Logica per l'Informatica

Ricorsione ed Induzione strutturale

28/11/2023

Esercizio 1: Il tipo dei numeri naturali (in unario). Ricorda la definizione dei numeri naturali come tipo induttivo:

$$\texttt{Nat} ::= \texttt{O} \mid \texttt{S Nat}$$

e come sempre, diciamo che il termine $\texttt{O}$ rappresenta il numero naturale $0 \in \mathbb{N}$, il termine $\texttt{1} := \texttt{S}\,\texttt{O}$ rappresenta $1 \in \mathbb{N}$, il termine $\texttt{2} := \texttt{S}\,(\texttt{S}\,\texttt{O})$ rappresenta $2 \in \mathbb{N}$ ecc (si tratta della rappresentazione unaria dei naturali).

1. Definisci (senza usare nessuna funzione supplementaria!) una funzione ricorsiva strutturale $\texttt{dbl} : \texttt{Nat} \to \texttt{Nat}$ che ritorna il (termine di $\texttt{Nat}$ che rappresenta il) doppio del numero naturale (rappresentato dal termine di $\texttt{Nat}$) in input.

2. Verifica che:

   Theorem 1

   $$\texttt{dbl}\,\texttt{3} = \texttt{6}.$$

Esercizio 2: Il tipo delle liste di numeri naturali. Ricorda la definizione di liste di numeri naturali come tipo induttivo:

$$\texttt{listN} ::= [\,] \mid \texttt{Nat} :: \texttt{listN}$$

1. Definisci una funzione ricorsiva strutturale $\texttt{repeat} : \texttt{listN} \to \texttt{listN}$ tale che la lista $\texttt{repeat}\,l$ contiene ogni elemento di $l : \texttt{listN}$ in successione due volte.

   Per esempio, $\texttt{repeat}\,(\texttt{3} :: \texttt{2} :: \texttt{2} :: [\,]) = \texttt{3} :: \texttt{3} :: \texttt{2} :: \texttt{2} :: \texttt{2} :: \texttt{2} :: [\,]$.

2. Ricorda la definizione ricorsiva strutturale della funzione $\texttt{len} : \texttt{listN} \to \texttt{Nat}$ che calcola la lunghezza (espressa come un termine di tipo $\texttt{Nat}$) di una lista di numeri naturali (espressa come un termine di tipo $\texttt{listN}$):

$$\texttt{len}\,[] := \texttt{0}$$
$$\texttt{len}\,(n :: l) := \texttt{S}\,(\texttt{len}\,l).$$

Dimostra il seguente:

Theorem 2

$$\forall l : \texttt{listN}. \qquad \texttt{len}\,(\texttt{repeat}\,l) = \texttt{dbl}\,(\texttt{len}\,l).$$

Esercizio 3: Il tipo dei booleani. Ricorda la definizione dei booleani come tipi induttivi:

$$\texttt{Bool} ::= \texttt{true} \mid \texttt{false}$$

1. Definisci per ricorsione strutturale la funzione $\texttt{not} : \texttt{Bool} \to \texttt{Bool}$ che esprime la negazione sui termini di tipo $\texttt{Bool}$.

   Dopodiché, definisci per ricorsione strutturale la funzione $\texttt{parity} : \texttt{Nat} \to \texttt{Bool}$ che ritorna $\texttt{true}$ se l'input rappresenta un numero naturale pari, e ritorna $\texttt{false}$ altrimenti. Per esempio, $\texttt{parity}\,\texttt{0} = \texttt{true}$ e $\texttt{parity}\,\texttt{3} = \texttt{false}$.

2. Dimostra il seguente:

   Theorem 3

   $$\forall n : \texttt{Nat}. \qquad \texttt{parity}\,(\texttt{dbl}\,n) = \texttt{true}.$$

3. Dimostra il seguente (lo si può dimostrare senza induzione, ma fallo per induzione lo stesso):

   Theorem 4

   $$\forall n : \texttt{Nat}. \forall m : \texttt{Nat}. \qquad n = \texttt{S}\,(\texttt{dbl}\,m) \to \texttt{parity}\,n = \texttt{false}.$$

   Avrai bisogno di usare la seguente proprietà, che puoi assumere come dimostrata: $\forall n : \texttt{Nat}.\texttt{0} \neq \texttt{S}\,n$.

