GIIM. Relación de problemas-Grupo A -adicionales. Tema 1. 16/09/2021

25. Usando la regla de la conjunción, demostrar que $\{i > 2\} i := 2 * i \{i > 4\}$

$$\frac{\begin{array}{rcl}\{V\}\, i & = & 2 * i\, \{i_\omega = 2 * i_\alpha\} \\ \{i_\alpha > 2\}\, i & = & 2 * i\, \{i_\alpha > 2\}\end{array}}{\{i_\alpha > 2\}\, i = 2 * i\, \{i_\omega > 4\}}$$

26. Se dan los siguientes triples de Hoare:

$$\{j > 1\}\, i = i + 2;\ j = j + 3;\ \{j > 4\}$$
$$\{i > 2\}\, i = i + 2;\ j = j + 3;\ \{i > 4\}$$

Demostrar que estos triples implican que $\{j > 1,\ i > 2\}\, i = i + 2;\ j = j + 3;$ $\{j > 4,\ i > 4\}$. ¿Qué regla se debe utilizar para la demostración?
La regla de la conjunción

27. Sean `A` y `B` los valores iniciales de `a` y `b` respectivamente. Escribir un fragmento de código que tenga `{a=A+B, b=A-B}` como poscondición y demostrar que el código es correcto. $\{a == A,\ b == B\}\ a = a + b; \{a == A, + B\ b == B\}\ b = a - 2 * b;\ \{a == A + B,\ b == A - B\}$

28. Demostrar que la siguiente sentencia tiene la poscondición $\{x \geq 0,\ x^2 = a^2\}$.
$if\ a > 0\ then\ x := a\ else\ x := -a$

$$\frac{\begin{array}{l}\{a > 0\}\, x = a\, \{x == a,\ a > 0\} \Rightarrow x^2 == a^2,\ x > 0\} \\ \{a \leq 0\}\, x = -a\, \{x == -a,\ a \leq 0\} \Rightarrow x^2 == a^2,\ x \geq 0\}\end{array}}{\{V\}\ if\ a > 0\ \ then\ \ x := a\ else\ x := -a\, \{x \geq 0,\ x^2 = a^2\}}$$

29. El siguiente fragmento de código tiene $\{P\} \equiv \{sum = j * (j-1)/2\}$ como precondición y poscondición. Demostrar que es verdadero: $\{P\}\ sum := sum + j; j := j + 1;\ \{P\}$
$\{sum = (j-1) * j/2\}\ sum = sum + j;\ \{sum = j * (j+1)/2\}\ j = j + 1;\ \{sum = (j-1) * j/2\}$

30. Demostrar que $\{i * j + 2 * j + 3 * i = 0\}\ j = j + 3;\ i = i + 2 \{i * j = 6\}$

$$\{i * j + 2 * j + 3 * i = 0\}\ j = j + 3;\ \{i * (j - 3) + 2 * (j - 3) + 3 * i = 0\}$$
$$\{i * (j - 3) + 2 * (j - 3) + 3 * i = 0\}\ i = i + 2; \{(i - 2) * (j - 3) + 2 * (j - 3) + 3 * (i - 2)$$
$$\Rightarrow \{i * j = 6\}$$

31. ¿Por qué en la regla del **while** B, la condición B debe ser verdadera al comienzo del bucle? Porque se necesita para demostrar el invariante final al terminar el bucle cuando la condición de salida del bucle es una desigualdad.

$$
\begin{array}{c}
sum = 0; i = 0; \\
\{sum == 0, i = 0\} \wedge \{sum == i * (i-1)/2\} \\
while(i < n)\ do \\
\{i < n\,,\ sum == i * (i-1)/2\} \\
i = i + 1; \\
\{i < n + 1,\ sum == (i-1) * (i-2)/2\} \\
sum = sum + i; \\
\{i < n + 1,\ sum == (i-1) * i/2\} \\
enddo; \\
\{i \geq n\} \wedge \{i < n + 1,\ sum == i * (i-1)/2\} \Rightarrow \{sum == (n-1) * n/2\}
\end{array}
$$

32. Considerar una función con dos argumentos que se usa en un programa. Explicar por qué el uso de alias puede ser un problema en este caso. Si los argumentos se pasan por referencia, se podrían producir 'alias' y entonces habría que considerar n casos de prueba.

