Programación Lógica

Fundamentos teóricos:

Resolución

Instituto de Computación - Facultad de Ingeniería

Resumen

Definimos:

- Lenguaje de 1er orden
  - Alfabeto
  - Fórmulas Atómicas
  - Fórmulas del lenguaje (f.b.f)
- Variables libres y ligadas, fórmulas cerradas
- Cláusulas
- Cláusulas definidas

Resumen

Definimos:

- Interpretación
  - Dominio no vacío
  - Mapeos
    - Constantes
    - Funciones
    - Predicados
  - Valuación de variables
- Valor de verdad de una fórmula en una interpretación
- Modelo

Resumen

Definimos:

- Dado un conjunto de fórmulas S y una fórmula F
  - S es satisfactible
  - S es válido
  - S es insatisfactible
  - S es no válido
  - $S \models F$

Resumen

Propiedades

- S $\models$ F sii S U {¬F} es insatisfactible

Comentarios

- Un programa lógico P es un conjunto de cláusulas definidas.
- G, la negación de una consulta C, es una cláusula con todos los literales negativos.
- Deseamos probar P $\models$ C, para esto se prueba P U {G} es insatisfactible.

Objetivos

- Discutir como se prueba que un conjunto de fórmulas es insatisfactible.
- Definición de:
  - Sustitución
  - Unificación
  - Unificador más general
- Regla de inferencia: resolución

Sustituciones

Si tenemos variables libres podemos sustituirlas por términos cualesquiera.

Definimos:
Una sustitución θ es un conjunto finito de la forma {v1/t1,...,vn/tn}
en donde, para $1 \leq i \leq n$:

vi es una variable
ti es un término distinto de vi
vi $\neq$ vj si i $\neq$ j

Ejemplos
¿Son sustituciones?
{x/1, y/f(h,x)}
{x/f(y), z/z}
{x/1, y/f(h,x), x/2}

Sustituciones

Si tenemos variables libres podemos sustituirlas por términos cualesquiera.

Definimos:
Una sustitución θ es un conjunto finito de la forma {v1/t1,...,vn/tn}
en donde, para $1 \leq i \leq n$:

vi es una variable
ti es un término distinto de vi
$vi \neq vj$ si $i \neq j$

Ejemplos
{x/1, y/f(h,x)} es una sust.
{x/f(y), z/z} no es una sust.
{x/1, y/f(h,x), x/2} no es una sust.

Sustituciones

Si tenemos variables libres podemos sustituirlas por términos cualesquiera.

Definimos:
Una sustitución θ es un conjunto finito de la forma {v1/t1,...,vn/tn}
en donde, para $1 \leq i \leq n$:

vi es una variable
ti es un término distinto de vi
vi $\neq$ vj si i $\neq$ j

Prohibimos sust. vacías

Ejemplos
{x/1, y/f(h,x)} es una sust.
{x/f(y), z/z} no es una sust.
{x/1, y/f(h,x), x/2} no es una sust.

Sustituciones

Si tenemos variables libres podemos sustituirlas por términos cualesquiera.

Definimos:

Una sustitución θ es un conjunto finito de la forma {v1/t1,...,vn/tn}

en donde, para $1 \leq i \leq n$:

vi es una variable

ti es un término distinto de vi

$vi \neq vj$ si $i \neq j$

Prohibimos sust. vacías

Un solo valor para c/variable

Ejemplos

{x/1, y/f(h,x)} es una sust.

{x/f(y), z/z} no es una sust.

{x/1, y/f(h,x), x/2} no es una sust.

Instancia

Def. Instancia de una fórmula según una sustitución

Sea E una expresión (término o fórmula sin cuantificadores),

y θ = {v1/t1,...,vn/tn} una sustitución :

Eθ, la **instancia** de E por θ, es el resultado de la aplicación **simultánea** del reemplazo de cada variable vi por el término ti en E.

Ejemplo

Si E= p(x,y,f(b,x,z)) y θ={x/a, y/x, z/f(z)}

Eθ = ?

