Università degli Studi di Bologna

Corso di Laurea in Informatica
Esercitazione scritta di LOGICA PER L'INFORMATICA
Esempio per l'anno accademico 2018-2019

1 (1 punto). Dare la sintassi per le formule della logica proposizionale.
Esercizio di riscaldamento: chiede sempre o la sintassi della logica proposizionale o quella del prim'ordine.

Soluzione:

$$F ::= \bot \mid \top \mid A \mid B \mid \ldots \mid \neg F \mid F \wedge F \mid F \vee F \mid F \Rightarrow F$$

2 (5 punti). Considerare la seguente sintassi delle liste di X: $L ::= [] \mid X :: L$ dove :: è associativo a destra. Scrivere la funzione ricorsiva f che su una lista L di numeri che restituisca la lista LL di liste **non vuote** di numeri tale che:

(a) Se $LL = L_1 :: \ldots :: L_n :: []$ allora $L = L_1 @ \ldots @ L_n$ dove @ è la funzione che concatena due liste. In altre parole, LL è fatta da frammenti di L nell'ordine in cui occorono in L.

(b) ogni L_i è una lista monotona, non crescente o non decrescente, di numeri

(c) L_1 è monotona **non decrescente**

(d) le liste L_i sono di **lunghezza massimale**, ovvero L_i e L_{i+1} non possono essere concatenate per ottenere un'unica lista monotona.

Esempi:

$$f(1 :: 3 :: 7 :: 6 :: 6 :: 4 :: 8 :: 10 :: []) = (1 :: 3 :: 7 :: []) :: (6 :: 6 :: 4 :: []) :: (8 :: 10 :: []) :: []$$

$$f(7 :: 4 :: 2 :: 9 :: []) = (7 :: []) :: (4 :: 2 :: []) :: (9 :: []) :: []$$

È possibile utilizzare funzioni ausiliarie su liste, da definirsi usando la ricorsione strutturale, funzioni ausiliarie su numeri (da non definirsi) e/o passare parametri ausiliari alle funzioni.

Esercizio sulla ricorsione strutturale: vedi esercizi 2 dati nell'a.a. 2015-2016, 2016-2017, 2017-2018.

Possibile soluzione:

- Primo problema:

  Specifica: data L restituire la lista di liste LL come specificato nell'esercizio.

  Funzione che lo risolve: $f(L)$

  Codice:

```
f([]) = []
f(N::L) =
  if L ≠ [] and N <= first(L) then
     add_to_first_list(N,f(L))
   else
     (N::[])::g(L)
```

  Nota: nel caso $N :: L$ la chiamata ricorsiva strutturale $f(L)$ risolve il problema su L. Mi resta da capire cosa farne di N. Se N può fare parte della prima sequenza monotona di L, lo aggiungo ad essa. Altrimenti N deve formare una sequenza singoletto da solo e deve essere seguito dalla lista di sequennze che inizia con una monotona non crescente. Risolvo tale problema con la funzione g. La f e la g sono definite per ricorsione mutua: ognuna richiama l'altra solo su sottotermini immediati dell'input, rispettando i dettami della ricorsione strutturale. Un'altra soluzione possibile è che la $f()$ e la $g()$ siano la stessa funzione che prende in input un parametro aggiuntivo (un booleano) per determinare quale dei due problemi risolvono (prima sequenza non decrescente vs non crescente).

- Secondo problema:

  Specifica: data L restituire la lista di liste LL come specificato nell'esercizio ma con la differenza che la lista L_1 deve essere monotona non crescente.

  Funzione che lo risolve: $g(L)$

  Codice:

```
g([]) = []
g(N::L) =
  if L ≠ [] and N >= first(L) then
     add_to_first_list(N,g(L))
   else
     (N::[])::f(L)
```

- Terzo problema:

Specifica: dato un numero N e una lista di liste di numeri LL, aggiungere N in testa alla prima lista di LL. Se LL non contiene liste, il comportamento non è specificato (= fate un poco come volete). Esempio: `add_to_first_list(3,(1::4::0::[])::(5::[])::[]) = (3::1::4::0::[])::(5::[])::[]`

Funzione che lo risolve: `add_to_first_list(N,LL)`

Codice:

```
add_to_first_list(N,[]) = []
add_to_first_list(N,L::LL) = (N::L)::LL
```

- Quarto problema:

  Specifica: data una lista non vuota di numeri, restituirne il primo elemento. Se la lista è vuota il comportamento non è specificato.

  Funzione che lo risolve: `first(L)`

  Codice:

```
first([]) = 666999
first(N::L) = N
```

Nota: a me interessa solamente il codice. Tutto il resto ve l'ho scritto per farvi vedere come si ragiona. Quando pensate al codice pensate sempre in termini di problemi, ovvero come decomporre problemi in problemi più semplici, uno dei quali è sempre come ottenere la soluzione del problema dalla soluzione dei sottoproblemi.

