $y \neq 0$

Rta (literal): “Dado un n mayor a 1, Para todos los n' enteros se cumple que si son mayores a 1 y menores a n , entonces n no es múltiplo de n' ”

Rta (corta): “Ningún entero positivo es múltiplo de otro entero positivo menor”

- Todos los enteros entre 1 y 10 son pares:

$$(\forall n : \mathbb{Z})(1 \leq n \leq 10 \rightarrow n \bmod 2 = 0).$$

- Existe un entero entre 1 y 10 que es par:

$$(\exists n : \mathbb{Z})(1 \leq n \leq 10 \wedge n \bmod 2 = 0).$$

- En general, si queremos decir que todos los enteros x que cumplen $P(x)$ también cumplen $Q(x)$, decimos:

$$(\forall x : \mathbb{Z})(P(x) \rightarrow Q(x)).$$

- Para decir que existe un entero que cumple $P(x)$ y que también cumple $Q(x)$, decimos:

$$(\exists x : \mathbb{Z})(P(x) \wedge Q(x)).$$

Algunas reglas de deducción

- La **negación** del cuantificador universal es el cuantificador existencial, y viceversa:

$$\neg(\forall n)P(n) \leftrightarrow (\exists n)\neg P(n).$$

$$\neg(\exists n)P(n) \leftrightarrow (\forall n)\neg P(n).$$

Algunas reglas de deducción

- ▶ La **negación** del cuantificador universal es el cuantificador existencial, y viceversa:

$$\neg(\forall n)P(n) \leftrightarrow (\exists n)\neg P(n).$$

$$\neg(\exists n)P(n) \leftrightarrow (\forall n)\neg P(n).$$

- ▶ El cuantificador universal **generaliza la conjunción**:

$$(\forall n : \mathbb{Z})P(n) \leftrightarrow P(1) \wedge P(2) \wedge P(3) \wedge \dots$$

Algunas reglas de deducción

- ▶ La **negación** del cuantificador universal es el cuantificador existencial, y viceversa:

$$\neg(\forall n)P(n) \leftrightarrow (\exists n)\neg P(n).$$

$$\neg(\exists n)P(n) \leftrightarrow (\forall n)\neg P(n).$$

- ▶ El cuantificador universal **generaliza la conjunción**:

$$(\forall n : \mathbb{Z})P(n) \leftrightarrow P(1) \wedge P(2) \wedge P(3) \wedge \dots$$

- ▶ El cuantificador universal **generaliza la disyunción**:

$$(\exists n : \mathbb{Z})P(n) \leftrightarrow P(1) \vee P(2) \vee P(3) \vee \dots$$

Repaso - Lógica de Propocisional (PROP)

Lógica de Primer Orden (PRED)

Sintaxis de PRED

Semántica de PRED

Semántica trivaluada

Lenguaje de primer orden

Un lenguaje de primer orden (LPO) \mathcal{L} consiste en:

1. Un conjunto numerable \mathcal{C} de constantes: c_0, c_1, \dots
2. Un conjunto \mathcal{F} de símbolos de función cada uno con aridad¹
 $n > 0$: f_0, f_1, \dots, f_k
3. Un conjunto \mathcal{P} de símbolos de predicado cada uno con aridad
 $n \geq 0$: $P_0, P_1, \dots, P_m, \doteq$.
 - El símbolo de predicado \doteq denotará igualdad.

¹Aridad=Número de argumentos que toman

Lenguaje de primer orden

Un lenguaje de primer orden (LPO) \mathcal{L} consiste en:

1. Un conjunto numerable \mathcal{C} de constantes: c_0, c_1, \dots
2. Un conjunto \mathcal{F} de símbolos de función cada uno con aridad¹
 $n > 0$: f_0, f_1, \dots, f_k
3. Un conjunto \mathcal{P} de símbolos de predicado cada uno con aridad
 $n \geq 0$: $P_0, P_1, \dots, P_m, \dot{=}$.
 - El símbolo de predicado $\dot{=}$ denotará igualdad.

