Università degli Studi di Bologna

Corso di Laurea in Informatica
Esercitazione scritta di LOGICA PER L'INFORMATICA
13/01/2017

1 (1 punto). Dare la sintassi per le formule della logica proposizionale.
$F ::= \bot \mid \top \mid A \mid B \mid \ldots \mid F \wedge F \mid F \vee F \mid \neg F \mid F \Rightarrow F$

2 (5 punti). Scrivere una funzione $n(F)$ per ricorsione strutturale su F che restituisca la formula ottenuta a partire dalla F applicando il più possibile la regola di semplificazione $F \wedge F \equiv F$.

Esempi:

- $n((A \wedge B) \vee (A \wedge A)) = (A \wedge B) \vee A$
- $n((A \wedge A) \wedge (A \wedge A)) = A$

$n(\bot) = \bot$
$n(\top) = \top$
$n(A) = A$
$\ldots$
$n(F_1 \vee F_2) = n(F_1) \vee n(F_2)$
$n(F_1 \Rightarrow F_2) = n(F_1) \Rightarrow n(F_2)$
$n(\neg F) = \neg n(F)$
$n(F_1 \wedge F_2) = \texttt{if}\ n(F_1) = n(F_2)\ \texttt{then}\ n(F_1)\ \texttt{else}\ n(F_1) \wedge n(F_2)$

3 (1 punto). Enunciare il teorema di deduzione semantica.
$\forall \Gamma, F, G.\ (\Gamma \Vdash F \Rightarrow G) \iff (\Gamma, F \Vdash G)$

4 (1 punto). Dimostrare che una tabella di verità per un connettivo n-ario ha esattamente 2^n righe.
Per induzione su n. Caso 0: un connettivo 0-ario (una costante) assume un solo valore e la tabella ha una sola riga; $2^0 = 1$. Caso $n+1$: per ipotesi induttiva un connettivo n-ario ha 2^n righe. Le righe di un connettivo $(n+1)$-ario o assegnano 0 o assegnano 1 all'input $(n+1)$-esimo. Ci sono tante righe che finiscono con 0/1 quante righe ci sono per un connettivo n-ario, ovvero 2^n per ipotesi induttiva. Quindi il numero di riche è $2^n + 2^n = 2 \cdot 2^n = 2^{n+1}$.

5 (1 punto). Enunciare il teorema di compattezza per la logica proposizionale classica.
$\forall \Gamma, F.(\Gamma \Vdash F \Rightarrow \exists \Delta.(\Delta \subseteq \Gamma \wedge \Delta \texttt{finito} \wedge \Delta \Vdash F))$

6 (1 punto). Dimostrare il teorema di correttezza per la logica proposizionale classica, assumendo che tutti i connettivi siano locamente corretti.
Vedi prova nei lucidi.

7 (8 punti). Sia $\mathcal{P}_1$ il frammento della logica proposizionale comprendente solamente variabili, $\bot$, negazioni e congiunzioni, e sia $\mathcal{P}_2$ il frammento comprendente solamente variabili, $\top$, negazioni e disgiunzioni. Per ogni F, formula di $\mathcal{P}_1$, dimostrare, per induzione strutturale su F, l'esistenza di una formula G di $\mathcal{P}_2$ logicamente equivalente a F.

Suggerimento: è possibile utilizzare nella dimostrazione equivalenze logiche notevoli (tipo $A \vee A \equiv A$).
Enunciato: $\forall F \in \mathcal{P}_1.\exists G.(G \in \mathcal{P}_2 \wedge F \equiv G)$. Dimostrazione per induzione strutturale su F.
Caso A: dimostro $\exists G.(G \in \mathcal{P}_2 \wedge A \equiv G)$. Ovvio scegliendo A per G.
Caso $\bot$: dimostro $\exists G.(G \in \mathcal{P}_2 \wedge \bot \equiv G)$. Ovvio scegliendo $\neg\top$ per G.
Caso $\neg F_1$: per ipotesi induttiva $\exists G_1.(G_1 \in \mathcal{P}_2 \wedge F_1 \equiv G_1)$. Dimostro $\exists G.(G \in \mathcal{P}_2 \wedge \neg F_1 \equiv G)$. Per ipotesi induttiva sia $G_1 \in \mathcal{P}_2$ tale che $F_1 \equiv G_1$. La conclusione è ovvia scegliendo $\neg G_1$ per G. *Nota per gli studenti: $\neg G_1 \in \mathcal{P}_2$ perchè $G_1 \in \mathcal{P}_2$. Questo non varrebbe per esempio per $\neg F_1$.*
Caso $F_1 \wedge F_2$: per ipotesi induttiva $\exists G_i.(G_i \in \mathcal{P}_2 \wedge F_i \equiv G_i)$. Dimostro $\exists G.(G \in \mathcal{P}_2 \wedge (F_1 \wedge F_2 \equiv G))$. Per ipotesi induttiva siano $G_1, G_2 \in \mathcal{P}_2$ tali che $F_i \equiv G_i$ per $i \in \{1,2\}$. La conclusione è ovvia scegliendo $\neg(\neg G_1 \vee \neg G_2)$ per G.

8 (8 punti). Si consideri il seguente ragionamento:

Se i militanti fossero stati favorevoli e l'Alde li avesse voluti, i 5 stelle avrebbero aderito all'Alde. Per non rimanere fregati, i 5 stelle avrebbero dovuto aderire all'Alde o rimanere nel Ukip e alle stesse condizioni. I militanti sono stati favorevoli, portanto i 5 stelle a farsi fregare alla grande. Quindi l'Alde non li ha voluti e, se rimangono nel Ukip, non lo faranno alle stesse condizioni.

Verificare la correttezza del ragionamento utilizzando la deduzione naturale per la logica proposizionale. Preferire una prova intuizionista se possibile.

$M \wedge F \Rightarrow A$
$(A \vee U \wedge C) \Rightarrow \neg K$
$M \wedge K$
$\neg F \wedge (U \Rightarrow \neg C)$

È possibile costruire un albero di deduzione naturale intuizionista. (Albero omesso)

9 (2 punti). Nel seguente frammento di programma C++ fare l'inlining della funzione `f` in `main` (ovvero, espandere il codice della `f` nel corpo del `main` per evitare il costo associato alla chiamata di funzione), minimizzando il numero di cambi di nome alle variabili e non alterando la semantica del programma.

```
...

int f(int z, int y) {
        return (res*z+y*i);
}

int main() {
        for(int i = 0; i < BOUND; i++)
                res = f(res,i);
        return res;
}
```

Soluzione:

```
...
int main() {
        for(int j = 0; j < BOUND; j++)
                res = res*res + j*i);
        return res;
}
```

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

$$\neg(\forall x.\exists y.P(x,y)) \Rightarrow \exists x.\forall y.\neg P(x,y)$$

(Albero omesso. Fare attenzione quando si usa la regola di introduzione del $\forall$ che la variabile introdotta DEVE essere fresca. Inoltre quando si deve dimostrare un esistenziale è necessario usare la RAA per ritardare la scelta del testimone fino a quando non si è certi di quale sia.)