3 (3 punti). Dimostrare in teoria assiomatica degli insiemi che

$$\forall A \forall B \forall C \forall D (A \subseteq B \wedge C \subseteq D \Rightarrow A \cup C \subseteq B \cup D)$$

Vedi esempi di esercizi analoghi dati nello scorso anno accademico. L'importante è che i vari passaggi corrispondano a uno o più passi di deduzione naturale della logica intuizionista.
Soluzione: Siano A, B, C, D insiemi. Supponiamo $A \subseteq B \wedge C \subseteq D$ da cui $A \subseteq B$ (H1) e $C \subseteq D$ (H2). H1 è equivalente a $\forall X(X \in A \Rightarrow X \in B)$ mentre H2 è equivalente a $\forall X(X \in C \Rightarrow X \in D)$. Dobbiamo dimostrare $A \cup C \subseteq C \cup D$ che è equivalente a $\forall X(X \in A \cup C \Rightarrow X \in B \cup D)$. Sia X un insieme. Assumiamo $X \in A \cup C$. Per l'assioma dell'unione binaria ($\forall A \forall B \forall X(X \in A \cup B \iff X \in A \vee X \in B)$) si ha $X \in A \vee X \in C$. Procediamo per casi.

- Caso $X \in A$. Per H1 si ha $X \in B$ e quindi $X \in B \vee X \in D$. Per l'assioma dell'unione binaria si ha $X \in B \cup D$.

- Caso $X \in C$. Per HC si ha $X \in D$ e quindi $X \in B \vee X \in D$. Per l'assioma dell'unione binaria si ha $X \in B \cup D$.

Qed.

4 (1 punto). Scrivere le due leggi di assorbimento della logica proposizionale classica.
Gli esercizi 4-5-6 possono richiedere 1) enunciati e definizioni visti a lezione; 2) dimostrazioni di semplici teoremi visti a lezione o comunque ottenibili senza andare per induzione; 3) frammenti di dimostrazioni viste a lezione; 4) applicazioni banali delle definizioni e enunciati visti a lezione, per esempio fornendo esempi/controesempi per certe definizioni.

5 (1 punto). Enunciare il teorema di correttezza per la logica proposizionale classica.
Soluzione: per ogni formula F e per ogni insieme di formule Γ si ha che se $\Gamma \vdash F$ (usando le regole della deduzione naturale classica) allora $\Gamma \Vdash F$ (in logica proposizionale classica)

6 (1 punto). Dimostrare che $\{\Rightarrow, \neg\}$ è un insieme funzionalmente completo di connettivi per la logica propoosizionale classica.
Soluzione: riduco l'insieme funzionalmente completo $\{\bot, \top, \wedge, \vee\neg\}$ ad esso.

- $\neg$ appartiene all'insieme $\{\Rightarrow, \neg\}$
- $\top \equiv A \Rightarrow A$
- $\top \equiv \neg\bot$
- $F \vee G \equiv \neg F \Rightarrow G$
- $F \wedge G \equiv \neg(\neg F \vee \neg G)$

Nota: nella sequenza ho sempre usato solo connettivi dell'insieme $\{\Rightarrow, \neg\}$ o connettivi che ho già ridotto prima a tale insieme (es. uso il $\vee$ per codificare il $\wedge$).

7 (6 punti). Considerare le formule della logica proposizionale ristrette al frammento $F ::= A \mid B \mid \ldots \mid \top \mid F \wedge F$. Dimostrare, per induzione strutturale su F_2, che se $F_1 \Vdash F_2$ allora $FV(F_2) \subseteq FV(F_1)$ (dove $FV(F)$ sono le variabili proposizionali che occorrono in F).

Suggerimento: è possibile dimostrare lemmi ulteriori sempre per induzione strutturale.

Esercizio che richiede di fornire una dimostrazione per induzione strutturale (NON vista a lezione). Vedi esercizi 7 dell'a.a. 2015-2016, 2016-2017, 2017-2018 e esercizi analoghi nei precedenti a.a.
Soluzione:
Teorema: $\forall F_1, F_2, \quad (F_1 \Vdash F_2) \Rightarrow FV(F_2) \subseteq FV(F_1)$.
Dimostrazione: sia F_1 una formula fissata. Procedo per induzione strutturale su F_2.