Instancia

Def. Instancia de una fórmula según una sustitución

Sea E una expresión (término o fórmula sin cuantificadores),

y θ = {v1/t1,...,vn/tn} una sustitución :

Eθ, la **instancia** de E por θ, es el resultado de la aplicación **simultánea** del reemplazo de cada variable vi por el término ti en E.

Ejemplo

Si E= p(x,y,f(b,x,z)) y θ={x/a, y/x, z/f(z)}

Eθ = p(a,x,f(b,a,f(z)))

Instancia

La noción de instancia se extiende a **conjuntos** de fórmulas:

Si S = {E1,...,En} es un conjunto de expresiones y θ una sustitución, Sθ, **instancia del conjunto** S por θ, es el conjunto {E1θ,...,Enθ}

Composición de sustituciones

Def. Composición de sustituciones

Sean σ = {v1/s1,...,vn/sn} y θ = {u1/t1,…,um/tm} dos sustituciones :
la **composición $\sigma \circ \theta$ de σ y θ** es la sustitución que resulta de

{v1/s1θ, ... , vn/snθ, u1/t1,…,um/tm }

luego de eliminar:

los pares vi/siθ t.q. vi=siθ y

los pares uj/tj tales que uj $\in$ {v1,...,vn}

Composición de sustituciones

Def. Composición de sustituciones

Sean σ = {v1/s1,...,vn/sn} y θ = {u1/t1,…,um/tm} dos sustituciones :
la **composición $\sigma \circ \theta$ de σ y θ** es la sustitución que resulta de

{v1/s1θ, ... , vn/snθ, u1/t1,…,um/tm }

luego de eliminar:

los pares vi/siθ t.q. vi=siθ y

Prohibimos sust. vacías

los pares uj/tj tales que uj $\in$ {v1,...,vn}

Composición de sustituciones

Def. Composición de sustituciones

Sean σ = {v1/s1,...,vn/sn} y θ = {u1/t1,…,um/tm} dos sustituciones :
la **composición $\sigma \circ \theta$ de σ y θ** es la sustitución que resulta de

{v1/s1θ, ... , vn/snθ, u1/t1,…,um/tm }

luego de eliminar:

los pares vi/siθ t.q. vi=siθ y

los pares uj/tj tales que uj $\in$ {v1,...,vn}

Prohibimos sust. vacías

Un solo valor para c/variable

Composición de sustituciones

Def. Composición de sustituciones

Sean σ = {v1/s1,...,vn/sn} y θ = {u1/t1,…,um/tm} dos sustituciones : la **composición $\sigma \circ \theta$ de σ y θ** es la sustitución que resulta de

{v1/s1θ, ... , vn/snθ, u1/t1,…,um/tm }

luego de eliminar:

los pares vi/siθ t.q. vi=siθ y

los pares uj/tj tales que uj $\in$ {v1,...,vn}

Prohibimos sust. vacías

Un solo valor para c/variable

θ1 = {x|a, y|f(z), z|y}
θ2 = {x|b, y|z, z|g(x)} sustituciones.
θ1 $\circ$ θ2 = ?

Composición de sustituciones

Propiedades de la composición de sustituciones.

$\theta \circ \varepsilon = \varepsilon \circ \theta = \theta$, siendo ε la sustitución definida por el conjunto vacío

$(E\sigma)\theta = E(\sigma \circ \theta)$

$(\gamma \circ \sigma) \circ \theta = \gamma \circ (\sigma \circ \theta)$ (Asociatividad de la composición).

Unificación

Def. Unificador

Sea S un conjunto de expresiones. Una sustitución θ es un unificador de S si $S\theta$ es un conjunto con un único elemento (*singleton*).

(Dicho de otro modo:

θ es unificador de S si para todos Ei, Ej $\in$ S,

$E_i\theta \equiv E_j\theta$)

Ejemplo:

S = {p(x, y, z), p(a,f(x),z)}

Unificación

Def. Unificador

Sea S un conjunto de expresiones. Una sustitución θ es un unificador de S si $S\theta$ es un conjunto con un único elemento (*singleton*).