$$
\begin{array}{l}
procedure P(var\ x,\ y) \\
begin \\
\{V\} \\
x = x + y; \\
\{x_\omega == x_\alpha + y_\alpha\} \\
end;
\end{array}
$$

Entonces, $P(a,\ a)$ tendría como poscondición $\{a == a + a \equiv F\}$

33. Demostrar la corrección parcial del siguiente fragmento de programa:

$$
\begin{array}{c}
sum := 0; j := 1; \\
while(j \neq c) do \\
begin \\
sum := sum + j; j := j + 1; \\
end \\
\{sum = c * (c-1)/2\}
\end{array}
$$

$$
\begin{array}{l}
\quad sum := 0; j := 1; \\
\{sum == 0, j == 1\} \wedge sum == j * (j-1)/2 \\
\quad while(j \neq c)\ do \\
\quad \{j \neq c,\ sum == j * (j-1)/2\} \\
\quad\quad begin \\
\quad\quad\quad sum = sum + j; \\
\quad \{j \neq c,\ sum == j * (j-1)/2 + j \equiv sum == (j+1) * j/2\} \\
\quad\quad\quad j = j + 1; \\
\{j \neq c + 1,\ sum == (j - 1 + 1) * (j - 1)/2 \equiv sum == j * (j - 1)/2\} \\
\quad\quad end \\
\{j == c\}, \{j \neq c + 1,\ sum == j * (j - 1)/2\} \\
\Rightarrow \{sum = c * (c-1)/2\}
\end{array}
$$

34. Demostrar la corrección del siguiente triple: $\{a[i] \geq 0\}\ a[i] = a[i] + a[j]\{a[i] \geq a[j]\}$

$$\{a[i] \geq 0\}\ a[i] = a[i] + a[j]\{a[i] - a[j] \geq 0\} \Rightarrow \{a[i] \geq a[j]$$

35. Verificar el siguiente segmento de programa:

$$
\begin{array}{c}
\{n \geq 0\} \\
i = 1; \\
while\ i \leq n\ do \\
begin \\
a[i] = b[i]; \\
i = i + 1; \\
end \\
\{\bigwedge_{i=1}^{n} (a[i] = b[i])\}
\end{array}
$$

$$\{n \geq 0\}$$
$$i = 1;$$
$$\{n \geq 0,\ i == 1\} \wedge \bigwedge_{j=1}^{i-1} a[j] == b[j]\}$$
$$while\ i \leq n\ do$$
$$\{n \geq 0,\ i \leq n\} \wedge \{\bigwedge_{j=1}^{i-1} a[j] == b[j]\}$$
$$begin$$
$$a[i] = b[i];$$
$$\{n \geq 0,\ i \leq n\} \wedge \{\bigwedge_{j=1}^{i-1} a[j] == b[j],\ a[i] == b[i]\} \equiv \{\bigwedge_{j=1}^{i} a[j] == b[j]\}$$
$$i = i + 1;$$
$$\{n \geq 0,\ i \leq n + 1\} \wedge \{\bigwedge_{j=1}^{i-1} a[j] == b[j]\}$$
$$end$$
$$\{i > n\} \wedge \{n \geq 0,\ i \leq n + 1\} \wedge \{\bigwedge_{j=1}^{i-1} a[j] == b[j]\} \Rightarrow$$
$$\{i == n + 1\} \wedge \{\bigwedge_{i=1}^{n} a[i] == b[i]\} \Rightarrow$$
$$\{\bigwedge_{i=1}^{n} (a[i] = b[i])\}$$

36. El siguiente fragmento de programa calcula $\Sigma_{i=1}^{n}\ i!$. Demostrar que es correcto.

$$i := 1; sum := 0;\ f := 1;$$
$$while\ i \neq n + 1\ \ do$$
$$begin$$
$$sum := sum + f;$$
$$i := i + 1;$$
$$f := f * i;$$
$$end$$

$$i = 1; sum = 0; f = 1;$$

$$\{ i == 1, sum == 0, f == 1\} \wedge \{sum = \sum_{j=1}^{i-1} j\,!,\ f = i\,!\}$$

