Evaluación Continua Algoritmos Iterativos

Fecha de Creación	@21 de octubre de 2023 11:03
Asignatura	FAL
Fecha de Modificación	@21 de octubre de 2023 11:06

Pregunta 1
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

¿Cuáles de los siguientes elementos intervienen en la demostración de la **terminación** de un algoritmo iterativo?:

Seleccione una o más de una:

- [ ] a. La precondición.
- [ ] b. La función de tiempo.
- [ ] c. La postcondición.
- [x] d. La expresión de cota. ✔ **Cierto**. La expresión de cota permite demostrar que el algoritmo termina.
- [ ] e. La complejidad asintótica.

a. **Falso**. Forma parte de la especificación que debe satisfacer el algoritmo, y la terminación de un algoritmo no depende de su especificación, sino de su implementación.
b. **Falso**. Tiene que ver con la eficiencia del algoritmo, pero no se usa en la demostración de su terminación.
c. **Falso**. Forma parte de la especificación que debe satisfacer el algoritmo, y la terminación de un algoritmo no depende de su especificación, sino de su implementación.
d. **Cierto**. La expresión de cota permite demostrar que el algoritmo termina.
e. **Falso**. Tiene que ver con la eficiencia del algoritmo, pero no se usa en la demostración de su terminación.

La respuesta correcta es: La expresión de cota.

Pregunta 2
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Considérese el siguiente algoritmo:

```
int algo(int a[], int n) {
    int resul = 0;
    int i = 0;
    int j = n-1;
    while (j >= i) {
        if (a[i] > 0) {
            resul++;
        }
        if (j > i && a[j] > 0) {
            resul++;
        }
        i++;
        j--;
    }
    return resul;
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- [x] a. $resul = \# u : 0 \leq u < i \vee j < u < n : a[u] > 0$ es un invariante del bucle. ✔ **Cierto**. El vector se procesa en *abanico*: en $resul$ estará el número de positivos en los tramos a[0..i) y a(j..n), que es, precisamente, lo que dice este predicado.
- [x] b. La postcondición puede especificarse como $resul = \# i : 0 \leq i < n : a[i] > 0$. ✔ **Cierto**. El algoritmo calcula el número de valores positivos del vector, que es, precisamente, lo que especifica el predicado.
- [x] c. $j - i$ es una expresión de cota del bucle. ✔ **Cierto**. $j - i$ decrece en cada iteración, y nunca baja de -1.
- [ ] d. j es una expresión de cota del bucle.
- [ ] e. $resul = \# u : j < u < n : a[u] > 0$ es un invariante del bucle.

a. **Cierto**. El vector se procesa en *abanico*: en $resul$ estará el número de positivos en los tramos a[0..i) y a(j..n), que es, precisamente, lo que dice este predicado.
b. **Cierto**. El algoritmo calcula el número de valores positivos del vector, que es, precisamente, lo que especifica el predicado.
c. **Cierto**. $j - i$ decrece en cada iteración, y nunca baja de -1.
d. **Falso**. Aunque j disminuye en cada iteración, no puede garantizarse que tenga un mínimo.
e. **Falso**. El vector se procesa en *abanico*: en $resul$ estará el número de positivos en los tramos a[0..i) y a(j..n), y lo que dice este predicado es que en $resul$ está el número de positivos en a(j..n).

Las respuestas correctas son: $resul = \# u : 0 \leq u < i \vee j < u < n : a[u] > 0$ es un invariante del bucle.
, La postcondición puede especificarse como $resul = \# i : 0 \leq i < n : a[i] > 0$.
, $j - i$ es una expresión de cota del bucle.

Pregunta **3**
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Considérese el siguiente algoritmo:

P:$\{0 \le n \le tam(a)\}$

