Tecnología Digital 1: Introducción a la Programación - TP1

Autores: de Nuñez, Federico
Yapoudjian, Tadeo
Loza Montaña, Gastón

Justificación de por qué los programas hacen lo esperado

Función misma_paridad:

La función misma_paridad determina si dos números tienen la misma paridad. Recibe 2 argumentos de tipo entero n y m. Esta función le asigna a la variable vr (de tipo bool) el valor de verdad que tiene la expresión:

es_par(n) and es_par(m) or not es_par(n) and not es_par(m)

La expresión utiliza la función es_par para determinar si los argumentos son pares o impares, y de acuerdo a eso, con los operadores lógicos correspondientes se determina si ambos argumentos coinciden en su paridad.

Tabla de verdad de la expresión para cada caso:

p: "n es par"

q: "m es par"

Buscamos: p and q or not p and not q

p	q	p and q	not p	not q	not p and not q	p and q or not p and not q
True	True	True	False	False	False	True
True	False	False	False	True	False	False
False	True	False	True	False	False	False
False	False	False	True	True	True	True

Función alterna_paridad:

La función alterna_paridad determina si los dígitos de un número n alternan su paridad.

Se le otorga un argumento n a la función. Se evaluará cada uno de los dígitos de n (desde el segundo) para ver si cada uno de ellos comparte o no la paridad con su antecesor.

<u>Terminación del ciclo:</u>

- La variable i empieza valiendo 1.
- En cada ejecución del cuerpo del ciclo, i se incrementa en 1.
- La cantidad de dígitos de n (len(string_n)) es siempre la misma.
- Es inevitable que en algún momento, i iguale el valor de len(string_n)

- En ese momento, la condición i < len(string_n) será falsa, y el ciclo terminará.

<u>Correctitud del ciclo:</u>

Predicado invariante del ciclo: 1 <= i <= len(string_n); vr equivale a que los dígitos de n alternan su paridad hasta el análisis en la posición i. El predicado invariante vale en (A), donde i = 1 y vr = True.

Supongamos que el predicado invariante vale en (B). Si llamamos K al valor de i en (B), entonces vr equivale a que los dígitos alternan la paridad en n antes de la posición K.

En la primera asignación del cuerpo del bucle, vr agrega a su valor mediante el operador lógico *and* la evaluación de la función misma_paridad con el dígito de la posición K y el dígito de la posición K-1 como argumentos. Luego, vr equivale a que los dígitos de n alternan su paridad antes de la posición K + 1.

En la segunda asignación del cuerpo del bucle, i se incrementa en 1, es decir que pasa a valer K + 1.

En (C) nuevamente vr equivale a que los dígitos de n alternan su paridad antes de la posición de i.

Además, en (B) 1 <= i <= len(string_n), por lo que luego de sumarle 1 a i, en (C) podemos afirmar que 1 <= i <= len(string_n). Por lo que el predicado invariante sigue siendo válido.

Por último, en (D) el predicado invariante es válido y sabemos que i es igual a len(string_n), podemos entonces afirmar que vr equivale a que los dígitos de n alternan su paridad antes de la posición len(string_n) (es decir, en en todos los dígitos de n). Y se cumple la postcondición.

Función es_peculiar:

La función es_peculiar determina si n es lo que se considera un número peculiar.

Se le otorga un argumento n (que pertenece al conjunto de naturales más el 0) a la función. Ésta le asigna a la variable vr (de tipo bool) el valor de verdad que tiene la expresión:

multiplo_de_22(n) == 0 and alterna_paridad(n)

La expresión utiliza la función multiplo_de_22 para determinar si n es múltiplo de 22 y alterna_paridad para determinar si los dígitos de n alternan su paridad. De acuerdo a eso, con el operador lógico and se evalua si ambas condiciones se cumplen para saber si n es un número peculiar.