$$while\ i \neq n+1\ \ do$$

$$\{i \neq n+1\} \wedge \{sum = \sum_{j=1}^{i-1} j\,!,\ f = i\,!\}$$

$$begin$$

$$sum = sum + f;$$

$$\{i \neq n+1, \} \wedge \{sum = \sum_{j=1}^{i-1} j\,! + f\} \equiv \{sum = \sum_{j=1}^{i-1} j\,!, + i\,!\} \Rightarrow$$

$$\{sum = \sum_{j=1}^{i} j\,!\} \wedge f == i\,!$$

$$i := i + 1;$$

$$\{sum = \sum_{j=1}^{i-1} j\,!\} \wedge f == (i-1)\,!$$

$$f = f * i\,;$$

$$\{sum = \sum_{j=1}^{i-1} j\,!\} \wedge f == i\,!$$

$$end$$

$$\{i == n+1\} \wedge \{sum = \sum_{i=1}^{n} i\,!\} \wedge f == (n+1)\,! \Rightarrow sum == \sum_{i=1}^{n} i\,!$$

37. Hallar la precondición $\{P\}$ que hace que el siguiente triple sea correcto: $\{P\}\, a[i] := 2{*}b\, \{j \leq i,\ k < i,\ a[i] + a[j-1] + a[k] > b\}$

$$\{j \leq i,\ k < i,\ b + a[j-1] + a[k] > 0\}$$

38. Demostrar que para $n > 0$ el siguiente fragmento de programa termina.

$$\begin{array}{l} i := 1; f := 1; \\ \quad while\ i \neq n\ do \\ \quad begin \\ \quad\quad i := i+1; \\ \quad\quad f := f * r; \\ \quad end \end{array}$$

La única variable en la condición de entrada al bucle que varía es la variable 'i', por lo tanto el *vector variante* en este caso es sólo $\{ i \}$, que describe la secuencia $\{0,\ 1,\ \ldots\ n-1,\ n\}$. Consecuentemente, cuando $i == n$, la condición de salida del bucle se cumnple y este terminará.

39. Hallar la precondición de la terna: $\{P\}\ a[i] := b\ \{a[j] = 2 * a[i]\}$ Hay 2 posibles casos, dependiendo si los valores de las variables i y j coinciden:
    - Caso: $i \neq j$: precondición, $\{a[j] = 2 * a[i]\}_b^{a[i]} \equiv \{a[j] == 2 * b\}$
    - Caso: $i == j$: entonces $a[i]$ y $a[j]$ han de considerarse la misma variable y cualquier sustitución de $a[i]$ también ha de aplicarse a $a[j]$. Por consiguiente, la precondición que se obtiene maplicando la regla de la asignación es $b == 2 * b \Leftrightarrow b == 0$

40. Para cada uno de los siguientes fragmentos de código, obtener la poscondición apropiada:
    (a) $\{i < 10\}\ i := 2 * i + 1\ \{i < 21\}$
    (b) $\{i > 0\}\ i := i - 1\ \{i + 1 > 0\} \Rightarrow \{i \geq 0\}$
    (c) $\{i > j\}\ i := i + 1; \{i \geq j + 1\}\ j := j + 1\ \{i > j\}$
    (d) $\{V\}\ i := 3;\ j := 2 * i\ \{j == 6\}$

41. Para cada uno de los siguientes fragmentos de código, obtener las precondiciones apropiadas.
    (a) $\{k > 2\}\ i := 3 * k\ \{i > 6\}$
    (b) $\{\, b * c == 1\}\ a := b * c\ \{a = 1\,\}$
    (c) $\{c \neq 2\}$ b:= c – 2; $\{b \neq 0\}$ a:= a/b; $\{V\}$

42. Verificar el siguiente código. Indicar todas las reglas usadas.
    $\{y > 0\}\ xa := x + y;\ \{xa > x,\ y > 0\}$ xb := x-y $\{xa > xb + y,\ y > 0\} \Rightarrow \{xa > xb\}$
    Reglas usadas: axioma de asignación, consecuencia, precondición más fuerte, poscondición más débil

43. Verificar el siguiente código, indicando todas las reglas usadas.
    $\{V\}\ if\ x < 0\ then\ x = -x \{x \geq 0\}$

$$\frac{\{x < 0\}\ x = -x\ \{x > 0\} \Rightarrow \{x \geq 0\} \qquad \neg\{x < 0\} \Rightarrow \{x \geq 0\}}{\{V\}\ if\ x < 0\ then\ x := -x\ \{x \geq 0\}}$$