Ejemplo: Lenguaje de primer orden para la aritmética

1. Constantes: 0
2. Símbolos de función: $S, +, *$
3. Símbolos de predicado: $\dot{=}, <$

¹Aridad=Número de argumentos que toman

Términos de primer orden

Sea $\mathcal{V} = \{x_0, x_1, \dots\}$ un conjunto numerable de variables y \mathcal{L} un LPO. El conjunto de \mathcal{L} -términos se define inductivamente como:

1. Toda constante de \mathcal{L} y toda variable es un \mathcal{L} -término
2. Si $t_1, \dots, t_n \in \mathcal{L}$ -términos y f es un símbolo de función de aridad n , entonces $f(t_1, \dots, t_n) \in \mathcal{L}$ -términos

En notación abreviada:

$$t ::= c \mid x \mid f(t, \dots, t)$$

Términos de primer orden

Sea $\mathcal{V} = \{x_0, x_1, \dots\}$ un conjunto numerable de variables y \mathcal{L} un LPO. El conjunto de \mathcal{L} -términos se define inductivamente como:

1. Toda constante de \mathcal{L} y toda variable es un \mathcal{L} -término
2. Si $t_1, \dots, t_n \in \mathcal{L}$ -términos y f es un símbolo de función de aridad n , entonces $f(t_1, \dots, t_n) \in \mathcal{L}$ -términos

En notación abreviada:

$$t ::= c \mid x \mid f(t, \dots, t)$$

Ejemplo: Aritmética (cont.)

- ▶ $S(0)$
- ▶ $+(S(0), S(S(0)))$
- ▶ $*(S(x_1), +(x_2, S(x_3)))$

Fórmulas atómicas

Sea \mathcal{V} un conjunto numerable de variables y \mathcal{L} un LPO. El conjunto de \mathcal{L} -fórmulas atómicas se define inductivamente como:

1. Todo símbolo de predicado de aridad 0 es una \mathcal{L} -fórmula atómica
2. Si $t_1, \dots, t_n \in \mathcal{L}$ -términos y P es un símbolo de predicado de aridad n , entonces $P(t_1, \dots, t_n)$ es una \mathcal{L} -fórmula atómica
Por ejemplo: Si $t_1, t_2 \in \mathcal{L}$ -términos, entonces $t_1 \doteq t_2$ es una \mathcal{L} -fórmula atómica

Fórmulas atómicas

Sea \mathcal{V} un conjunto numerable de variables y \mathcal{L} un LPO. El conjunto de \mathcal{L} -fórmulas atómicas se define inductivamente como:

1. Todo símbolo de predicado de aridad 0 es una \mathcal{L} -fórmula atómica
2. Si $t_1, \dots, t_n \in \mathcal{L}$ -términos y P es un símbolo de predicado de aridad n , entonces $P(t_1, \dots, t_n)$ es una \mathcal{L} -fórmula atómica
Por ejemplo: Si $t_1, t_2 \in \mathcal{L}$ -términos, entonces $t_1 \doteq t_2$ es una \mathcal{L} -fórmula atómica

Ejemplo: Aritmética (cont.)

- ▶ $< (0, S(0))$
- ▶ $< (x_1, +(S(0), x_2))$
- ▶ $\doteq (0, S(S(x_1)))$

Fórmulas de primer orden

Sea \mathcal{V} un conjunto numerable de variables y \mathcal{L} un LPO. El conjunto de \mathcal{L} -fórmulas se define inductivamente como:

1. Toda \mathcal{L} -fórmula atómica es una \mathcal{L} -fórmula
2. Si $\phi, \psi \in \mathcal{L}$ -fórmulas, entonces $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \leftrightarrow \psi)$ y $(\neg \phi)$ son \mathcal{L} -fórmulas
3. Para toda variable x_i y cualquier \mathcal{L} -fórmula ϕ , $(\forall x_i)(\phi)$ y $(\exists x_i)(\phi)$ son \mathcal{L} -fórmulas

Variables libres y ligadas

- ▶ Una ocurrencia de x en ϕ es ligada si x ocurre en un subtérmino de la forma $(\forall x)(\psi)$ o $(\exists x)(\psi)$.
- ▶ Una ocurrencia es libre si no es ligada.
- ▶ Una variable es libre (ligada) en una fórmula si ocurre libre (ligada) en la fórmula.

