

Lezioni del 17/3 e 18/3 su DCL

Modelli di Herbrandt e procedura
bottom up.
Universo e Mondo chiuso

- Dobbiamo estendere la semantica alla nuova sintassi
- Interpretazione $I = (D, \Phi, \pi)$
 - D dominio di individui
 - Φ : simbolo di costante $c \mapsto$ elemento $c^{\Phi} \in D$
 - simbolo di funzione n-aria
 - $f \mapsto$ (funzione $f^{\Phi} : D^n \rightarrow D$)
 - π : simbolo di predicato n-ario
 - $p \mapsto$ (predicato $p^{\pi} : D^n \rightarrow \{v, f\}$)

A) Interpretazioni di Herbrandt

- **A) Universo di Herbrandt** $U(KB)$ di una KB: insieme dei termini ground ottenuti con costanti e funzioni di KB
- **B) Base di Herbrandt** $B(KB)$: insieme degli atomi ground ottenuti con costanti, funzioni e predicati di KB
 - **Interpretazione di Herbrandt** $H = (U(KB), \Phi_H, I)$:
 - $c^{\Phi_H} = c$
 - $f^{\Phi_H}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$
 - $I \subseteq B(KB)$ con il significato:
 - $p(t_1, \dots, t_m) = v$ sse $p(t_1, \dots, t_m) \in I$

- Siccome il dominio di ogni interpretazione di Herbrandt è sempre $U(KB)$ e le costanti e funzioni sono interpretate sempre allo stesso modo, **un'interpretazione di Herbrandt è univocamente rappresentata dall'insieme degli atomi ground veri in essa.**

Semantica interna/esterna

La definizione di $f^{\Phi_H} : U(KB)^n \rightarrow U(KB)$ data da

$$f^{\Phi_H}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$$

interpreta **l'universo $U(KB)$ come insieme di dati strutturati e f come costruttore di dati**

Base di Herbrandt $B(KB)$: rappresenta l'insieme di tutte le proprietà esprimibili con i predicati di KB, nell'universo di Herbrandt

Interpretazione di Herbrandt: ciò che è vero in una specifica situazione, rappresenta il significato interno

Esempio: Una **scimmia** si trova in una stanza. Appesa al soffitto c'è una **banana**. Nella stanza ci sono tre **cubi** che possono **essere sovrapposti**. Se la scimmia è intelligente, sovrappone i cubi, vi **sale sopra** e **prende** la banana

Introduciamo la segnatura per una KB corrispondente alle nostre astrazioni:

cubo1, cubo2, cubo3, banana, scimmia: costanti

su operazione (funzione) di sovrapposizione di oggetti

prende, sale_sopra: predicati binari

Non avendo tipi, l'universo è un calderone:

- cubo1, cubo2, cubo3, banana, scimmia
- su(cubo1,cubo2), su(cubo1,banana),
- su(cubo1,su(cubo1,cubo2)), su(cubo1,su(cubo2,cubo3)), ..

Esercizio: quali dei termini precedenti rappresentano costruzioni reali nel mondo che vogliamo rappresentare?

Vediamo alcuni atomi della base di Herbrandt

- sale_sopra(banana,scimmia), prende(cubo,scimmia),
- sale_sopra(scimmia,su(cubo1,cubo2)), prende(scimmia,banana)

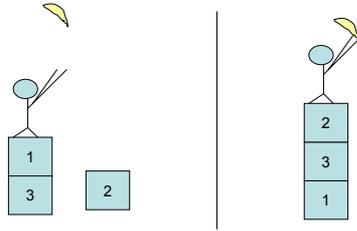
Esercizio: quali dei precedenti atomi mettereste in un'interpretazione di Herbrandt che rappresenti una situazione sensata?

Esercizio: A) Scrivere le interpretazioni di Herbrandt che corrispondono ai due riquadri della vignetta

B) Se interpretassi **su** come predicato binario e non come funzione, riuscirei egualmente a rappresentare la vignetta?

NO: spiegare

SI: come? Confrontare con A)



Esempio con tipi dati infiniti

- $U(\text{sum}) = \{0, s(0), s(s(0)), \dots\}$
- $B(\text{sum}) = \{ \text{sum}(s^n(0), s^m(0), s^h(0)) \mid n,m,h \geq 0 \}$
- $H_+ = \{ \text{sum}(s^n(0), s^m(0), s^{n+m}(0)) \mid n,m \geq 0 \}$
- $H_- = \{ \text{sum}(s^n(0), s^m(0), s^{n^*m}(0)) \mid n,m \geq 0 \}$
- ecc.
- Naturalmente il programma sum calcola H_+ e non H_- .
- Il modello minimo si genera, al solito, con la procedura bottom up

B) La procedura Bottom-up

- È sempre quella, calcola le conseguenze immediate usando le istanze ground delle calusole.
 - si possono avere infinite istanze.
- C conseguenza immediata di H e di $A \leftarrow B_1 \wedge \dots \wedge B_n$ sse esiste un'istanza ground $A\sigma \leftarrow B_1\sigma \wedge \dots \wedge B_n\sigma$ tale che $C = A\sigma$

Esempio

$\text{sum}(X,0,X).$
 $\text{sum}(X,s(Y),s(Z)) \leftarrow \text{sum}(X,Y,Z).$

$T_{\text{sum}}(\{\}) = I_1 = \{ \text{sum}(s^n(0),0,s^n(0)) \mid n \geq 0 \}$
 $T_{\text{sum}}(I_1) = I_2 = I_1 \cup \{ \text{sum}(s^n(0), s(0), s^{n+1}(0)) \mid n \geq 0 \}$

$\text{sum}^* = \{ \text{sum}(s^n(0), s^m(0), s^{n+m}(0)) \mid n,m \geq 0 \}$

Valgono i soliti teoremi su KB^* già visti per DLP e DataLog

- **Validità e completezza**
- **Minimo punto fisso**
- **Modello minimo.**
- Non ha più senso parlare di complessità circa il calcolo di KB^* : se il minimo modello di Herbrandt è infinito, while "termina" solo dopo ω passi (ω è il primo ordinale infinito).