(Dicho de otro modo:

θ es unificador de S si para todos Ei, Ej $\in$ S,

$Ei\theta \equiv Ej\theta$)

Ejemplo:

S = {p(x, y, z), p(a,f(x),z)}

θ = {x/a, y/f(a), z/b} es un unificador, $S\theta$ = {p(a,f(a),b)}

Unificación

Def. Unificador

Sea S un conjunto de expresiones. Una sustitución θ es un unificador de S si $S\theta$ es un conjunto con un único elemento (*singleton*).

(Dicho de otro modo:

θ es unificador de S si para todos Ei, Ej Є S,

$Ei\theta \equiv Ej\theta$)

Ejemplo:

S = {p(x, y, z), p(a,f(x),z)}

θ = {x/a, y/f(a), z/b} es un unificador, $S\theta$ = {p(a,f(a),b)}

Notar que σ = {x/a, y/f(a) } es también un unificador: $S\sigma$ = {p(a,f(a),z)}

Unificador más general

Def. Unificador más general (m.g.u.)

θ es un mgu para E1 y E2 si:

a. θ es un unificador para E1 y E2 y

b. si σ es un unificador para E1 y E2 , existe una sustitución γ t.q. $\theta \circ \gamma = \sigma$

El mgu de dos expresiones no tiene por que ser único, p.ej., tanto $\theta 1 = \{x/y\}$ como $\theta 2 = \{y/x\}$ son mgus para p(x) y p(y).

Unificador más general

Def. Unificador más general (m.g.u.)

θ es un mgu para E1 y E2 si:

a. θ es un unificador para E1 y E2 y

b. si σ es un unificador para E1 y E2 , existe una sustitución γ t.q. $\theta \circ \gamma = \sigma$

El mgu de dos expresiones no tiene por que ser único, p.ej., tanto $\theta 1=\{x/y\}$ como $\theta 2=\{y/x\}$ son mgus para p(x) y p(y).

Pero el mgu es único a menos de un renombre de variables.

Algoritmo de unificación

Algoritmo de unificación.

- ***Existe un algoritmo que determina el mgu de un conjunto de expresiones o reporta falla si las expresiones no son unificables.***

- **Veremos el algoritmo para el caso de dos expresiones.**

Algoritmo de unificación

Entrada: E1 y E2

Salida: μ, mgu de las expresiones, o *no unifican* si E1 y E2 no son unificables.

Inicialización: PUSH E1 = E2; $\mu = \varepsilon$;

Mientras el stack no se vacíe

POP X=Y

En caso de que

1. X es una variable que no ocurre en Y:

$\mu = \mu \circ \{X/Y\}$;

sustituir Y por X en el stack, o sea, stack = stack $\circ$ $\{X/Y\}$;

2. Y es una variable que no ocurre en X:

$\mu = \mu \circ \{Y/X\}$; stack = stack $\circ$ $\{Y/X\}$;

3. **X e Y son el mismo término:** skip

4. X es $f(X1,\ldots,Xn)$, Y es $f(Y1,\ldots,Yn)$:

PUSH $Xi = Yi;\ i = 1 : : : n$.

5. en otro caso: retornar *no unifican*.

Fin mientras

Retornar μ

Algoritmo de unificación

Entrada: E1 y E2

Salida: μ, mgu de las expresiones, o *no unifican* si E1 y E2 no son unificables.

Inicialización: PUSH E1 = E2; $\mu = \varepsilon$;

Mientras el stack no se vacíe

POP X=Y

En caso de que

1. X es una variable que no ocurre en Y:

$\mu = \mu \circ \{X/Y\}$;

sustituir Y por X en el stack, o sea, stack = stack $\circ$ $\{X/Y\}$;

2. Y es una variable que no ocurre en X:

$\mu = \mu \circ \{Y/X\}$; stack = stack $\circ$ $\{Y/X\}$;

3. X e Y son el mismo término: skip

4. X es $f(X1,\ldots,Xn)$, Y es $f(Y1,\ldots,Yn)$:

PUSH $Xi = Yi;\ i = 1 : : : n$.

