Lezione 18

Estensioni di DCL

Provvisoria

1. Datalog, DCL e altri frammenti di FOL

- **Datalog** è un frammento della logica del primo ordine con identità (**FOL**, First Order Logic).
- Fra **Datalog** e **FOL** vi sono diversi sistemi intermedi;
  - **DCL** è uno di questi
  - Aggiungendo *NAF* (Negation as Finite Failure) nel corpo delle clausole, si ottiene **NCL** (Normal Clause Logic, sostanzialmente = Prolog)
  - Spesso i sistemi intermedi sono rafforzati con assunzioni extra-logiche: *NAF* (Negation as Failure) e *CWA* (Closed World Assumption) sono fra queste e verranno discusse in queste lezioni

1.1. FOL con identità

- Vediamo sommariamente FOL, assumendo per semplicità, l'ipotesi del mondo chiuso (trattazione generale sui testi di logica).
- In FOL le formule atomiche sono definite esattamente come gli atomi in DCL
  - Si definiscono i termini (vedere lezioni precedenti)
  - Atomiche:
    - p(t1,..,tn) dove p è un simbolo di predicato n-ario e t1,..,tn sono termini
    - t1 = t2 dove t1, t2 sono termini
- formula: atomica
  (¬ formula), (formula ∧ formula), (formula ∨ formula),
  (formula → formula),
  (∀x formula), (∃x formula)

Formule aperte e chiuse

- In presenza dei quantificatori ∃, ∀, la definizione di istanza ground va rimpiazzata con quella di ***formula chiusa***
  - in (Qx A), A è detta ***scopo del quantificatore*** Qx
  - Una occorrenza di una variabile x si dice ***vincolata*** in una formula se è compresa nello scopo di un quantificatore Qx;
  - Si dice ***libera*** altrimenti
  - Le variabili libere di una formula sono quelle che hanno almeno una occorrenza libera
  - ***Una formula si dice chiusa sse non ha variabili libere***

Esempio

∀**x**(amichevole(**x**) → ∃**y** amico(**x**,**y**)) ∧ cibo(**y**)

Vincolate da ∀**x**

Vincolata da ∃**y**

libera

Le istanze ground si ottengono sostituendo solo le occorrenze libere delle variabili libere

Esempio

(∀**x** (amichevole(**x**) → ∃**y** amico(**x**,**y**)) ∧ cibo(**y**)){**y**/pizza}
=
∀**x**(amichevole(**x**) → ∃**y** amico(**x**,**y**)) ∧ cibo(pizza)

Interpretazioni e verità

- Le interpretazioni interpretano simboli di funzione e di predicato, esattamente come in DCL:
  - $I = (D, \Phi, \pi)$
  - $\pi (=)$ è, in ogni I, l'identità in D
  - come in DCL, si ha che in I
    - ogni termine ground t denota un valore del dominio D
    - ogni atomo ground assume un valore di verità (**v**, **f**)
- La verità in I di una formula chiusa F, indicata con

  I |= F

  si definisce induttivamente come segue

- Base: F atomica

  I |= F sse il valore di verità di F in I è **v**
- Passo: a seconda dei casi

I |= $\neg$H	sse	non I |= H
I |= H $\wedge$ K	sse	I |= H e I |= K
I |= H $\vee$ K	sse	I |= H o I |= K
I |= H $\rightarrow$ K	sse	(non I |= H) o I |= K
I |= $\forall$x H	sse	per ogni {x/t} ground, I |= H {x/t}
I |= $\exists$x H	sse	esiste {x/t} ground tale che I |= H {x/t}