Esercizio 4. Ricorda la definizione della funzione ricorsiva strutturale $+ : \texttt{Nat} \to \texttt{Nat} \to \texttt{Nat}$ che fa la somma dei numeri naturali in input:

$$0 + m := m$$

$$(\texttt{S}\, n) + m := \texttt{S}\,(n + m).$$

1. Dimostra il seguente[1]:

   Theorem 5

   $$\forall n : \texttt{Nat}.\, \forall m : \texttt{Nat}.$$
   $$((\, \texttt{parity}\, n = \texttt{true} \to \texttt{parity}\,(n+m) = \texttt{parity}\, m\,)$$
   $$\wedge$$
   $$(\, \texttt{parity}\, n = \texttt{false} \to \texttt{parity}\,(n+m) = \texttt{not}\,(\texttt{parity}\, m)\,)).$$

2. Ricorda anche la definizione della funzione ricorsiva strutturale $\texttt{suml} : \texttt{listN} \to \texttt{Nat}$ che ritorna la somma (espressa come un termine di tipo $\texttt{Nat}$) di tutti i numeri naturali contenuti nella lista in input:

   $$\texttt{suml}\,[\,] := \texttt{O}$$

   $$\texttt{suml}\,(n :: l) := n + (\texttt{suml}\, l).$$

   Assumendo che $+$ sia associativa[2] ed assumendo come dimostrata la proprietà: $\forall n : \texttt{Nat}. n + n = \texttt{dbl}\, n$, dimostra il seguente (ricordati che puoi usare i teoremi precedentemente dimostrati):

   Theorem 6

   $$\forall l : \texttt{listN}. \qquad \texttt{parity}\,(\texttt{suml}\,(\texttt{repeat}\, l)) = \texttt{true}.$$

Esercizio 5: Il tipo degli alberi binari etichettati con numeri naturali. Definiamo il tipo induttivo degli alberi binari con numeri naturali come nodi come segue:

$$\texttt{Tree} ::= \texttt{Leaf Nat} \mid \texttt{Node Nat Tree Tree}$$

Per esempio, disegna su carta l'albero rappresentato dal termine $T_0 : \texttt{Tree}$ seguente:

$$\texttt{Node 3}\,(\texttt{Node O}\,(\texttt{Leaf 3})\,(\texttt{Node O}\,(\texttt{Leaf 1})\,(\texttt{Leaf 2})))\,(\texttt{Leaf 6}).$$

[1] L'enunciato è equivalente a dire che $\texttt{parity}\,(n+m)$ è lo XNOR di $\texttt{parity}\, n$ e $\texttt{parity}\, m$. Lo XNOR è il connettivo definito dalla seguente tabella di verità: $1\,\mathrm{XNOR}\,1 = 1 = 0\,\mathrm{XNOR}\,0$, $1\,\mathrm{XNOR}\,0 = 0 = 0\,\mathrm{XNOR}\,1$.

[2] Questo significa che $(n+m)+k = n+(m+k)$ per ogni $n, m, k : \texttt{Nat}$.

1. Definisci due funzioni ricorsive strutturali $\texttt{rmost}, \texttt{lmost} : \texttt{Tree} \to \texttt{Nat}$ che ritornano, rispettivamente, la foglia più a destra dell'albero in input, e la foglia più a sinistra dell'albero in input.

2. Usando $\texttt{app}$, definisci una funzione ricorsiva strutturale $\texttt{foglie} : \texttt{Tree} \to \texttt{listN}$ che ritorna la lista delle foglie dell'albero in input, lette da sinistra a destra.

Esercizio 6.

1. Ricorda la definizione della funzione $\texttt{app} : \texttt{listN} \to \texttt{listN} \to \texttt{listN}$ che concatena le due liste in input:

$$\texttt{app}\,[]\,l' := l'$$

$$\texttt{app}\,(n :: l)\,l' := n :: (\texttt{app}\,l\,l').$$

Ricorda la definizione della funzione $\texttt{head}$ che prende in input una lista di naturali e ritorna in output il primo elemento della lista in input, se esiste; altrimenti ritorna un simbolo fissato $\perp$:

$$\texttt{head}\,[] := \perp$$

$$\texttt{head}\,(n :: l) := n.$$

Dimostra il seguente:

Theorem 7

$$\forall l : \texttt{listN}.\, \forall l' : \texttt{listN}. \qquad l \neq [] \to \texttt{head}\,(\texttt{app}\,l\,l') = \texttt{head}\,l.$$

2. Dimostra il seguente (ricorda che puoi usare i teoremi che hai dimostrato precedentemente):

Theorem 8

$$\forall T : \texttt{Tree}. \qquad \texttt{head}\,(\texttt{foglie}\,T) = \texttt{lmost}\,T.$$

Per farlo avrai bisogno di un lemma, scopri qual è e dimostralo.