5. en otro caso: retornar *no unifican*.

Fin mientras

Retornar μ

Unificar:

q(f(X,Y), X, h(a))

q(f(b,Z), W, h(Z))

Resolución

Juan estudia francés o inglés.

Juan no estudia inglés.

Se deduce:

Juan estudia francés.

$\{P \vee Q \,,\, \neg P\} \;\Rightarrow\; Q$

$\{\neg \exists x\, P(x) \;\vee Q(a) \,,\, Q(b) \vee \exists x\, P(x)\} \;\Rightarrow Q(a) \vee Q(b)$

Resolución

El esquema de derivación que se utiliza se llama resolución

$$\frac{\alpha \lor \beta, \quad \gamma \lor \neg\beta}{\alpha \lor \gamma}$$

- La propuesta de esta regla es de Robinson (1965) y está asociada a sistemas de demostración automática.

Resolución

El esquema de derivación que se utiliza se llama resolución

$$\frac{\alpha \lor \beta, \quad \gamma \lor \neg\beta}{\alpha \lor \gamma}$$

- La regla de resolución es **correcta**.

- Resolución es una regla de derivación, se aplica sintácticamente.
- La correctitud hace corresponder consecuencia lógica a derivación.

Resolución

El esquema de derivación que se utiliza se llama resolución

$$\frac{\alpha \lor \beta, \quad \gamma \lor \neg\beta}{\alpha \lor \gamma} \qquad (\alpha \lor \beta, \gamma \lor \neg\beta) \vdash \alpha \lor \gamma$$

- La regla de resolución es **correcta**.
- En efecto, si ambas premisas son verdaderas en una interpretación I, una al menos de las fórmulas α o γ deben ser verdaderas en I. Por lo tanto, $\alpha \lor \gamma$ es verdadera en I. Se cumple entonces que $\alpha \lor \gamma$ es consecuencia lógica de las premisas.

$(\alpha \lor \beta, \gamma \lor \neg\beta) \models \alpha \lor \gamma$

Resolución para cláusulas proposicionales

En un paso de resolución entre dos cláusulas se elimina **un par de literales complementarios** entre dos cláusulas. La cláusula resultante (la conclusión) es la disyunción de los resultantes literales.

Sean

C1: L1 v L2 v ... v Ln

C2: M1 v M2 v ... v Mh

y Li = ¬Mj

C3: L1 v L2 v ... v Li-1 v Li+1 v ... v Ln v M1 v … v Mj-1 v Mj+1 v … v Mh

Se obtiene por resolución de C1 y C2.

Decimos que C3 es una resolvente de C1 y C2.

Procedimientos de resolución

Objetivo

Probar que un conjunto de cláusulas definidas es insatisfactible.

Método

Aplicación sistemática de resolución entre pares de cláusulas hasta llegar a la cláusula vacía.

Procedimientos de resolución

Objetivo

Probar que un conjunto de cláusulas definidas es insatisfactible.

Método

Aplicación sistemática de resolución entre pares de cláusulas hasta llegar a la cláusula vacía.

Por qué deseamos llegar a la cláusula vacía?

Procedimientos de resolución

Objetivo

Probar que un conjunto de cláusulas definidas es insatisfactible.

Método

Aplicación sistemática de resolución entre pares de cláusulas hasta llegar a la cláusula vacía.

Por qué deseamos llegar a la cláusula vacía?

La cláusula vacía no tiene modelos.

Procedimientos de resolución

Aplicación sistemática de resolución

- Saturación por niveles
- Resolución lineal
- Resolución con cláusula de entrada

Ejs.-

$S = \{\{p,q\}, \{\neg p,q\}, \{p, \neg q\}, \{\neg p, \neg q\}\}$

Resumen

- Cálculo de sustituciones
- Unificación, m.g.u
- Algoritmo de unificación
- Introducción a resolución
- Resolución para lógica proposicional

Próxima

- Resolución para cláusulas de 1er orden
- Resolución lineal
- Resolución con cláusulas de entrada
- Resolución SLD
- Arbol SLD
- **Prolog puro !!**