Ejemplo

$$P(x) \wedge (\forall x)(R(x, y) \rightarrow (\exists z)P(z))$$

- ▶ y es libre
- ▶ z es ligada
- ▶ x es libre y ligada

Variables libres y ligadas

- ▶ Usamos $FV(\phi)$ y $BV(\phi)$ para referirnos al conjunto de las variables libres y ligadas de ϕ , respectivamente (Free y Bounded)
- ▶ $FV(\phi)$ y $BV(\phi)$ se pueden definir por inducción estructural en ϕ

Ejemplo

Si $\phi = (\forall x)(R(x, y) \rightarrow P(x))$, entonces $FV(\phi) = \{y\}$ y $BV(\phi) = \{x\}$

- ▶ **Sentencia**: fórmula cerrada (i.e. sin variables libres)

Repaso - Lógica de Propocisional (PROP)

Lógica de Primer Orden (PRED)

Sintaxis de PRED

Semántica de PRED

Semántica trivaluada

Estructura de primer orden

Dado un lenguaje de primer orden \mathcal{L} , una estructura para \mathcal{L} , \mathcal{M} , es un par

$$\mathcal{M} = (M, I)$$

donde

- ▶ M (universo) es un conjunto no vacío
- ▶ I (función de interpretación) asigna funciones y predicados sobre M a símbolos de \mathcal{L} de la siguiente manera:
 1. Para toda constante c , $I(c) \in M$
 2. Para todo f de aridad $n > 0$, $I(f) : M^n \rightarrow M$
 3. Para todo predicado P de aridad $n \geq 0$ $I(P) \subseteq M^n$
 4. $I(\doteq)$ es la relación de identidad sobre M

Asignación

Asignación

Sea \mathcal{M} una estructura para \mathcal{L} . Una **asignación** es una función $s : \mathcal{V} \rightarrow M$

Dado s podemos definir \widehat{s} que se puede aplicar a términos para obtener el individuo del universo que denota

Extensión de una asignación a términos

$$\begin{aligned}\widehat{s}(x) &\stackrel{\text{def}}{=} s(x) \\ \widehat{s}(c) &\stackrel{\text{def}}{=} I(c) \\ \widehat{s}(f(t_1, \dots, t_n)) &\stackrel{\text{def}}{=} I(f)(\widehat{s}(t_1), \dots, \widehat{s}(t_n))\end{aligned}$$

Nota: a veces abusamos de la notación y escribimos simplemente s en lugar de \widehat{s}

Satisfactibilidad

Satisfactibilidad

La relación $s \models_{\mathcal{M}} \phi$ establece que la asignación s satisface la fórmula ϕ en la estructura \mathcal{M}

- ▶ Vamos a definir la relación $s \models_{\mathcal{M}} \phi$ de manera formal usando inducción estructural en ϕ
- ▶ Si s es una asignación y $a \in M$, usamos la notación $s[x \leftarrow a]$ para denotar la asignación que se comporta igual que s salvo en el elemento x , en cuyo caso retorna a