44. Verificar el siguiente segmento de programa:

$$
\begin{array}{l}
max := a[1]; i = 1; \\
\quad while\ (i \neq n+1)\ do \\
\quad\quad begin \\
\quad\quad\quad if\ (a[i] >= max)\ then\ max := a[i]; \\
\quad\quad\quad i = i+1 \\
\quad\quad end \\
\quad\quad \{\bigwedge_{i=1}^{n}(max \geq a[i])\}
\end{array}
$$

$$max = a[1];\ i = 1; \{max == a[1],\ i == 1\} \wedge \sum_{k=1}^{i-1} max \geq a[k]$$

$$while\ (i \neq n+1)\ do$$

$$\{i \neq n+1\} \wedge \{\sum_{k=1}^{i-1} max \geq a[k]\}$$

$$begin$$

$$if\ (a[i] >= max)\ then\ max = a[i];$$

$$\{a[i] \geq max\} \wedge \{i \neq n+1\} \wedge \{\sum_{k=1}^{i-1} max \geq a[k]\}$$

$$max = a[i]; \Rightarrow \{\sum_{k=1}^{i} max \geq a[k]\} \wedge \{i \neq n+1\}$$

$$\{a[i] < max\} \wedge \{\sum_{k=1}^{i-1} max \geq a[k]\} \wedge \{i \neq n+1\} \Rightarrow \{\sum_{k=1}^{i} max \geq a[k]\} \wedge \{i \neq n+1\}$$

$$\{\sum_{k=1}^{i} max \geq a[k]\}$$

$$i := i+1$$

$$\{\sum_{k=1}^{i-1} max \geq a[k]\}$$

$$end$$

$$\neg\{i \neq n+1\} \wedge \{\sum_{k=1}^{i-1} max \geq a[k]\} \equiv \{i == n+1\} \wedge \{\sum_{k=1}^{i-1} max \geq a[k]\} \Rightarrow$$

$$\{\sum_{i=1}^{n}(max \geq a[i])\}$$

45. Demostrar la corrección parcial del siguiente código:

$$max = a[1]; i = 1;$$
$$while(i < n)\ do$$
$$begin$$
$$i = i + 1;$$
$$if(a[i] >= max)\ then\ max = a[i];$$
$$end \quad \{\bigwedge_{i=1}^{n}(max \geq a[i]), \bigvee_{j=1}^{n} max = a[i])\}$$

$$max = a[1]; i = 1;$$

$$\{max == a[1],\ i == 1\} \wedge \{\bigwedge_{k=1}^{i} max \geq a[k], \bigvee_{k=1}^{i} max == a[k]\}$$

$$while\ (i < n)\ do$$
$$begin$$

$$\{i < n\} \wedge \{\bigwedge_{k=1}^{i} max \geq a[k], \bigvee_{k=1}^{i} max == a[k]\}$$

$$i = i + 1$$

$$\{i < n + 1\} \wedge \{\bigwedge_{k=1}^{i-1} max \geq a[k], \bigvee_{k=1}^{i-1} max == a[k]\}$$

$$if\ (a[i] >= max)\ then\ max := a[i];$$

$$\{a[i] \geq max\} \wedge \{i < n+1\} \wedge \{\bigwedge_{k=1}^{i-1} max \geq a[k], \bigvee_{k=1}^{i-1} max == a[k]\}$$

$$max = a[i];$$

$$\{i < n+1\} \wedge \{\bigwedge_{k=1}^{i} max \geq a[k]\}, \bigvee_{k=1}^{i} max == a[k]\}$$

$$\{a[i] < max\} \wedge \{i < n+1\} \wedge \{\bigwedge_{k=1}^{i-1} max \geq a[k], \bigvee_{k=1}^{i-1} max == a[k]\} \Rightarrow$$

$$\{i < n+1\} \wedge \{\bigwedge_{k=1}^{i} max \geq a[k]\}, \bigvee_{k=1}^{i} max == a[k]\}$$

$$\{i \geq n\} \wedge \{i < n+1\} \wedge \{\bigwedge_{k=1}^{i} max \geq a[k], \bigvee_{k=1}^{i} max == a[k]\}$$

