Marcelo Alvarez

Resumen LTC PEP 1

Proposición ⊢ consecuencia
premisa

Conectores lógicos

$\sim$ $\neg$ no

$\vee$ o

$\wedge$ y

$\to$ si ..., entonces ... / implica

1. Deducción natural

premisa 1, premisa 2, ... ⊢ conclusión } Secuente

Un secuente es válido si encontramos una prueba para él.

Reglas

1. conjunción $\wedge$
   - ↳ Introducción: $A, B \Rightarrow A \wedge B$
   - ↳ Eliminación: $A \wedge B \Rightarrow A, B$

2. disyunción $\vee$
   - ↳ Introducción: $A \Rightarrow A \vee B$ $\quad B \Rightarrow A \wedge B$
   - ↳ Eliminación. $A \vee B, \boxed{\begin{array}{c}A\\ \vdots\\ X\end{array}}, \boxed{\begin{array}{c}B\\ \vdots\\ X\end{array}} \Rightarrow X$

3. implicancia $\to$
   - ↳ Introducción: $\boxed{\begin{array}{c}A\\ \vdots\\ B\end{array}} \Rightarrow A \to B$
   - ↳ Eliminación: $A \to B, A \Rightarrow B$
   - ↳ Modus Tollens: $A \to B, \sim B \Rightarrow \sim A$

4. negación $\sim$
   - ↳ Introducción: $\boxed{\begin{array}{c}A\\ \vdots\\ \bot\end{array}} \Rightarrow \sim A$
   - ↳ Eliminación: $A, \sim A \Rightarrow \bot$

2. Lógica proposicional (LP)

Inducción estructural

- Caso base: La estructura más simple
- Hipótesis inductiva: Suposición de que la propiedad es verdadera para todas las subestructuras más pequeñas.
- Paso inductivo: Si la propiedad es cierta para la estructura simple, entonces demostrar que también lo es para la estructura completa.

☆ Las estructuras de la inducción estructural operan recursivamente (descomponen una estructura en partes más pequeñas).

Inducción fuerte

- Caso base: Caso para el primer número natural (0 o 1, dependiendo de la propiedad)
- Paso inductivo: Suponer que la propiedad es verdadera para todos los números naturales menores o iguales a k.
- Caso inductivo: demostrar que la propiedad es válida para $k + 1$

Cláusula de Forma Canónica (CFC)

↳ Forma estandarizada de presentar una FBF.
Hay dos tipos:

- FNC: conjunción de disyunción (cláusulas) de literales
  Ejemplo: $(p \vee q) \wedge (r \vee \sim s) \wedge \ldots$
- FND: disyunción de conjunción (cláusulas duales) de literales
  Ejemplo: $(p \wedge q) \vee (r \wedge \sim s) \vee \ldots$

* Para saber si una fórmula es una FBF, hay que hacer un análisis sintáctico.

Fórmula satisfacible

↳ Una fórmula es satisfacible si existe alguna asignación a sus variables que haga que toda la fórmula sea verdadera.

- Tautología : La fórmula siempre es V (→ validez)
- Contradicción : La fórmula siempre es F
- Contingencia : Es posible que la fórmula sea V o F

Notación de satisfacibilidad:

$\sigma \models \varphi$

valuación ↵ ↳ fórmul

$\sigma \models \Sigma$

valuación ↵ ↳ conjunto de fórmulas

Notación de validez:

$\models \varphi$

Significa que una fórmula es válida para cualquier σ

Inferencia

↳ Cuando todas las premisas sean verdaderas, entonces la conclusión será válida.

$$\frac{\varphi_1, \dots, \varphi_n}{\psi}$$

Entonces, ψ es una consecuencia lógica de $\Sigma = \{\varphi_1, \dots, \varphi_n\}$

$\Sigma \models \psi$

Teorema de deducción

$\Sigma \models (\varphi \to \psi)$ si y sólo si $\Sigma \cup \{\varphi\} \models \psi$

(nos permite ↳

Demostración por Resolución (DPR)

Se basa en el principio de contradicción.

Si queremos demostrar que $\Sigma \models \psi$, entonces asumimos que $\sim\psi$ es verdadero, y buscamos una contradicción

Es decir, encontrar que $\Sigma \cup \{\sim\psi\}$ es inconsistente.

