↔←→¬∨∧⊆≠∪∈∉

Lezione 3

LP e “DataLog” proposizionale

- LP (Logica Proposizionale)
  - Abbiamo dato sintassi e semantica delle ***formule*** di LP
  - Oggi vediamo
    - KB: rappresentazione di conoscenze in LP
    - SR: sistema di ragionamento per LP

3.1. LP: teorie (KB) e modelli

- **Una teoria** KB è un insieme di formule
  - le teorie fanno parte della sintassi: sono la sintassi in cui esprimere una base di conoscenza KB
  - le formula di KB sono dette assiomi della teoria
- Un **modello** di KB è un’interpretazione I tale che
  I |= F per ogni F appartenente a KB
  - I modelli fanno parte della semantica: corrispondono alle realtà rappresentate dalla teoria

Notazione: I(A) = **v** *si scrive* I |= A

Esercizio

- KB1 = {nuvolo, ventoso, in_barca}
- KB2 = { nuvolo, ventoso,
  ventoso ∧ ¬temporale → in_barca }
- I = {nuvolo,ventoso,in_barca} è modello di KB1 e di KB2?
- Dare una I che non sia modello di KB1 ma sia modello di KB2.
- KB1 ha I come (unico) ***modello minimo***. (EC)
  Modello minimo: ***assunzione mondo chiuso***
- Se l’agente deve simulare un patito della barca a vela, quale delle due KB è più “intelligente”?

3.1.1. Conseguenze logiche e dimostrazioni

- Una formula F è **conseguenza logica** di KB:
  KB |= F sse, per ogni modello I di KB, I |= F
  - la nozione di conseguenza logica fa parte della semantica; riguarda le proprietà vere in tutte le situazioni modellate da KB
- Un calcolo è un insieme di regole di dimostrazione. Una ***dimostrazione*** è una successione di passaggi certificabili in base alle ***regole del calcolo***, che porta dagli ***assiomi*** di KB ad un ***teorema*** F:
  KB |- F (da KB si dimostra F) sse esiste una dimostrazione del teorema F che usa gli assiomi di KB;

3.1.2. Calcoli = sistemi di ragionamento per LP

TEOREMI (ripasso da logica)

- **Validità:** KB |- F ➔ KB |= F
- **Completezza:** KB |= F ➔ KB |- F
- Per i teoremi di validità e completezza, un ***calcolo*** per la logica proposizionale classica è un **SR** (Sistema di Ragionamento) per LP:
  - **Domande:** F ? (cioè KB |= F?)
  - **Risposte:** SI / NO
- FATTO: in LP KB |= F è decidibile con complessità esponenziale (se KB è finita).

Dove siamo: concetti, notazioni e teoremi da conoscere

- ***Interpretazione*** $I : \boldsymbol{P} \rightarrow \{\mathbf{v}, \mathbf{f}\}$ o $I \subseteq \boldsymbol{P}$
- ***Verità*** di una formula F in I: I |= F
- Interpretazione I ***modello*** di una teoria KB : I |= KB
- Modelli (e interpretazioni) di Herbrandt
- Formula F ***conseguenza logica*** di una teoria KB : KB |= F
- ***Dimostrazione:*** successione di passaggi certificabili in base alle ***regole di un calcolo***, che porta dagli ***assiomi*** di KB ad un ***teorema*** F:
  KB |- F sse esiste una dimostrazione di F
- ***Validità:*** KB |- F ==> KB |= F
- ***Completezza:*** KB |= F ==> KB |- F
- ***Domande:*** F?
- ***Risposte:*** SI/NO

3.2. DataLog Proposizionale (DLP)

- Sottosistema di LP, primo passo verso DataLog
- Il calcolo per DLP verrà esteso a DataLog e Prolog
  - Sinonimi di "calcolo":
    - Sistema di ragionamento (SR)
    - "motore inferenziale"
    - …

La sintassi di DLP

- *Atomi:* atomiche (simboli proposizionali ***P***)
- *Corpo* (Body): congiunzione di *atomi* B1 ∧ .. ∧ Bn
  Es. sereno ∧ ventoso
- *Clausola Definita:*
  - *Fatto:* *Atomo*
    Es: soleggiato.
  - *Regola:* *Atomo* ← *Corpo*
    Es. in_mare ← sereno ∧ ventoso.
    Atomo: TESTA (Head) della clausola.

- *Base Conoscenza* (o *Programma*):
  insieme di *Clausole Definite*

ES:
soleggiato ← sereno.
soleggiato ← nubi_passeggere.
fresco ← nubi_passeggere.
fresco ← ventoso.
benessere ← fresco ∧ soleggiato.
ventoso.
nubi_passeggere.

3.2.1. SR: sistema di ragionamento

Regola di ragionamento basata su Modus Ponens

$$\frac{B1, \ldots, Bn \qquad A \leftarrow B1 \wedge .. \wedge Bn}{A}$$

Validità della regola: in ogni interpretazione I, se le premesse (sopra la riga) sono vere in I, allora la conseguenza è vera in I

IL CALCOLO: LE SUE PROVE

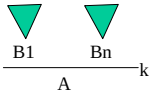
KB = 1. Clausola$_1$. ….. n. Clausola$_n$.

Base

$$\frac{}{A}\ i$$

i. A fatto di KB

Passo

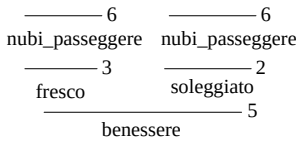


k. A ← B1∧.. ∧Bn
regola di KB

ricorsivamente, prove

Esempio di prova

1. soleggiato ← sereno.
2. soleggiato ← nubi_passeggere.
3. fresco ← nubi_passeggere.
4. fresco ← ventoso
5. benessere ← fresco ∧ soleggiato.
6. nubi_passeggere.



Validità e completezza

- DEF. KB |- A sse esiste un albero di prova con radice A.
- Validità: KB |- A ➔ KB |= A
  - Dim. Per induzione sulla profondità della prova (lavagna)
- Completezza: KB |= A ➔ KB |- A
  - Dim. Vedremo.

3.2.2. Motore inferenziale. Procedura non deterministica Top Down (o Goal-Oriented o backward)

Ingresso: **KB** + ***Goal: atomo A da dimostrare***.

- Albero Parziale AP := A?
- Fino a che AP contiene un F? e nessun F?FAIL:
  - ***scegli*** un F?
  - Se esiste una clausola con testa F
    prendi k. F ← B1∧.. ∧Bn
    e applica la regola k all'indietro (vedi esempio)
    altrimenti
    marca F? con FAIL (cioè F?FAIL)

Non determinismo

- ***Scegli:*** don't-care
  - Fair selection rule (evitare starvation)
- ***Prendi***
  - Non determinismo don't-know (un oracolo può fare le scelte giuste)
  - Procedendo deterministicamente: necessità di BACKTRAKING (alberi di ricerca)
- THEO. La ricerca di una prova ha "non-determinisicamente successo" se e solo se la prova esiste