$$end$$

$$\{i == n\} \wedge \{\bigwedge_{k=1}^{i} max \geq a[k], \bigvee_{k=1}^{i} max == a[k]\} \Rightarrow$$

$$\{\bigwedge_{i=1}^{n}(max \geq a[i]), \bigvee_{j=1}^{n} max = a[i])\}$$

46. Demostrar la corrección parcial del siguiente código:

$$
\begin{array}{l}
i := 0; j := n; \\
\mathit{while}\ (i < n)\ \mathit{do} \\
\quad \mathit{begin} \\
\qquad i = i + 1; \\
\qquad j = j - 1; \\
\quad a[i] = b[j] \\
\quad \mathit{end} \\
\{\bigwedge_{i=1}^{n} (a[i] == b[n - i])\}
\end{array}
$$

$$
\begin{array}{l}
i = 0; j = n; \\
\{i == 0,\ j == n\} \wedge \{\bigwedge_{k=1,\ j=n-k}^{i,(n-i)} a[k] == b[j]\} \wedge \{j \leq n - i\} \\
\mathit{while}\ (i < n)\ \mathit{do} \\
\quad \mathit{begin} \\
\{i < n\} \wedge \{\bigwedge_{k=1,\ j=n-k}^{i,(n-i)} a[k] == b[j]\} \wedge j \leq (n - i)\} \\
i = i + 1;\ j = j - 1; \\
\{i < n + 1,\ j \leq (n - i + 1)\} \wedge \{\bigwedge_{k=1,\ j=n-k-1}^{i-1,(n-i+1)} a[k] = b[j]\} \\
a[i] = b[j]; \\
\{i < n + 1,\ j \leq (n - i)\} \wedge \{\bigwedge_{k=1,\ j=n-k}^{i,(n-i)} a[k] = b[j]\} \\
\quad \mathit{end} \\
\{i < n + 1\} \wedge \{i \geq n\} \wedge \{j \leq n - i\} \wedge \{\bigwedge_{k=1,\ j=n-k}^{i,(n-i)} a[k] = b[j]\} \Rightarrow \\
\{\bigwedge_{k=1,j=n-k}^{n,\ 0} (a[i] == b[j])\} \Rightarrow \\
\{\bigwedge_{i=1}^{n} (a[i] == b[n - i])\}
\end{array}
$$

47. Demostrar la corrección parcial del siguiente código:

$$
\begin{array}{r}
i = 1; \\
s = 0; \\
while\ (i \leq n)\ do \\
begin \\
s = s + a[i]; \\
a[i] = s; \\
i = i + 1; \\
end
\end{array}
$$

$$\{\bigwedge_{i=1}^{n}(a[i] == \sum_{k=1}^{n} A[k]\}$$

$$i = 1; s = 0;$$

$$\{\bigwedge_{k=1}^{i-1} a[k] == s,\ s == \sum_{k=1}^{i-1} A[k],\ \bigwedge_{k=i,i\geq 1}^{n} a[k] == A[k]\} \wedge \{i = 1,\ \ s = 0\}$$

$$while\ (i \leq n)\ do$$

$$\{i \leq n\} \wedge \{\bigwedge_{k=1}^{i-1} a[k] == s,\ s == \sum_{k=1}^{i-1} A[k],\ \bigwedge_{k=i,i\geq 1}^{n} a[k] == A[k]\}$$

$$s = s + a[i];\ a[i] = s;$$

$$\{i \leq n\} \wedge \{\bigwedge_{k=1}^{i} a[k] == s,\ s == \sum_{k=1}^{i} A[k],\ \bigwedge_{k=i+1,i\geq 1}^{n} a[k] == A[k]\}$$

$$i = i + 1;$$

$$\{i \leq n+1\} \wedge \{\bigwedge_{k=1}^{i-1} a[k] == s,\ s == \sum_{k=1}^{i-1} A[k],\ \bigwedge_{k=i,i\geq 1}^{n} a[k] == A[k]\}$$

$$end$$

$$\{i == n + 1\} \wedge \{\bigwedge_{k=1}^{i-1} a[k] == s,\ s == \sum_{k=1}^{i-1} A[k],\ \bigwedge_{k=i,i\geq 1}^{n} a[k] == A[k]\} \Rightarrow$$