Satisfactibilidad

La relación $s \models_{\mathcal{M}} \phi$ se define inductivamente como:

$$s \models_{\mathcal{M}} P(t_1, \dots, t_n) \quad \text{sii} \quad (\widehat{s}(t_1), \dots, \widehat{s}(t_n)) \in I(P)$$

$$s \models_{\mathcal{M}} \neg \phi \quad \text{sii} \quad s \not\models_{\mathcal{M}} \phi$$

$$s \models_{\mathcal{M}} (\phi \wedge \psi) \quad \text{sii} \quad s \models_{\mathcal{M}} \phi \text{ y } s \models_{\mathcal{M}} \psi$$

$$s \models_{\mathcal{M}} (\phi \vee \psi) \quad \text{sii} \quad s \models_{\mathcal{M}} \phi \text{ o } s \models_{\mathcal{M}} \psi$$

$$s \models_{\mathcal{M}} (\phi \rightarrow \psi) \quad \text{sii} \quad s \not\models_{\mathcal{M}} \phi \text{ o } s \models_{\mathcal{M}} \psi$$

$$s \models_{\mathcal{M}} (\phi \leftrightarrow \psi) \quad \text{sii} \quad (s \models_{\mathcal{M}} \phi \text{ sii } s \models_{\mathcal{M}} \psi)$$

Satisfactibilidad

La relación $s \models_{\mathcal{M}} \phi$ se define inductivamente como:

$s \models_{\mathcal{M}} P(t_1, \dots, t_n)$	<i>sii</i>	$(\widehat{s}(t_1), \dots, \widehat{s}(t_n)) \in I(P)$
$s \models_{\mathcal{M}} \neg \phi$	<i>sii</i>	$s \not\models_{\mathcal{M}} \phi$
$s \models_{\mathcal{M}} (\phi \wedge \psi)$	<i>sii</i>	$s \models_{\mathcal{M}} \phi$ y $s \models_{\mathcal{M}} \psi$
$s \models_{\mathcal{M}} (\phi \vee \psi)$	<i>sii</i>	$s \models_{\mathcal{M}} \phi$ o $s \models_{\mathcal{M}} \psi$
$s \models_{\mathcal{M}} (\phi \rightarrow \psi)$	<i>sii</i>	$s \not\models_{\mathcal{M}} \phi$ o $s \models_{\mathcal{M}} \psi$
$s \models_{\mathcal{M}} (\phi \leftrightarrow \psi)$	<i>sii</i>	$(s \models_{\mathcal{M}} \phi \text{ sii } s \models_{\mathcal{M}} \psi)$
$s \models_{\mathcal{M}} (\forall x_i) \phi$	<i>sii</i>	$s[x_i \leftarrow a] \models_{\mathcal{M}} \phi$ para todo $a \in M$
$s \models_{\mathcal{M}} (\exists x_i) \phi$	<i>sii</i>	$s[x_i \leftarrow a] \models_{\mathcal{M}} \phi$ para algún $a \in M$

Validez

- ▶ Una fórmula ϕ es **satisfactible en \mathcal{M}** sii existe una asignación s tal que

$$s \models_{\mathcal{M}} \phi$$

- ▶ Una fórmula ϕ es **satisfactible** sii existe un \mathcal{M} tal que ϕ es satisfactible en \mathcal{M} . En caso contrario se dice que ϕ es **insatisfactible**.

- ▶ Una fórmula ϕ es **válida o verdadera en \mathcal{M}** sii

$$s \models_{\mathcal{M}} \phi, \text{ para toda asignación } s$$

- ▶ Una fórmula ϕ es **válida** sii es válida en toda estructura \mathcal{M} .
- ▶ **Nota:** ϕ es válida sii $\neg\phi$ es insatisfactible.

Ejemplos de fórmulas válidas

- ▶ $\phi \wedge \neg\phi$
- ▶ $(\forall x)\phi(x) \rightarrow (\exists x)\phi(x)$
- ▶ $(\forall x)\phi(x) \rightarrow \neg(\forall x)\neg\phi(x)$
- ▶ $(\forall x)\forall y.\phi(x, y) \rightarrow (\forall y)(\forall x)\phi(x, y)$
- ▶ $(\forall x)(\phi(x) \wedge \psi(x)) \rightarrow (\forall x)\phi(x) \wedge (\forall x)\psi(x)$
- ▶ y las que vimos antes...