Esempio

- Consideriamo il dominio rappresentato dalle costanti

  ugo, mario, gigi, pizza, pasta
- e l'interpretazione di Herbrand

  I = {amichevole(ugo), amico(ugo,mario), cibo(pizza), cibo(pasta)}

I |= $\forall$**x**(amichevole(**x**) $\rightarrow$ $\exists$**y** amico(**x**,**y**)) $\wedge$ cibo(**pizza**):
- I |= $\forall$**x**(amichevole(**x**) $\rightarrow$ $\exists$**y** amico(**x**,**y**)):
  - I |= amichevole(**ugo**) $\rightarrow$ $\exists$**y** amico(**ugo**,**y**)):
    - I |= $\exists$ **y** amico(**ugo**,**y**)):
      - I |= amico(**ugo**,**mario**) perché amico(ugo,mario) è **v**
  - I |= amichevole(**gigi**) $\rightarrow$ $\exists$**y** amico(**gigi**,**y**)):
    - non I |= amichevole(**gigi**) perché amichevole(gigi) è **f**
  - I |= amichevole(**mario**) $\rightarrow$ $\exists$**y** amico(**mario**,**y**)):
    - non I |= amichevole(**mario**) perché amichevole(mario) è **f**
  - I |= amichevole(**pizza**) $\rightarrow$ $\exists$**y** amico(**pizza**,**y**)):
    - non I |= amichevole(**pizza**) perché amichevole(pizza) è **f**
  - I |= amichevole(**pasta**) $\rightarrow$ $\exists$**y** amico(**pizza**,**y**)):
    - non I |= amichevole(**pasta**) perché amichevole(pizza) è **f**
- I |= cibo(**pizza**) perché cibo(pizza) è **v**

- Chiaramente il procedimento di analisi della verità appena visto non è realizzabile computazionalmente per domini infiniti. La definizione di verità costituisce un procedimento "ideale".

- **ESERCIZIO:** dare una interpretazione che falsifichi la formula esaminata nel precedente esempio, semplicemente aggiungendo un fatto vero all'interpretazione di Herbrandt considerata

KB in FOL

- In FOL una base di conoscenza è assiomatizzata mediante un ***insieme di formule chiuse*** KB
- KB è solitamente detta una ***teoria assiomatica del primo ordine***
- I modelli si definiscono nel solito modo:

  I modello di KB, scritto I |= KB

  sse I |= A per ogni A in KB

Esercizio

- Considerare la KB
  KBamici = {($\forall$x amichevole(x) $\rightarrow$ $\exists$y amico(x,y)),
  $\exists$x amichevole(x),
  $\forall$x $\forall$y (amico(x,y) $\rightarrow$ amico(y,x)) }

- Si consideri il dominio di costanti {mario, ugo}
- Elencare tutti i modelli di Herbrandt di KBamici con tale dominio.
- Fra i modelli trovati, esiste il modello minimo?

Validità e completezza

- Si ha la solita definizione di ***conseguenza logica*** (per F chiusa):
  - KB |= F sse, per ogni modello M di KB, M |= F
- Si hanno vari ***calcoli*** (sistemi di inferenza) ***validi e completi:***
  - KB |- F sse KB |= F

1.2 Frammenti di FOL

- Vediamo come aumentare DCL gradualmente, ottenendo frammenti di FOL via via più ampi.
- FOL resta il quadro di riferimento.
- Cominciamo considerando l'identità.

2. Identità e disuguaglianza

- L'unificazione corrisponde all'assunzione del nome unico:
  - termini ground distinti denotano individui distinti
- L'unificazione lavora anche su termini aperti e possiamo usarla nel corpo delle clausole
- Considerando i modelli di Herbrandt, l'unificazione porta ad un risultato di validità e completezza anche usando = nel corpo delle clausole:
  - validità: se σ è una sostituzione di risposta di un goal A, allora KB |= Aσ
  - completezza: se KB |= $\exists$ x A , allora vi è una sostituzione di risposta

Esempio

```
sum(X,0,Z) :- Z=X.
sum(X,s(Y),Z) :- sum(X,Y,A), Z=s(A).
```


Disuguaglianza

- Le cose cambiano se usiamo la disuguaglianza con termini aperti.

```
freq(mario,cl(1,a)).
freq(gigi,cl(1,a)).
freq(ugo,cl(2,a)).
freq(lea,cl(3,a)).

amico(X,Y) :- X ≠ Y, freq(X,A), freq(Y,A).
```

Soluzione

- Ritardare le disuguaglianze aperte.

- NOTA: /== prolog non è la disuguaglianza:
  T1 /== T2 vale se T1 e T2 non coincidono sintatticamente; ad es. 2 /== X è vero;
  per usarlo correttamente come disuguaglianza, T1 e T2 devono essere ground

2.1. Estensioni: CLP, Eqlog

- Talora l'assunzione del nome unico è problematica. Ad esempio, fatti del tipo
  2 + 2 = 4
  sono esclusi dal nome unico.
- Vi sono estensioni di Prolog che consentono l'uso (eventualmente lim itato a determinati operatori) di operatori quale la somma, che portano a non avere più nome unico
  - CLP (Constraint Logic Programming)
  - Eqlog (implementazioni?)

3. Negazione

- Vediamo prima i legami fra negazione e assunzione del mondo chiuso.
- Poi consideriamo la negazione come fallimento finito e il completamento di Clark