↳ Método de DPR

1. Transformar $\Sigma \cup \{\sim\psi\}$ a FNC. $(p \to q) \iff \sim p \vee q$
2. Aplicar la regla de resolución hasta que ya no se pueda más, o hasta que se encuentre una cláusula vacía.

Regla de resolución

↳ Si tengo una cláusula que tenga a un literal, y otra cláusula que tenga su negación, entonces puedo "inferir" una nueva cláusula que contenga los literales restantes.

3. Si encontramos una cláusula vacía, entonces hemos demostrado que $\Sigma \models \psi$. Si no, no se pudo demostrar.

* * * * * * * * * * * * *

3. Lógica de primer orden (LPO)

Recordemos que:

↳ LP asume que el mundo solo tiene hechos

↳ LPO asume que el mundo tiene objetos, predicados (o relaciones) y funciones

Cuantificadores lógicos

↳ $\forall$: para todo — universal

↳ $\exists$: existe a lo menos uno — existencial

Diferencia entre predicados y funciones.

↳ Los predicados son mapeos de un objeto a un V o F

↳ Las funciones son mapeos de muchos a un objeto

Vocabulario $L = \{\{P_1, P_2, \dots, P_m, \dots\}, \{f_1, f_2, \dots, f_m\}, \{c_1, c_2, \dots, c_m, \dots\}\}$

Variables → Se utilizan en predicados y funciones.

↳ Pueden estar ligadas a algún cuantificador o ser libres

Ejemplo : $P(x, y) \wedge \forall z Q(z)$

Una variable puede ser libre y ligada a la vez

Ejemplo: $P(x) + \forall x\, Q(x)$

↳ disyunción

↳ La variable es libre para P

↳ La variable es ligada para Q

Este ejemplo es contradictorio, → porque tengo dos condiciones con la misma letra así que estas "x" deben ser distintas.

Renombramiento: $P(v) + \forall x\, Q(x)$

Ejercicio propuesto de renombramiento:

$$\forall x\, (P(x) \to \exists x\, (Q(x) \to \forall x\, (R(x))))$$

$$\iff \forall x\, (P(x) \to \exists y\, (Q(y) \to \forall z\, (R(z))))$$

Términos → ejemplos:

x, y, z, $suma(x,s)$, $sucesor(sucesor(1))$

Una fórmula puede ser:

↳ Cerrada: no tiene variables libres

↳ Abierta: si tiene al menos una variable libre

y también se pueden clasificar en:

↳ Fórmulas atómicas → $P(t_1, \dots, t_k)$

↳ Fórmulas bien formadas → Incluyen a las atómicas

↳ A diferencia de LP, ahora se incluyen $(\forall x \varphi)$ y $(\exists x \varphi)$

Equivalencias

$\neg\forall x\, \varphi x$	equivale a	$\exists x\, \neg\varphi x$
$\neg\exists x\, \varphi x$	equivale a	$\forall x\, \neg\varphi x$
$\forall x\, \varphi x$	equivale a	$\neg\exists x\, \neg\varphi x$
$\exists x\, \varphi x$	equivale a	$\neg\forall x\, \neg\varphi x$
$\neg\forall x\, (\varphi x \to \psi x)$	equivale a	$\exists x\, \neg(\varphi x \to \psi x)$
	equivale a	$\exists x\, (\varphi x \wedge \neg\psi)$
$\forall x\, (\varphi x \to \neg\psi x)$	equivale a	$\neg\exists x\, \neg(\varphi x \to \neg\psi x)$
	equivale a	$\neg\exists x\, (\varphi x \wedge \psi x)$
$\forall x\, (\varphi x \wedge \psi x)$	equivale a	$\forall x\, \varphi x \wedge \forall x\, \psi x$
$\exists x\, (\varphi x \vee \psi x)$	equivale a	$\exists x\, \varphi x \vee \exists x\, \psi x$

Formas Normales

- FNC: conjunción de cláusulas
- FND: disyunción de cláusulas duales
- FNR (rectificada):
  - ninguna variable es libre y ligada a la vez
  - cada cuantificador actúa sobre una variable distinta
- FNP (Prenex):
  - los cuantificadores están solo al comienzo
  - ↳ Ejemplo: $\exists y \forall x\, [P(x) \wedge Q(y)]$
- FNS (Skolem):
  - FNP pero sin existenciales
  - ↳ Ejemplo: $\forall x \forall y \forall z\, [P(x) \wedge R(y) \vee Q(z))]$