Resumen de PRED

- ▶ Sintaxis
 - ▶ Lenguaje de primer orden \mathcal{L}
 - ▶ Términos sobre \mathcal{L} (denotan individuos)
 - ▶ Fórmulas sobre \mathcal{L} (denotan valores de verdad)
- ▶ Semántica
 - ▶ Estructuras: universo+interpretación de los símbolos de \mathcal{L}

Repaso - Lógica de Propocisional (PROP)

Lógica de Primer Orden (PRED)

Sintaxis de PRED

Semántica de PRED

Semántica trivaluada

Tablas de verdad

Conociendo el valor de las variables proposicionales de una fórmula, conocemos el valor de verdad de la fórmula

ϕ	$(\neg\phi)$
T	F
F	T

ϕ	ψ	$(\phi \wedge \psi)$
T	T	T
T	F	F
F	T	F
F	F	F

ϕ	ψ	$(\phi \vee \psi)$
T	T	T
T	F	T
F	T	T
F	F	F

ϕ	ψ	$(\phi \rightarrow \psi)$
T	T	T
T	F	F
F	T	T
F	F	T

ϕ	ψ	$(\phi \leftrightarrow \psi)$
T	T	T
T	F	F
F	T	F
F	F	T

Semántica trivaluada

- ▶ Supongamos que contamos con un símbolo relacional $==$ que nos permite comparar números reales
- ▶ ¿Valor de verdad de las siguientes fórmulas?

$$1 == 1 \qquad (1 + 1) == 2 \qquad 0,5 == 2/4$$

Semántica trivaluada

- ▶ Supongamos que contamos con un símbolo relacional $==$ que nos permite comparar números reales
- ▶ ¿Valor de verdad de las siguientes fórmulas?

$$1 == 1 \qquad (1 + 1) == 2 \qquad 0,5 == 2/4$$

- ▶ ¿Y esta?

$$1/0 == 2$$

Semántica trivaluada

Pasos para determinar si $e_1 == e_2$ es verdadero o falso

1. Obtener el número real r_1 denotado por e_1
2. Obtener el número real r_2 denotado por e_2
3. Comparar r_1 con r_2 para determinar si son iguales o no

Consideremos

$$1/0 == 2$$

Semántica trivaluada

Pasos para determinar si $e_1 == e_2$ es verdadero o falso

1. Obtener el número real r_1 denotado por e_1
2. Obtener el número real r_2 denotado por e_2
3. Comparar r_1 con r_2 para determinar si son iguales o no

Consideremos

$$1/0 == 2$$

- ▶ Trabado en paso 1
- ▶ Expresión $1/0$ no denota **ningún** número
- ▶ $1/0 == 2$ no es ni verdadera ni falsa porque no contamos con los números a comparar
- ▶ Le damos un valor especial: \perp (**indefinido**)

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	T
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	\perp
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	T
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	\perp
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	T
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	T
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	\perp
F	\perp	
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	T
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	\perp
F	\perp	T
\perp	T	
\perp	F	
\perp	\perp	

Semántica trivaluada (secuencial)

Se llama **secuencial** porque ...

- ▶ los términos se evalúan de izquierda a derecha,
- ▶ la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos \wedge_L (y-luego, o *conditional and*, o **cand**), \vee_L (o-luego o *conditional or*, o **cor**), y \rightarrow_L (implica-luego).

p	q	$(p \wedge_L q)$
T	T	T
T	F	F
F	T	F
F	F	F
T	\perp	\perp
F	\perp	F
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \vee_L q)$
T	T	T
T	F	T
F	T	T
F	F	F
T	\perp	T
F	\perp	\perp
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp

p	q	$(p \rightarrow_L q)$
T	T	T
T	F	F
F	T	T
F	F	T
T	\perp	\perp
F	\perp	T
\perp	T	\perp
\perp	F	\perp
\perp	\perp	\perp