Laborator 7

Demonstrații prin inducție

Demonstrații prin inducție

- Pentru a demonstra o proprietate a unei specificații prin inducție, proprietatea trebuie demonstrată pentru fiecare constructor al specificației.

Un prim exemplu

Să considerăm specificația pentru numere naturale în care avem operația:

$$sum(N) = 0 + 1 + \ldots + N.$$

```
fmod NAT-SUM is
   protecting NAT .
   var N : Nat .
   op sum : Nat -> Nat .
   eq sum(0) = 0 .
   eq sum(s N) = sum(N) + (s N) .
endfm
```

Un prim exemplu

Vrem să demonstrăm prin inducție pe numere naturale că:

$$0 + 1 + \ldots + N = \frac{N(N+1)}{2}.$$

O soluție în Maude este următoarea (o demonstrație prin inducție):

- Definim un modul în care includem
  - proprietatea ce trebuie demonstrată ca o operație cu rezultat `Bool`.
  - o constantă pentru pasul de inducție (devine o variabilă cuantificată existențial)

```
fmod SUM-LEMMA is
protecting NAT-SUM .
var N : Nat .
op sum-lemma : Nat -> Bool .
eq sum-lemma(N) = (2 * sum(N) == N * (s N)) .
*** constant for proof
op n : -> Nat .
endfm
```

Un prim exemplu

- Verificăm pasul de bază (pentru 0):

```
*** base step
red sum-lemma(0) .
*** should be true
```

- Verificăm pasul de inducție ($n \to s\ n$):

```
*** induction step
red sum-lemma(n) implies sum-lemma(s n) .
*** should be true
```

Exercițiul 1

Demonstrați prin inducție următoarele proprietăți pentru specificația:

```
fmod MYNAT is
   sort Nat .
   op 0 : -> Nat .
   op s_ : Nat -> Nat .
   op _+_ : Nat Nat -> Nat .
   vars N M : Nat .
   eq N + 0 = N .
   eq N + s M = s(N + M) .
endfm
```

1. `s(N) + M = s(N + M)` (Atenție! Inducția se face după *M*.)
2. `0 + M = M`
3. `N + M = M + N` (comutativitatea)
   - într-o demonstrație puteți folosi orice proprietate deja demonstrată ca ipoteză
4. `(N + M) + P = N + (M + P)` (asociativitatea)

Exercițiul 2

Demonstrați prin inducție următoarele proprietăți pentru specificația:

```
fmod MYNAT is
   sort Nat .
   op 0 : -> Nat .
   op s_ : Nat -> Nat .
   op _+_ : Nat Nat -> Nat .
   vars N M : Nat .
   eq N + 0 = N .
   eq N + s M = s(N + M) .
   op _<_ : Nat Nat -> Bool .
   eq s(N) < s(M) = N < M .
   eq 0 < s(M) = true .
   eq N < 0 = false .
endfm
```

Exercițiul 2 (cont.)

1. `N < s(N) = true`
2. `M1 + N < M2 + N = true if M1 < M2`
3. `M + N1 < M + N2 = true if N1 < N2`
4. `(M1 + N1) + M2 < (M1 + M2) + N2 = true if N1 < N2`

Exercițiul 3

Demonstrați prin inducție asociativitatea lui append pentru specificația:

```
fmod LIST-INT2 is
   protecting INT .
   sorts NList List .
   subsort Int < NList < List .
   op nil : -> List .
   op _ _ : Int List -> NList [id: nil] .
   op append : List List -> List .
   var I : Int . vars L1 L2 : List .
   eq append(nil,L2) = L2 .
   eq append(I,L2) = I L2 .
   eq append(I L1,L2) = I append(L1,L2) .
endfm
```

Pe săptămâna viitoare!