C) Sull'universo chiuso

- La segnatura di una KB è
 - l'insieme dei simboli di funzione e costante usati in KB;
 - genera l'**universo di Herbrandt** U_{KB}
 - l'insieme dei simboli di predicato usati in KB;
 - genera la **base di Herbrandt** B_{KB}
- Possiamo fare diverse ipotesi circa l'universo:

- **Ipotesi del dominio chiuso:**
 - tutti gli individui sono rappresentati da termini di U_{KB}
- **Ipotesi del dominio aperto:**
 - non conosciamo a priori l'insieme di tutti gli individui; alcuni non hanno denotazione in U_{KB}
- **Ipotesi del nome unico:**
 - termini ground distinti rappresentano individui diversi.

Esercizio

- Si consideri il programma Vocali:
 - vocale(a).
 - vocale(e).
 - vocale(i).
 - vocale(o).
 - vocale(u).
 - consonante(d).
- Qual è la segnatura di Vocali?
- Qual è l'universo di Herbrandt?
- Pensando all'alfabeto, ha senso l'ipotesi del nome unico?
- E quella del dominio chiuso?

Esercizio critico

- Si vogliono rappresentare i record di naturali.
- Qual è la segnatura?
- Definire il programma per il predicato $leq(A,B)$ che confronta due naturali secondo l'usuale ordinamento e due record componente per componente.
- L'universo di Herbrandt contiene ospiti indesiderati?

D) Sul mondo chiuso CWA (Closed World Assumption) e definizioni induttive

- A livello di tipi di dati, conviene collegare CWA e definizioni induttive.
- Un predicato $p(X_1, \dots, X_n)$ è definito induttivamente da:
 - Base: $p(X_1, \dots, X_n) = \text{vero}$ se **Base**(X_1, \dots, X_n)
 - Passo: se $p(X_1, \dots, X_n) = \text{vero}$ e **Passo**($X_1, \dots, X_n, Y_1, \dots, Y_n$), allora $p(Y_1, \dots, Y_n) = \text{vero}$
 - Minimalità: se $p(X_1, \dots, X_n)$ non è derivabile da Base e Passo, allora è falso
- L'ipotesi di minimalità corrisponde al mondo chiuso.

In DCL

$p(X_1, \dots, X_n) :- \text{Base}(X_1, \dots, X_n).$
 $p(Y_1, \dots, Y_n) :- p(X_1, \dots, X_n), \text{Passo}(X_1, \dots, X_n, Y_1, \dots, Y_n).$

- Si tratta di programmare **Base**(X_1, \dots, X_n) e **Passo**($X_1, \dots, X_n, Y_1, \dots, Y_n$).
- L'ipotesi di minimalità corrisponde ad assumere come falso ciò che non vale nel modello di Herbrandt minimo

Esempio: definizione di un tipo

Tipo nat

Base: $X:\text{nat}$ se $X = 0$

Passo: se $X:\text{nat}$ e $Y=s(X)$, allora $Y:\text{nat}$

Minimalità: nient'altro è di tipo nat

Siccome **Base**(X) è $X=0$ e **Passo**(X, Y) è $Y=s(X)$, conviene

Base: $0:\text{nat}$

Passo: se $X:\text{nat}$, allora $s(X):\text{nat}$

In DCL:

$\text{is}(0,\text{nat})$.

$\text{is}(s(X),\text{nat}) :- \text{is}(X,\text{nat})$.

Esempio: definizione di un predicato

Predicato $X \leq Y$

Base: $0 \leq Y$

Passo: se $X \leq Y$, allora $s(X) \leq s(Y)$

Minimalità: in nessun altro caso, $X \leq Y$

In DCL:

$\text{leq}(0,X)$.

$\text{leq}(s(X),s(Y)) :- \text{leq}(X,Y)$.

Esempio: relazioni funzionali

Definiamo $\text{half}(X,Y) = \text{def } x \text{ div } 2$.

Base: $\text{half}(X,0)$ se $X=0$ or $X=s(0)$.

Passo: se $\text{half}(X,Y)$, allora $\text{half}(s(s(X)),s(Y))$

Minimalità: al solito

In DCL

$\text{half}(0,0)$.

$\text{half}(s(0),0)$.

$\text{half}(s(s(X)),s(Y)) :- \text{half}(X,Y)$.

ESERCIZI PROPOSTI

- Datalog:
 - Dare una KB datalog per rappresentare un labirinto, in cui si abbiano stanze e corridoi; un corridoio può collegare due stanze, una se è cieco. Una stanza si può aprire su più corridoi. Definire la raggiungibilità in al più 5 passi.
- DCL
 - Rappresentare gli stati di un sistema produttore-coda-consumatore, in modo poter definire il predicato di transizione di di stato possibili con una coda con capacità 2. Sarebbe possibile in DataLog?
 - Dare somma e prodotto di naturali
 - Provare a definire differenza e divisione come operazioni inverse (spiegazioni in aula). Programmare i due casi in Prolog. Cosa succede?
 - Rappresentare il problema visto a lezione, con scimmia, banana, cubi. Problema: come definire le pile di cubi?