Convertir a FNP

1. Rectificar

$$\forall x \varphi \equiv \forall y \varphi[x/y]$$
$$\exists x \varphi \equiv \exists y \varphi[x/y]$$

2. Eliminar bicondicionales y condicionales

$$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$$
$$\varphi \to \psi \equiv \sim\varphi \vee \psi$$

3. Interiorizar negaciones

$$\sim(\varphi \wedge \psi) \equiv (\sim\varphi \vee \sim\psi)$$
$$\sim(\varphi \wedge \psi) \equiv (\sim\varphi \wedge \sim\psi)$$
$$\sim\sim\varphi \equiv \varphi$$
$$\sim\forall x \varphi \equiv \exists x \sim\varphi$$
$$\sim\exists x \varphi \equiv \forall x \sim\varphi$$

4. Exteriorizar cuantificadores

Si x no es libre en ψ

Por la izquierda: $\forall x \varphi \wedge \psi \equiv \forall x(\varphi \wedge \psi)$

$\exists x \varphi \wedge \psi \equiv \exists x(\varphi \wedge \psi)$

· Por la derecha: $\psi \wedge \forall x \varphi \equiv \forall x(\psi \wedge \varphi)$

$\psi \wedge \exists x \varphi \equiv \exists x(\psi \wedge \varphi)$

y lo mismo con la disyunción

Convertir a FNS.

1. Transformar a FNP
2. Skolemización

↳ Los existenciales se eliminan de izquierda a derecha

↳ Si el existencial está al principio, entonces:

$\exists x \ldots \varphi(x) \Rightarrow \varphi(c)$ o $\varphi(h(c))$

cambiar x por constante, o por función de constante

↳ Si el existencial no está al principio, entonces:

$\forall x \forall y \exists z \forall w\, S(x,y,z,w) \Rightarrow \forall x \forall y \forall w\, S(x,y,f(x,y),w)$

función que dependa de las variables anteriores

Unificación

↳ Sustitución: Asignar un valor a una variable, reemplazándola en toda la fórmula.

↳ $t\{x/s\}$: reemplazar todas las ocurrencias de x por s en t

↳ $A\{x/s\}$: reemplazar todas las ocurrencias libres de x por s en A

- Dos expresiones son unificables si tienen un unificador
- t es una instancia común de t_1 y t_2 si existe una sustitución θ tal que $t = t_1\theta = t_2\theta$

☆ Para que t_1 y t_2 sean unificables:

- mismo símbolo de relación e.g. $P(x)$ y $P(f(y))$
- x y $f(x)$ no son unificables
- dos valores diferentes no pueden sustituir a x

Unificador Más General (UMG)

↳ Un unificador θ va a ser más general que un unificador τ $(\theta > \tau)$ si existe otro unificador λ tal que: $t_1\theta\lambda = t_2\tau$

↳ Un unificador θ es el UMG si $\theta > \tau$ para cualquier unificador τ aplicable.

En resumen, el UMG es aquel que unifica dos expresiones en la menor cantidad de sustituciones.

Algoritmo de Robinson

↳ Algoritmo de resolución, es decir, prueba que $\Sigma \models \varphi$

↳ Nos ayuda a encontrar, de ser posible, el UMG

Sean E y F dos términos que queremos unificar.
Consideramos inicialmente $E_0 = \sigma_0(E)$ y $F_0 = \sigma_0(F)$

$\sigma_0 = \{\}$

1. Si $E_k = F_k$ entonces E y F son unificables, y un UMG es $\sigma = \sigma_k \cdots \sigma_0$.
   El término E_k es el unificado.
   Si se cumple, el proceso termina aquí.

2. Si $E_k \neq F_k$, se busca el primer par de discordancia entre E_k y F_k.
   Sea éste D_k

3. Si D_k contiene una variable y un término, pasamos al siguiente paso. En otro caso, los términos no son unificables.

4. Si la variable aparece en el término, E y F no se unifican y termina el proceso.

5. Construimos una nueva sustitución que vincule la variable con el término de D_k
   Esta sustitución es σ_{k+1}.
   Y dos nuevos términos $E_{k+1} = \sigma_{k+1} E_k$
   $F_{k+1} = \sigma_{k+1} F_k$

El UMG será

$\sigma = \sigma_n \sigma_{n-1} \ldots \sigma_0$