UNED. Centro Asociado de VITORIA-GASTEIZ
Escuela Técnica de Informática
Programación II

La presente colección de ejercicios ha sido elaborada para su distribución gratuita a los alumnos de Programación II de Informática de Sistemas y Gestión del CENTRO ASOCIADO DE VITORIA-GASTEIZ de la Escuela Técnica de Informática de U.N.E.D.. La colección se ofrece para uso personal de dichos alumnos y no para ningún tipo de actividad sujeta a intercambio comercial de ningún tipo. En cualesquiera otras condiciones la copia, transmisión o almacenamiento por cualquier medio fotográfico, informático, telemático u otros requiere la autorización expresa del autor.

Esta es una versión preliminar y se ha inspirado en el capítulo 2 del libro "Programación Metódica", Jose Luis Balcázar, McGraw-Hill 1993

Se agradecerá la comunicación de cualquier errata así como cualquier comentario o sugerencia.

© Jerónimo Quesada 1995

Ejemplos de verificación y derivación de instrucciones

EJEMPLO 1

Traducir a asignaciones simples la asignación múltiple:

<x,y,z>:=<x+5,y+x,z-y>

Solución:

Una posible traducción es (introduciendo las variables auxiliares xcopia e ycopia):

```
xcopia:=x;
ycopia:=y;
x:=x+5;
y:=y+xcopia;
z:=z-ycopia;
```

de esta forma se asegura que la secuencia de asignaciones es equivalente a la asignación simultánea de las tres variables.

Otra solución es:

```
z:=z-y; {se opera con y antes de que sea modificada}
y:=y+x; {se opera con x antes de que sea modificada,
         z no interviene }
x:=x+5; {no intervienen y ni z }
```

EJEMPLO 2

Traducir a asignaciones simples la asignación múltiple

<x,y,z>:=<x+y+z,x+2y+3z,5x+y-z>

SOLUCIÓN:

(usando las variables auxiliares xcopia e ycopia)

```
xcopia:=x;
ycopia:=y;
x:=x+y+z;
y:=xcopia+y+z;
z:=xcopia+ycopia+z;
```

no se puede obtener una solución más simple reordenando las asignaciones puesto que en todas las asignaciones a una variable dada intervienen las otras dos.

EJEMPLO 3

Establecer la precondición más general Q para la siguiente sentencia (x,y,z son números reales}:

{Q}
x:=y/z;
{R≡x>10}

Solución:

Antes de la asignación se ha de verificar:

$R_x^{y/z} \wedge Dom(y/z) \equiv (y/z>10) \wedge Dom(y/z)$

$y/z>10 \Leftrightarrow y>10*z$

Dom(y/z) determina los valores de z e y para los que la expresión se puede calcular. La expresión es calculable siempre que z>0 luego:

$Q \equiv y>10*z \wedge z>0$

EJEMPLO 4

Encontrar la precondición más general Q para el programa:
{Q}
caso x/y≥1 → x:=x-y;
□ x/y<1 ->x:=y-x;
{R≡ x≥0 }

Solución:

Si se ha llegado a la postcondición pasando por el primer caso se tendrá:

$R_x^{x-y} \equiv x/y \geq 1 \wedge x-y \geq 0$ y además y>0 para que se pueda evaluar

la división de protección del caso.

De forma análoga, si se ha llegado pasando por el segundo caso se tendrá:

$(x/y<1 \wedge x-y \geq 0) \wedge y>0$

Luego en cualquier caso:

$((x/y \geq 1 \wedge x-y \geq 0) \vee (x/y<1 \wedge x-y \geq 0)) \wedge y>0$

simplificando se obtiene:

$((x \geq y) \vee (x<y)) \wedge y>0 \equiv \{cierto\} \wedge y>0 \equiv y>0$ luego la precondición es:

$Q \equiv y>0$

EJEMPLO 5 (Balcázar)

Deducir las sentencias S en:

$\{Q\equiv x+y=T \wedge z=Z\}$
S
$\{R\equiv x+y=T \wedge x=z \wedge z=Z\}$

(se trata de cambiar el valor de x asignándole el de z pero manteniendo la suma x+y constante, z no se puede modificar)

SOLUCIÓN:

Para cumplir la segunda conjunción de la postcondición intentamos la sentencia x:=z y se tiene:

$\{z+y=T \wedge z=z \wedge z=Z \equiv z+y=T \wedge z=Z\}$
x:=z;
$\{R\equiv x+y=T \wedge x=z \wedge z=Z\}$

puesto que z no puede variarse para hacercarse a la precondición se hará una asignación sobre y, a y se la asignará una expresión genérica que puede ser función de x,y,z

así:

$\{Q\equiv x+y=T \wedge z=Z\}$
y:=E(x,y,z)
$\{z+y=T \wedge z=Z\}$
x:=z;
$\{R\equiv x+y=T \wedge x=z \wedge z=Z\}$

E(x,y,z) ha de ser tal que:

$\{z+y=T \wedge z=Z\}^{E(x,y,z)}_{y} \equiv \{z+E(x,y,z)=T=x+y \wedge z=Z\}$

de donde se deduce E(x,y,z)=x+y-z, el programa queda:

$\{Q\equiv x+y=T \wedge z=Z\}$
y:=x+y-z;
x:=z;
$\{R\equiv x+y=T \wedge x=z \wedge z=Z\}$

EJEMPLO 6 (Balcázar)
Deducir las sentencias S tales que:

{Q≡ x=X ∧ y=Y}
S
{R≡ x=Y ∧ y=X }

(intercambio de valores)

SOLUCIÓN:

Para cumplir la primera conjunción de la precondición se intenta: x:=y, así:

{y=Y ∧ y=X}
x:=y;
{R≡ x=Y ∧ y=X }

el predicado y=Y ∧ y=X sólo se puede cumplir si X=Y que puede no ser cierto ya que no está garantizado en la precondición, por lo tanto la sentencia obtenida no es válida. Se se intenta la sentencia y:=x se llega a una situación similar. Se introducirá entonces una nueva variable z y se hace x:=z:

{z=Y ∧ y=X} [1]
x:=z;
{R≡ x=Y ∧ y=X }

ahora se puede avanzar intentando cumplir la primera conjunción del predicado obtenido haciendo z:=y así:

{y=Y ∧ y=X}
z:=y;
{z=Y ∧ y=X}
x:=z;
{R≡ x=Y ∧ y=X }

y otra vez se llega una situación sin solución. Si en lugar de intentar cumplir la primera conjunción en [1] se intenta la segunda se tiene:

{z=Y ∧ x=X} [2]
y:=x
{z=Y ∧ y=X} [1]
x:=z;
{R≡ x=Y ∧ y=X }

ahora si se aproxima la precondición, basta intentar cumplir la primera conjunción del aserto [2]. Para ello se hace z:=y :

{y=Y ∧ x=X} [3]
z:=y;
{z=Y ∧ x=X} [2]
y:=x
{z=Y ∧ y=X} [1]
x:=z;
{R≡ x=Y ∧ y=X }
y [3] ya coincide con la precondición.

EJEMPLO 7 (Balcázar)

Determinar las sentencias S tales que:

{cierto}
S
{R≡ (z=x∨z=y)∧(z≥x∧z≥y)}

(z= máximo de x e y)

SOLUCIÓN:

Se puede expresar la postcondición como (intentarlo aplicando operaciones sobre predicados (capítulo 2 Peña)):

$(z\geq y\wedge z=x)\vee(z\geq x\wedge z=y)$

siempre que se tiene una disyunción de esta forma se pueden tratar cada uno de los miembros como casos independientes u tener en cuenta que algunas de las condiciones a cumplir en los miembros pueden alcanzarse con los predicados de protección del caso, el primero es $z\geq y\wedge z=x$, la segunda conjunción la intentamos cumplir con la asignación z:=x y se tiene:

$\{x\geq y\wedge x=x\}\equiv\{x\geq y\}$
z:=x;
$\{z\geq y\wedge z=x\}$

el predicado $x\geq y$ se coloca como protección del caso así se tiene:

caso $x\geq y \rightarrow$ z:=x;

para $z\geq x\wedge z=y$ se obtiene de forma análoga:

caso $y\geq x \rightarrow$ z:=y;

por tanto la solución es:

{cierto}

caso $x\geq y \rightarrow$ z:=x;
□ $y\geq x \rightarrow$ z:=y;
fcaso

$\{R\equiv (z=x\vee z=y)\wedge(z\geq x\wedge z\geq y)\}$

EJEMPLO 8 (Balcázar)

(Actualización de cociente y resto)

Deducir las sentencias S tales que:

$\{Q\equiv a=2*b*c+r\wedge 0\leq r<2*b\}$
S
$\{R\equiv a=b*c+r\wedge 0\leq r<b\}$

(a y b se han de mantener, no se indica expresamente para no complicar los predicados)

SOLUCIÓN:

Puesto que a y b se han de mantener se puede pensar en una asignación múltiple para c y r de la forma <c,r>:=<E1,E2> siendo E1 y E2 expresiones genéricas, así:

$\{a=b*E1+E2\wedge 0\leq E2<b\}$
<c,r>:=<E1,E2>
$\{R\equiv a=b*c+r\wedge 0\leq r<b\}$

Con objeto de acercarse a la precondición se puede pensar en E1=2*c y E2=r, así:

$\{a=b*2*c+r\wedge 0\leq r<b\}$
<c,r>:=<2*c,r>
$\{R\equiv a=b*c+r\wedge 0\leq r<b\}$

el único problema es que en la precondición se tiene r<2*b y antes de la asignación sólo se puede garantizar r<b. Si se considera la condición r<2*b dividida en dos casos: r<b y r≥b se tiene que la asignación es válida para el primer caso pero no para el segundo, se puede escribir:

caso r<b → <c,r>:=<2*c,r>;
□ r≥b → <c,r>:=<E3,E4>;
fcaso

se ha de intentar que

$R^{E3,E4}_{c,r}\wedge r\geq b \equiv a=b*2*c+r\wedge 0\leq r<2*b\wedge r\geq b \equiv a=b*2*c+r\wedge b\leq r<2*b$

es decir:

$a=b*E3+E4\wedge 0\leq E4<b\wedge r\geq b \equiv a=b*2*c+r\wedge b\leq r<2*b$

comparando $0\leq E4<b$ con $b\leq r<2*b$ parece una solución E4=r-b

entonces para que a=b*E3+E4=b*E3+r-b coincida con a=b*2*c+r se tiene E3=(b*2*c+r-r+b)/b=2*c+1, el programa queda:

caso r<b → <c,r>:=<2*c,r>;
□ r≥b → <c,r>:=<2*c+1,r-b>;
fcaso

EJEMPLO 9 (Balcázar)
(Acarreo del máximo)
Deducir las sentencias S tales que:

$\{Q\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K \wedge 1\leq k<n\}$
S
$\{R\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K+1 \wedge 1\leq k\leq n\}$

S pueden ser las sentencias interiores a un bucle de la forma

mientras k<n hacer
S
fmientras

que tiene como objetivo hallar el valor máximo de un vector, Q sería equivalente a P $\wedge$ k<n siendo P el invariante. Tras avanzar en el bucle se ha de mantener el invariante P, esto equivale a R.

SOLUCIÓN:

Primero se trata de cumplir la condición k=K+1 (avanzar) con la asignación k:=k+1, se tiene:

$\{\exists\alpha\in\{1..k+1\}.a[\alpha]=x \wedge \forall \beta\in\{1..k+1\}.a[\beta]\leq x \wedge k+1=K+1 \wedge 1\leq k+1\leq n\}$
k:=k+1;
$\{R\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K+1 \wedge 1\leq k\leq n\}$

Para conseguir $\exists\alpha\in\{1..k+1\}.a[\alpha]=x$ no se ha de hacer nada especial pues está implicado por la conjunción $\exists\alpha\in\{1..k+1\}.a[\alpha]=x$ de la precondición, k+1=K+1 está también implicado por la condición k=K de la precondición y $1\leq k+1\leq n$ lo está por la condición $1\leq k<n$ de la precondición. La única condición no garantizada por la precondición es entonces:

$\forall \beta\in\{1..k+1\}.a[\beta]\leq x$

que se puede expresar como:

$\forall \beta\in\{1..k\}.a[\beta]\leq x \wedge a[k+1]\leq x$

la primera conjunción está garantizada en la precondición, para garantizar la segunda se tiene lo siguiente:

si $a[k+1]\leq x$ no hay que hacer nada

si $a[k+1]>x$ se garantiza $a[k+1]\leq x$ haciendo x:=a[k+1] y se puede comprobar que esta asignación no estropea ninguna de las otras condiciones de la precondición. La solución obtenida es entonces:

$\{Q\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K \wedge 1\leq k<n\}$
caso $a[k+1]\leq x \rightarrow$ seguir;
$\square$ $a[k+1]>x \rightarrow$ x:=a[k+1];
fcaso
k:=k+1;
$\{R\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K+1 \wedge 1\leq k\leq n\}$
que equivale a:

$\{Q\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K \wedge 1\leq k<n\}$
si $a[k+1]>x$ hacer
x:=a[k+1];
fsi;
k:=k+1;
$\{R\equiv \exists\alpha\in\{1..k\}.a[\alpha]=x \wedge \forall \beta\in\{1..k\}.a[\beta]\leq x \wedge k=K+1 \wedge 1\leq k\leq n\}$

EJEMPLO 10
Deducir las sentencia S tal que

$$\{Q\equiv s=\sum_{\alpha=1}^{k} a[\alpha]\wedge k=K\}$$

S;

$$\{R\equiv s=\sum_{\alpha=1}^{k} a[\alpha]\wedge k=K+1\}$$

(sentencias interiores a un bucle de suma de un vector, para simplificar no se incluye la condición $1\leq k \leq n$)

SOLUCIÓN:

La sentencia avanzar será k:=k+1, es decir

$$\{ s=\sum_{\alpha=1}^{k+1} a[\alpha]\wedge k+1=K+1 \}$$

k:=k+1;

$$\{R\equiv s=\sum_{\alpha=1}^{k} a[\alpha]\wedge k=K+1 \}$$

y

$$s=\sum_{\alpha=1}^{k+1} a[\alpha]\wedge k+1=K+1 \text{ equivale a } s=\sum_{\alpha=1}^{k} a[\alpha]+a[k+1]\wedge k=K$$

que se puede conseguir con la sentencia s:=s+a[k+1], así:

$$\{s+a[k+1]=\sum_{\alpha=1}^{k+1} a[\alpha]\wedge k+1=K+1 \}$$

s:=s+a[k+1];

$$\{s=\sum_{\alpha=1}^{k+1} a[\alpha]\wedge k+1=K+1 \}$$

k:=k+1;

$$\{R\equiv s=\sum_{\alpha=1}^{k} a[\alpha]\wedge k=K+1 \}$$

Se puede comprobar que el predicado obtenido coincide con la precondición ya que:

$$s+a[k+1]=\sum_{\alpha=1}^{k+1} a[\alpha] \text{ equivale a } s=\sum_{\alpha=1}^{k} a[\alpha], \text{ la solución queda:}$$

$$\{Q\equiv s=\sum_{\alpha=1}^{k} a[\alpha]\wedge k=K \}$$

s:=s+a[k+1];
k:=k+1;

$$\{R\equiv s=\sum_{\alpha=1}^{k} a[\alpha]\wedge k=K+1 \}$$

EJEMPLO 11

Deducir las sentencias S tal que:

Deducir las sentencia S tal que

$$\{Q\equiv h=\sum_{\alpha=1}^{k}(a[\alpha]=x)\wedge k=K\}$$

S;

$$\{R\equiv h=\sum_{\alpha=1}^{k}(a[\alpha]=x)\wedge k=K+1\}$$

(sentencias interiores a un bucle que cuenta los elementos de un vector iguales a uno dado x, para simplificar no se incluye la condición $1\leq k\leq n$)

SOLUCIÓN:

Siguiendo un proceso similar al de los dos ejercicios anteriores se tiene:

$$\{Q\equiv h=\sum_{\alpha=1}^{k}(a[\alpha]=x)\wedge k=K\}$$

```
 si a[k+1]=x
   h:=h+1;
 fsi
 k:=k+1;
```

$$\{R\equiv h=\sum_{\alpha=1}^{k}(a[\alpha]=x)\wedge k=K+1\}$$