

Exercicis de Programació 2:

Correctesa de programes iteratius

Ricard Gavaldà

29 de març de 2016

Preguntes prèvies de comprensió:

- Digues què vol dir per a un programa “ser correcte” respecte a una especificació Pre/Post
- Digues què significa una Precondició “cert” a la vista d’aquesta definició.
- Digues què significa una Postcondició “cert” a la vista d’aquesta definició.
- Digues què pot significar una Precondició “fals” a la vista d’aquesta definició. I una Postcondició “fals”?
- Un invariant és 1) una variable 2) un nombre enter 3) una expressió booleana 4) una expressió entera 5) Una instrucció. Quina és correcta?
- Cert o fals: tot programa té invariant, fins i tot si està format només per assignacions i if’s.
- Cert o fals: l’invariant d’un bucle es compleix abans d’entrar al bucle.
- Cert o fals: l’invariant d’un bucle es compleix després que el bucle ha acabat.
- Cert o fals: l’invariant d’un bucle serveix per demostrar que en acabar es compleix la Postcondició.
- Cert o fals: l’invariant d’un bucle és compleix sempre que la condició d’entrada al bucle sigui certa.

Exercici 1. Digues l'invariant d'aquest bucle:

```
//Pre: x = X > 0, y = Y
while (x > 0) { y += y; --x; }
//Post: y = Y * 2^X (on ^ indica potència)
```

i usa'l per demostrar la seva correctesa.

Exercici 2. Digues l'invariant d'aquest bucle:

```
//Pre: l = L, x = 0
while (l.size() > 0) { x += *(l.begin()); l.erase(l.begin()); }
//Post: l és buida, x és la suma dels elements de L
```

i usa'l per demostrar la seva correctesa.

Exercici 3. Digues l'invariant d'aquest bucle:

```
//Pre: (i == v.size) and (s és la suma dels elements de v)
while (i > 0) {
    --i;
    if (v[i] % 2 == 0) s -= v[i];
}
//Post: s és la suma dels elements senars de v
```

i usa'l per demostrar la seva correctesa.

Exercici 4. Digues l'invariant d'aquest bucle:

```
//Pre: w és un vector de mida 0, v conté només elements positius,
//      i = 0, j = v.size()-1
while (i <= j) {
    if (v[i] > v[j]) w.push_back(2*v[i]);
    else w.push_back(2*v[j]);
    ++i; --j;
}
//Post: w.size() >= v.size()/2 and la suma dels elements de w
//      és més gran o igual que la suma dels elements de v
```

i usa'l per demostrar la seva correctesa.

Nota: en aquest cas, la Post és feble en el sentit que no caracteritza ben bé què fa el bucle. Això és intencionat. Poden haver-hi Posts que no determinin de manera única una “solució”. Per exemple, noteu que el següent algorisme té la mateixa Pre/Post amb un invariant completament diferent:

```
while (i < v.size()) {  
    w.push_back(v[i]+1);  
    ++i;  
}
```