Lógica Computacional, 2017-2

Práctica 8: Razonamiento ecuacional

Víctor Zamora Gutiérrez   Manuel Soto Romero

Fecha de inicio: 17 de mayo de 2017
Fecha de término: 26 de mayo de 2017

Instrucciones generales

- La entrega es por **equipos de 3 a 4 integrantes**. Seguir los lineamientos especificados en: `http://sites.ciencias.unam.mx/logica-computacional-2017-2/laboratorio/lineamientos`.

Ejercicios

El objetivo de esta práctica es utilizar las reglas de deducción natural en Coq. Para ello, se utilizará el tipo booleano definido como sigue:

```
Inductive booleano: Type :=
|btrue: booleano
|bfalse: booleano
```

y sus operaciones `neg`, `and`, `or` e `impl`:

```
Definition and (b1 b2: booleano) : booleano:=
   match b1 with
   |bfalse => bfalse
   |_ => b2
   end.

Definition or (b1 b2: booleano) : booleano:=
   match b1 with
   |btrue => btrue
   |_ => b2
   end.

Definition impl (b1 b2: booleano) : booleano:=
   match b1 with
   |bfalse => b2
   |_ => btrue
   end.

Definition neg (b1:booleano) : booleano:=
   match b1 with
   |btrue => bfalse
```

```
|bfalse => btrue
end.
```

Parte I

Primero, se deben probar algunas reglas de deducción natural y con base en esto se probarán algunas otras propiedades:

1. **Reglas de la conjunción**
   `Theorem andI: forall (p q :booleano), p = btrue -> q = btrue -> and p q = btrue.`
   `Theorem andE1: forall (p q:booleano), and p q = btrue -> p = btrue.`
   `Theorem andE2: forall (p q:booleano), and p q = btrue -> q = btrue.`
2. **Regla de la doble negación**
   `Theorem dobleNeg: forall (p:booleano), neg(neg(p)) = p.`
3. **Eliminación de la impliación**
   `Theorem elimImpl: forall(f g:booleano), f=btrue -> impl f g=btrue -> g=btrue.`
4. **Modus Tollens**
   `Theorem modusTollens: forall(f g: booleano), impl f g = btrue -> neg g = btrue -> neg f = btrue.`
5. `Theorem introImpl: forall (p q:booleano), p=btrue-> q=btrue-> impl p q=btrue.`

Parte II

Probar los siguientes ejemplos: solo está permitido usar las tácticas `apply`, `intros`, `rewrite`, `exact` y `trivial`. Además, si se escriben lemas auxiliares, se deben usar exclusivamente estas tácticas para probarlos.

1. `Example ejemplo1: forall (p q r:booleano), and p q = btrue-> r = btrue -> and q r=btrue.`
2. `Example ejemplo2: forall (p q r: booleano), p = btrue-> neg (neg(and q r))=btrue-> and (neg (neg p)) r= btrue.`
3. `Example ejemplo3: forall (p q r:booleano), and (neg p) q = btrue -> impl (and (neg p) q) (or r (neg p))=btrue -> or r (neg p)=btrue.`
4. `Example ejemplo4: forall (p q r:booleano), impl p (impl q r)= btrue -> p=btrue->neg r = btrue ->neg q = btrue.`

Parte III

Probar las siguientes propiedades para lógica de predicados:

1. `Theorem elimForall: forall (p:A->Prop), (forall y:A, p y) -> p x.`

2. `Theorem introExist: forall (p: A->Prop), p x->exists (y:A), p y.`

Parte IV

Demostrar los siguientes ejemplos usando solo las tácticas `intros`, `apply`, `trivial`, `exists`, `destruct` y `exact`.

1. `Example ejemplo5: forall (p q :A->Prop), (forall (y:A), p y -> q y) -> (forall (y:A), p y) -> (forall (y:A), q y).`
2. `Example ejemplo6: forall (p:A->Prop), (forall (y:A), p y) -> (exists (y:A), p y).`
3. `Example ejemplo7: forall (p q :A->Prop), (forall (y:A), p y -> q y) -> (exists (y:A), p y) -> (exists (y:A), q y).`