- Caso A: dobbiamo dimostrare $(F_1 \Vdash A) \Rightarrow FV(A) \subseteq FV(F_1)$ che è equivalente a $(F_1 \Vdash A) \Rightarrow \{A\} \subseteq FV(F_1)$. Vedi lemma successivo.
- Casi B, C, ...: analoghi al caso A.
- Caso $\top$: dobbiamo dimostrare $(F_1 \Vdash \top) \Rightarrow FV(\top) \subseteq FV(F_1)$ che è equivalente a $(F_1 \Vdash \top) \Rightarrow \emptyset \subseteq FV(F_1)$. Vero in quanto la conclusione è vera per l'assioma dell'insieme vuoto.
- Caso $G_1 \wedge G_2$.
  Per ipotesi induttiva per $i \in \{1, 2\}$ si ha $(F_1 \Vdash G_i) \Rightarrow FV(G_i) \subseteq FV(F_1)$.
  Dobbiamo dimostrare $(F_1 \Vdash G_1 \wedge G_2) \Rightarrow FV(G_1 \wedge G_2) \subseteq FV(F_1)$ che è equivalente a $(F_1 \Vdash G_1 \wedge G_2) \Rightarrow FV(G_1) \cup FV(G_2) \subseteq FV(F_1)$.
  Assumiamo $F_1 \Vdash G_1 \wedge G_2$, ovvero in ogni mondo v t.g. $v \Vdash F_1$ si ha $[\![G_1 \wedge G_2]\!]^v = 1$ ovvero $[\![G_1]\!]^v = [\![G_2]\!]^v = 1$. Quindi $F_1 \Vdash G_1$ e $F_1 \Vdash G_2$ e, per ipotesi induttiva, $FV(G_1) \subseteq FV(F_1)$ e $FV(G_2) \subseteq FV(F_1)$. Quindi $FV(G_1) \cup FV(G_2) \subseteq FV(F_1) \cup FV(F_1) = FV(F_1)$. Qed.

Lemma: per ogni F_1, $(F_1 \Vdash A) \Rightarrow A \in FV(F_1)$.
Dimostrazione: per induzione strutturale su F_1.

- Caso A: dimostro $(A \Vdash A) \Rightarrow A \in FV(A)$ che è equivalente a $(A \Vdash A) \Rightarrow A \in \{A\}$. Vero perchè la conclusione lo è.
- Caso B: dimostro $(B \Vdash A) \Rightarrow A \in FV(B)$. Vero perchè la premessa è falsa come dimostra l'esempio del mondo in cui B è vera e A è falsa.
- Casi C, D, ...: analoghi al precedente.
- Caso $\top$: dimostro $(\top \Vdash A) \Rightarrow A \in FV(B)$. Vero perchè la premessa è falsa come dimostra l'esempio del mondo in cui A è falso.

- Caso $G_1 \wedge G_2$: per ipotesi induttiva per $i \in \{1,2\}$ si ha $(G_i \Vdash A) \Rightarrow A \in FV(G_i)$.
  Dobbiamo dimostrare $(G_1 \wedge G_2 \Vdash A) \Rightarrow A \in FV(G_1 \wedge G_2)$ che è equivalente a $(G_1 \wedge G_2 \Vdash A) \Rightarrow A \in FV(G_1) \cup FV(G_2)$.
  Supponiamo $G_1 \wedge G_2 \Vdash A$, che è equivalente al fatto che in ogni mondo v tale che $[\![G_1 \wedge G_2]\!]^v = 1$ (che è equivalente a $[\![G_1]\!]^v = [\![G_2]\!]^v = 1$) si ha $v \Vdash A$ (H). Dimostriamo che per un qualche $i \in \{1,2\}$ si ha $G_i \Vdash A$, per poi concludere per ipotesi induttiva che $A \in FV(G_i) \subseteq FV(G_1) \cup FV(G_2)$. Sia v un mondo. Dobbiamo dimostrare che per un qualche i si ha che se $[\![G_i]\!]^v = 1$ allora $v \Vdash A$. Procediamo per casi su $(v \Vdash A) \vee (v \nVdash A)$. Nel primo caso non c'è altro da dimostrare perchè la conclusione è vera. Nel secondo, dimostriamo la premessa falsa. Da (H) la cui conclusione è falsa segue che la premessa $[\![G_1]\!]^v = [\![G_2]\!]^v = 1$ è falsa, ovvero c'è almeno un $i \in \{1,2\}$ tale che $[\![G_i]\!]^v \neq 1$. Qed.

8 (7 punti). Si consideri il seguente ragionamento:
Se Luca ha ricevuto il regalo allora glielo ha portato Babbo Natale o la Befana. Babbo Natale non porta mai il carbone. La Befana porta il carbone ai bimbi cattivi. Quindi se Luca ha ricevuto il regalo e il regalo era carbone allora Luca è stato cattivo.