```
int algo(int a[], int n) {
  int resul = 1;
  int i=n-1;
  while (i >= 0) {
    resul = resul * a[i];
    i--;
  }
  return resul;
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- a. $resul = \Pi u : 0 \le u < i : a[u]$ es un invariante del bucle.
- b. $resul = \Pi u : i < u < n : a[u]$ es un invariante del bucle. ✔ **Cierto**. En la iteración $i - esima$, en $resul$ estará acumulado el producto de los $n - i$ últimos elementos del vector, que es lo que especifica este predicado (otra cosa es que sea suficientemente fuerte para demostrar la corrección parcial del algoritmo; a este respecto, falta también indicar el rango de variación de `i`, pero este hecho no quita que el predicado siga siendo un invariante).
- c. La postcondición puede especificarse como $resul = \Pi i : 0 \le i < n : a[i]$. ✔ **Cierto**. El algoritmo calcula el producto de todos los elementos del vector, que es lo que especifica precisamente este predicado.
- d. $n - i$ es una expresión de cota del bucle.
- e. i es una expresión de cota del bucle. ✔ **Cierto**. i nunca será menor que -1 (por tanto, está limitado inferiormente), y, además, decrece en cada iteración.

a. **Falso**. El bucle va acumulando en resul $a[n-1]$, $a[n-1] * a[n-2]$, $a[n-1] * a[n-2] * a[n-3]$ ..., mientras que este predicado lo que dice es que en $resul$ se van acumulando $a[0]$, $a[0] * a[1]$, $a[0] * a[1] * a[2]$...
b. **Cierto**. En la iteración $i - esima$, en $resul$ estará acumulado el producto de los $n - i$ últimos elementos del vector, que es lo que especifica este predicado (otra cosa es que sea suficientemente fuerte para demostrar la corrección parcial del algoritmo; a este respecto, falta también indicar el rango de variación de `i`, pero este hecho no quita que el predicado siga siendo un invariante).
c. **Cierto**. El algoritmo calcula el producto de todos los elementos del vector, que es lo que especifica precisamente este predicado.
d. **Falso**. $n - i$ crece en cada iteración.
e. **Cierto**. i nunca será menor que -1 (por tanto, está limitado inferiormente), y, además, decrece en cada iteración.

Las respuestas correctas son: $resul = \Pi u : i < u < n : a[u]$ es un invariante del bucle.
, La postcondición puede especificarse como $resul = \Pi i : 0 \le i < n : a[i]$.
, i es una expresión de cota del bucle.

Pregunta **4**
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Considérese el siguiente algoritmo:

P:$\{0 \le n \le tam(a)\}$

```
int algo(int a[], int n) {
  int resul = 1;
  int i=0;
  while (i < n) {
    resul = resul * a[i];
    i++;
  }
  return resul;
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- a. $n - i$ es una expresión de cota del bucle. ✔ **Cierto**. $n - i$ nunca será menor que 0, y, además, decrece en cada iteración
- b. $resul = \Pi i : 0 \le i < n : a[i]$ es un invariante del bucle.
- c. i es una expresión de cota del bucle.
- d. $resul = \Pi u : 0 \le u < i : a[u]$ es un invariante del bucle. ✔ **Cierto**. En la iteración $i - esima$, en $resul$ estará acumulado el producto de los i primeros elementos del vector, que es lo que especifica este predicado (otra cosa es que sea suficientemente fuerte para demostrar la corrección parcial del algoritmo; a este respecto, falta también indicar el rango de variación de i, pero este hecho no quita que el predicado siga siendo un invariante).
- e. La postcondición puede especificarse como $resul = \Pi i : 0 \le i < n : a[i]$. ✔ **Cierto**. El algoritmo calcula el producto de todos los elementos del vector, que es lo que especifica precisamente este predicado.

a. **Cierto**. $n - i$ nunca será menor que 0, y, además, decrece en cada iteración
b. **Falso**. El bucle va acumulando en resul $a[0]$, $a[0] * a[1]$, $a[0] * a[1] * a[2]$ ..., mientras que este predicado lo que dice es que $resul$ siempre es igual al producto de todos los elementos del vector.
c. **Falso**. i crece en cada iteración, mientras que una expresión de cota debe decrecer en cada iteración.
d. **Cierto**. En la iteración $i - esima$, en $resul$ estará acumulado el producto de los i primeros elementos del vector, que es lo que especifica este predicado (otra cosa es que sea suficientemente fuerte para demostrar la corrección parcial del algoritmo; a este respecto, falta también indicar el rango de variación de i, pero este hecho no quita que el predicado siga siendo un invariante).
e. **Cierto**. El algoritmo calcula el producto de todos los elementos del vector, que es lo que especifica precisamente este predicado.

Las respuestas correctas son: $n - i$ es una expresión de cota del bucle.
, $resul = \Pi u : 0 \le u < i : a[u]$ es un invariante del bucle.
, La postcondición puede especificarse como $resul = \Pi i : 0 \le i < n : a[i]$.

Pregunta **5**
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Considérese el siguiente algoritmo:

```
int algo(int a[], int n, int v) {
    int resul = 0;
    int i = n-1;
    while (i >= 0) {
        if (a[i] == v) {
            resul++;
        }
        i--;
    }
    return resul;
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- a. $n - i$ es una expresión de cota del bucle.
- b. $resul = \# u : 0 \leq u < i : a[u] = v$ es un invariante del bucle.
- c. $resul = \# u : i < u < n : a[u] = v$ es un invariante del bucle. ✔ **Cierto**. El vector se va recorriendo desde el final hasta el principio: tras la i-esima interación, el problema se habrá resuelto para el tramo *a(i..n)*.
- d. La postcondición puede especificarse como $resul = \# i : 0 \leq i < n : a[i] = v$. ✔ **Cierto**. El algoritmo calcula el número de ocurrencias de v en el vector, que es lo que especifica precisamente este predicado.
- e. i es una expresión de cota del bucle. ✔ **Cierto**. i nunca será menor que -1, y, además, decrece en cada iteración.

a. **Falso**. $n - i$ crece en cada iteración, mientras que una expresión de cota debe decrecer en cada iteración.
b. **Falso**. El vector se va recorriendo desde el final hasta el principio: tras la i-esima interación, el problema se habrá resuelto para el tramo *a(i..n)*, en lugar de para el tramo *a[0..i)*, como afirma este predicado.
c. **Cierto**. El vector se va recorriendo desde el final hasta el principio: tras la i-esima interación, el problema se habrá resuelto para el tramo *a(i..n)*.
d. **Cierto**. El algoritmo calcula el número de ocurrencias de v en el vector, que es lo que especifica precisamente este predicado.
e. **Cierto**. i nunca será menor que -1, y, además, decrece en cada iteración.

Las respuestas correctas son: $resul = \# u : i < u < n : a[u] = v$ es un invariante del bucle.
, La postcondición puede especificarse como $resul = \# i : 0 \leq i < n : a[i] = v$.
, i es una expresión de cota del bucle.

Pregunta **6**
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Considérese el siguiente algoritmo:

P:$\{0 < n \leq tam(a)\}$

```
int algo(int a[], int n) {
  int resul = a[n-1];
  int i=n-2;
  while (i >= 0) {
    if(resul < a[i])
       resul = a[i];
    i--;
  }
  return resul;
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- a. La postcondición puede especificarse como $resul = max\ i : 0 \leq i < n : a[i]$. ✔ **Cierto**. El algoritmo calcula el máximo de los elementos del vector, que es lo que especifica precisamente este predicado.
- b. $resul = max\ u : 0 \leq u < i : a[u]$ es un invariante del bucle.
- c. i es una expresión de cota del bucle. ✔ **Cierto**. i decrece en cada iteración, y su valor nunca cae por debajo de -1.
- d. $n - i$ es una expresión de cota del bucle.
- e. $resul = max\ u : i < u < n : a[u]$ es un invariante del bucle. ✔ **Cierto**. En cada iteración, $resul$ es el máximo de $a[i+1]$ ... $a[n-1]$, que es lo que expresa este predicado.

a. **Cierto**. El algoritmo calcula el máximo de los elementos del vector, que es lo que especifica precisamente este predicado.
b. **Falso**. En cada iteración, $resul$ es el máximo de $a[i+1]$ ... $a[n-1]$, no el máximo de $a[0]$ ... $a[i-1]$, que es lo que expresa este predicado.
c. **Cierto**. i decrece en cada iteración, y su valor nunca cae por debajo de -1.
d. **Falso**. Como i decrece en cada iteración, $n - i$ crecerá, en lugar de decrecer.
e. **Cierto**. En cada iteración, $resul$ es el máximo de $a[i+1]$ ... $a[n-1]$, que es lo que expresa este predicado.

Las respuestas correctas son: La postcondición puede especificarse como $resul = max\ i : 0 \leq i < n : a[i]$.
, i es una expresión de cota del bucle.
, $resul = max\ u : i < u < n : a[u]$ es un invariante del bucle.

Pregunta **7**
Parcialmente correcta
Se puntúa 0,17 sobre 1,00
Marcar pregunta

¿Cuáles de los siguientes pasos pueden formar parte de la demostración de la **corrección parcial** de un algoritmo iterativo?:

Seleccione una o más de una:

- a. Demostrar que la negación de la condición del bucle y el invariante implican la postcondición.
- b. Demostrar que la condición del bucle y el invariante implican la postcondición. ✘ **Falso**. Es la negación del bucle la que debe considerarse.
- c. Demostrar que el bucle se ejecuta al menos una vez.
- d. Demostrar que el invariante se cumple justo antes del comienzo del bucle. ✔ **Cierto**. El invariante debe cumplirse justo antes de que el bucle comience a ejecutarse, y también tras cada iteración.
- e. Demostrar que la expresión de cota decrece en cada iteración.

a. **Cierto**. Una vez que termina el bucle su condición será falsa, y seguirá cumpliéndose su invariante. Debe, entonces, comprobarse si es posible satisfacer la postcondición.
b. **Falso**. Es la negación del bucle la que debe considerarse.
c. **Falso**. Puede haber estados para los que el bucle no se ejecuta.
d. **Cierto**. El invariante debe cumplirse justo antes de que el bucle comience a ejecutarse, y también tras cada iteración.
e. **Falso**. La expresión de cota se utiliza para demostrar la terminación, no la corrección parcial.

Las respuestas correctas son: Demostrar que la negación de la condición del bucle y el invariante implican la postcondición., Demostrar que el invariante se cumple justo antes del comienzo del bucle.

Pregunta **8**
Incorrecta
Se puntúa 0,00 sobre 1,00
Marcar pregunta

Supón que estás demostrando la *corrección parcial* de un algoritmo iterativo que tiene la siguiente estructura:

```
<código de inicialización>
while(<condición del bucle>) {
  <cuerpo del bucle>
}
```

¿Qué debes comprobar?

Seleccione una o más de una:

- a. Que el algoritmo termina.
- b. Que el invariante se cumple justo antes de la primera iteración. ✔ **Cierto**. Es una condición necesaria para que el invariante sea, efectivamente, un invariante.
- c. Que la negación de la condición del bucle y el invariante implican la postcondición.
- d. Que la precondición implica el invariante. ✘ **Falso**. La precondición no tiene porque implicar directamente el invariante: para ello está el código de inicialización.
- e. Que la condición del bucle y el invariante implican la postcondición. ✘ **Falso**. Cuando se sale del bucle lo que se verifica es la negación de su condición, no la condición en sí.

a. **Falso**. Cuando demostramos la corrección parcial no nos preocupa si el algoritmo termina o no; únicamente nos preocupa que, cuando termine, se satisfaga la postcondición.
b. **Cierto**. Es una condición necesaria para que el invariante sea, efectivamente, un invariante.
c. **Cierto**. En este caso, cuando salimos del bucle, el algoritmo termina. En este momento se satisface la negación de la condición del bucle y el invariante. Esto debe ser suficiente para garantizar la postcondición.
d. **Falso**. La precondición no tiene porque implicar directamente el invariante: para ello está el código de inicialización.
e. **Falso**. Cuando se sale del bucle lo que se verifica es la negación de su condición, no la condición en sí.

Las respuestas correctas son: Que el invariante se cumple justo antes de la primera iteración., Que la negación de la condición del bucle y el invariante implican la postcondición.

Pregunta **9**
Parcialmente correcta
Se puntúa 0,75 sobre 1,00
Marcar pregunta

¿Cuáles de los siguientes elementos intervienen en la demostración de la **corrección parcial** de un algoritmo iterativo?:

Seleccione una o más de una:

- a. La precondición. ✔ **Cierto**. Forma parte de la especificación que debe satisfacer el algoritmo.
- b. La condición del bucle.
- c. La postcondición. ✔ **Cierto**. Forma parte de la especificación que debe satisfacer el algoritmo.
- d. La expresión de cota.
- e. El invariante del bucle. ✔ **Cierto**. Permite caracterizar qué propiedades cumplen los estados por los que va transitando el proceso iterativo.

a. **Cierto**. Forma parte de la especificación que debe satisfacer el algoritmo.
b. **Cierto**. Se utiliza tanto para demostrar que, una vez que se sale del bucle, es posible alcanzar la postcondición, como para razonar que el cuerpo del bucle mantiene el invariante.
c. **Cierto**. Forma parte de la especificación que debe satisfacer el algoritmo.
d. **Falso**. La expresión de cota permite demostrar que el algoritmo termina (la corrección parcial no se ocupa de la terminación).
e. **Cierto**. Permite caracterizar qué propiedades cumplen los estados por los que va transitando el proceso iterativo.

Las respuestas correctas son: La precondición., La condición del bucle., La postcondición., El invariante del bucle.

Pregunta **10**

Parcialmente correcta

Se puntúa 0,17 sobre 1,00

Marcar pregunta

Supón que estás demostrando la *corrección total* de un algoritmo iterativo que tiene la siguiente estructura:

```
<código de inicialización>
while(<condición del bucle>) {
  <cuerpo del bucle>
}
```

¿Qué debes comprobar?

Seleccione una o más de una:

- [ ] a. Que el invariante se satisface en cualquier estado al que se llega inmediatamente antes de comenzar a ejecutar el bucle.
- [x] b. Que el invariante se satisface en cualquier estado que satisface a la precondición. ✖ **Falso**. Debe ejecutarse el código de inicialización.
- [ ] c. Que, en cualquier estado en el que se satisface el invariante y la condición del bucle, se satisface también la postcondición.
- [x] d. Que, si comienza en cualquier estado que satisface la precondición, entonces el algoritmo termina. ✔ **Cierto**. Porque se está demostrando la corrección total. Entonces, si, al comienzo de la ejecución, se satisface la precondición, el algoritmo debe necesariamente terminar en un estado que satisface la postcondición.
- [x] e. Que, si se supone cierto el invariante, y falsa la condición del bucle, entonces puede derivarse la postcondición. ✔ **Cierto**. En este momento se sale del bucle, por lo que será necesario demostrar que se cumple la postcondición.

a. **Cierto**. Porque el invariante debe satisfacerse inmediatamente antes de ejecutar el bucle.
b. **Falso**. Debe ejecutarse el código de inicialización.
c. **Falso**. Debe haberse falsificado la condición, para poder salir del bucle.
d. **Cierto**. Porque se está demostrando la corrección total. Entonces, si, al comienzo de la ejecución, se satisface la precondición, el algoritmo debe necesariamente terminar en un estado que satisface la postcondición.
e. **Cierto**. En este momento se sale del bucle, por lo que será necesario demostrar que se cumple la postcondición.

Las respuestas correctas son: Que el invariante se satisface en cualquier estado al que se llega inmediatamente antes de comenzar a ejecutar el bucle., Que, si comienza en cualquier estado que satisface la precondición, entonces el algoritmo termina., Que, si se supone cierto el invariante, y falsa la condición del bucle, entonces puede derivarse la postcondición.