$$\{\bigwedge_{i=1}^{n}(a[i] == \sum_{k=1}^{n} A[k]\}$$

48. Dados $n \geq 0$, $i \leq n$, demostrar que el siguiente segmento de programa evalúa $\frac{n!}{(i!*(n-i)!)}$:

$$k = 0; fact = 1;$$
$$while\ (k \neq n)\ do$$
$$begin$$
$$k = k+1; fact = fact * k;$$
$$if\ (k \leq i)$$
$$then\ afact = fact;$$
$$if\ (k \leq n-i)$$
$$then\ bfact = fact$$
$$end$$
$$bcof = \frac{fact}{(afact * bfact)}$$

$$k = 0; fact = 1;$$
$$\{k == 0, fact == 1\} \wedge \{i > k \vee afact = i!, n-i > k \vee bfact = (n-i)!, fact = k!\}$$
$$while\ (k \neq n)\ do$$
$$\{i > k \vee afact = i!, n-i > k \vee bfact = (n-i)!, fact = k!\} \wedge \{n \geq 0, i \leq n, k \neq n\}$$
$$begin$$
$$k = k+1;$$
$$\{i > k-1 \vee afact = i!, n-i > k-1 \vee bfact = (n-i)!, fact = (k-1)!\} \wedge$$
$$\{n \geq 0, i \leq n, k \neq n+1\}$$
$$fact = fact * k;$$
$$\{i \geq k \vee afact = i!, (n-i) \geq k \vee bfact = (n-i)!, fact = k!, n \geq 0, i \leq n, k \neq n+1\}$$
$$if\ (k \leq i)$$
$$then$$
$$\{k \leq i, fact = k!\}\ afact = fact\ \{k \leq i, afact == k!\} \wedge \{n \geq 0, i \leq n, k \neq n+1\}$$
$$\{k > i, fact = (i+1)!, afact = i!\} \wedge \{n \geq 0, i \leq n, k \neq n+1\} --alternativa\ else$$
$$\{i > k \vee afact = i!\} \wedge \{n \geq 0, i \leq n, k \neq n+1\}$$
$$if\ (k \leq n-i)$$
$$then$$
$$\{k \leq n-i, fact = k!\}\ bfact = fact\ \{k \leq n-i, bfact == k!\} \wedge \{n \geq 0, i \leq n, k \neq n+1\}$$
$$\{k > n-i, fact = (n-i+1)!, bfact = (n-i)!\} \wedge \{n \geq 0, i \leq n, k \neq n+1\} --else$$
$$\{n-i > k \vee bfact = (n-i)!\} \wedge \{n \geq 0, i \leq n, k \neq n+1\}$$
$$\{k \neq n+1\} \wedge \{((i > k) \vee afact = i!), ((n-i) > k$$
$$\vee bfact = (n-i)!), fact = k!, n \geq 0, i, \leq n\}$$
$$end$$
$$\{k == n\} \wedge \{(afact = i!)), (bfact = (n-i)!), fact = n!, n \geq 0, i \leq n\} \Rightarrow$$
$$bcof = \frac{fact}{(afact * bfact)}$$

49. Demostrar la terminación del fragmento de programa dado en el problema 44 ¿Qué condición se debe imponer para realizar la demostración? La expresión variante del bucle es $\mathrm{n} + 1 - \mathrm{i}$. Inicialmente, el valor de la variable $\mathrm{i} == 1$y, por tanto, la expresión variante se evalúa como n. En cada iteración del bucle,se decrementa la expresión hasta llegar a anularse ($\mathrm{n} + 1 - \mathrm{i} == 0$, entonces la condición del bucle se hace falsa y se termina el bucle con el valor de la variable $\mathrm{i} == \mathrm{n} + 1$

50. Demostrar la terminación del fragmento de programa dado en el Problema 45. Utilizar la noción de conjunto bien fundado para realizar la demostración. La única variable en la condición de iteración del bucle es la variable **i**, el valor de la citada variable **i** describe la secuencia: $\{0,\ 1,\ 2, \ldots, \mathrm{n} - 1$ que se considera "conjunto bien fundado" ya que hace la función de rango $\mathbf{n} - \mathbf{i}$ decrezca en su valor para cada transición de estados (iteración del bucle). En cuanto la secuencia anterior alcance el valor **0** ($\equiv\ i == n$), la condición de iteración del bucle deja de cumplirse y termina el programa.