Verificare la correttezza del ragionamento utilizzando la deduzione naturale per la logica proposizionale. Preferire una prova intuizionista se possibile.
Esercizio sulla formalizzazione e la deduzione naturale per la logica proposizionale, classica o intuizionista. Vedi esercizi analoghi per tutti gli a.a. passati.
Questo è un caso di formalizzazione in logica proposizionale di un ragionamento che è formulato usando (implicitamente) quantificatori. Inoltre vi è l'uso di "glielo" che complica il tutto.
R = Luca ha ricevuto il regalo
N = Babbo Natale ha portato il regalo a Luca
B = la Befana ha portato il regalo a Luca
C = il regalo è del carbone
X = Luca è cattivo

$$R \Rightarrow N \vee B, \quad N \Rightarrow \neg C, \quad B \wedge C \Rightarrow X \quad \vdash \quad R \wedge C \Rightarrow X$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{N \Rightarrow \neg C \quad [N]}{\neg C}\Rightarrow_e \quad \dfrac{[R \wedge C]}{C}\wedge_{e_2}}{\dfrac{\bot}{B}\bot_e} \quad \dfrac{[R \wedge C]}{R}\wedge_{e_1}}{N \vee B}\Rightarrow_e \quad [B]}{\mathrm{B}}\vee_e \quad \dfrac{[R \wedge C]}{C}\wedge_{e_2}}{B \wedge C}\wedge_i}{X}\Rightarrow_e}{R \wedge C \Rightarrow X}\Rightarrow_i \qquad B \wedge C \Rightarrow X$$

9 (2 punti). Effettuare la seguente sostituzione minimizzando il numero di cambi di nome alle variabili.

$$(\Sigma_{i=0}^{i+1}\Sigma_{j=0}^{i} j + x)[i + y/x]$$

Vedi esercizi 9 nello scorso anno accademico.
Soluzione:

$$\Sigma_{k=0}^{i+1}\Sigma_{j=0}^{k} j + i + y$$

La i legata dal Σ più esterna è stata ridenominata per non catturare la i globale che occorre libera in $i+y$. Notate che la i ad apice (l'estremo superiore della sommatoria) è libera in quanto lo scope della variabile legata in una sommatoria non ne include gli estremi.

Esercizio di applicazione della teoria. In questo caso relativa alla nozione di sostituzione nelle sintassi con binders. Un'altra possibilità potrebbe essere la formalizzazione di sentenza in logica del prim'ordine, o il partizionamento di formule secondo la relazione di α-conversione. Vedi esempi per gli a.a. precedenti.

10 (3 punti). Dimostrare il seguente teorema usando la deduzione naturale al prim'ordine, preferendo una prova intuizionista a una classica ove possibile:

$$\forall x.\forall y.(f(x) \leq y \Rightarrow x \leq g(y)) \vdash (\exists x.f(h(x)) \leq x) \Rightarrow \exists z.\exists w.(z \leq g(w))$$

Esercizio di deduzione naturale per la logica del prim'ordine. Vedi esercizi analoghi nell'a.a. 2016-2017.

$$\dfrac{[\exists x.f(h(x)) \leq x] \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x.\forall y.(f(x) \leq y \Rightarrow x \leq g(y))}{\forall y.(f(h(u)) \leq y \Rightarrow h(u) \leq g(y))}\forall_e}{f(h(u)) \leq u \Rightarrow h(u) \leq g(u)}\forall_e \qquad [f(h(u)) \leq u]}{h(u) \leq g(u)}\Rightarrow_e}{\exists w.(h(u) \leq g(w))}\exists_i}{\exists z.\exists w.(z \leq g(w))}\exists_i}{\exists z.\exists w.(z \leq g(w))}\exists_e}{(\exists x.f(h(x)) \leq x) \Rightarrow \exists z.\exists w.(z \leq g(w))}\Rightarrow_i$$

Ultime considerazioni

Il punteggio relativo degli esercizi 2/3/7/8 potrebbe variare per riflettere la difficoltà degli esercizi nel compito. Esempio: più punteggio al 2 e meno al 7.

Gli esercizi 2/3/7/8 conferiscono la maggior parte del punteggio e coprono le competenze più rilevanti che dovreste acquisire con questo esame.

É possibile ritirarsi dall'esame prima della consegna. In caso di respingimento all'esame (bocciatura), essa verrà registrata sul libretto e sarà pertanto visibile nella carriera. Chi si ritira non viene respinto.

La sufficienza corrisponde a 18/30. Al punteggio dello scritto viene poi sommato il bonus del laboratorio.

In caso di quasi sufficienza (punteggi $\in (16, 18)$) lo studente può accettare un 18 rinunciando a sommare poi il bonus di laboratorio.

Chi non ha frequentato il laboratorio deve sostenere l'orale, che può essere comunque sostenuto da tutti. Per chi sostiene l'orale il voto finale dell'esame è dato dal voto dell'orale. Per gli altri dal voto del solo scritto. È possibile essere respinti all'orale. In tal caso a discrezione del docente verrà chiesto di sostenere nuovamente l'orale o di risostenere anche lo scritto.