Tabla de verdad de la expresión para cada caso:

p: “n es múltiplo de 22”
q: “los dígitos de n alternan su paridad”
Buscamos: p and q

p	q	p and q
True	True	True
True	False	False
False	True	False
False	False	False

Donde True quiere decir que n es número peculiar, y False que no lo es.

Función n_esimo_peculiar:

La función n_ésimo_peculiar determina que número peculiar es el n_ésimo propuesto.

Se le otorga un argumento n (que pertenece al conjunto de naturales más el 0) a la función y esta determinará el número peculiar número n.

<u>Terminación del ciclo:</u>

- Las variables i y j empiezan valiendo 0.
- n es el argumento dado que se mantiene igual
- En cada ejecución del cuerpo del ciclo, j se incrementa en 1
- Si j es peculiar, i se incrementa en 1. De lo contrario se repite el ciclo.
- Eventualmente i llegará al valor de n
- En ese momento la condición i < n será falsa, y el ciclo terminará.

<u>Correctitud del ciclo:</u>

Predicado invariante del ciclo: 0 <= i <= n, vr equivale al i-ésimo peculiar.

El predicado invariante vale en (A), donde i = 0 y vr es el i-ésimo peculiar, es decir vr = 0.

Supongamos que el predicado invariante vale en (B). vr equivale al i_ésimo numéro peculiar.

En la primera asignación, j se incrementa en 1. En la segunda asignación, se evalúa si j es un número peculiar. Si es el caso, entonces i se incrementa en 1, de lo contrario se repite el ciclo.

En (C), nuevamente vr es el i_ésimo peculiar.

Además, en (B) 0 <= i <= n, por lo que luego de sumarle 1 a i, en (C) podemos afirmar que 0 <= i <= n. Por lo que el predicado invariante sigue siendo válido.

En (D), vale el predicado invariante porque cuando i es igual a n, vr es el n_ésimo_peculiar.

Función cant_peculiares_entre:

Determina la cantidad de números peculiares entre dos argumentos n y m.

Se le otorga dos argumentos n y m a la función y ésta evalúa la cantidad de números peculiares en el intervalo cerrado [n, m].

<u>Terminación del ciclo:</u>

- i comienza valiendo el extremo izquierdo (n) del intervalo en el cual queremos hallar cantidad de números peculiares
- m es el extremo derecho que siempre se mantiene igual.
- En cada ejecución del cuerpo del ciclo, i se incrementa de 1.
- Es inevitable que eventualmente i llegue al valor de m.
- En ese momento la condición i <= m será falsa, y el ciclo terminará.

Correctitud del ciclo:

Predicado invariante del ciclo: n <= i <= m y vr equivale a la cantidad de números peculiares entre n e i.

El predicado invariante se cumple en (A), donde i = n y vr = 0.

Supongamos que el predicado invariante vale en (B). En ese caso, vr equivale a la cantidad de números peculiares entre n e i.

En la primera asignación, se evalúa si la variable i es un número peculiar, si así lo fuere, se le suma 1 al valor de vr e i se incrementa en 1. De lo contrario, solo incrementa i en 1.

En (C), vr es la cantidad de números peculiares entre n e i. Se mantiene la validez del predicado invariante.

Además, en (B) n <= i <= m, por lo que luego de sumarle 1 a i, en (C) podemos afirmar que n <= i <= m. Por lo que el predicado invariante sigue siendo válido.

Por último, en (D) vale el predicado invariante y sabemos que i = m. Entonces, el ciclo termina y podemos afirmar que vr equivale a la cantidad de números peculiares entre n y m, que es lo que requiere la postcondición.

Aclaraciones adicionales (si las hubiera)

Función es_par:

La función es_par es una función que añadimos para facilitar el proceso de verificación de la paridad de un número.

Se le otorga un argumento n (que pertenece al conjunto de enteros) a la función y esta determina si el resto de la división del argumento por 2 es 0. En cuyo caso, el argumento es par.

Función multiplo_de_22:

La función multiplo_de_22 es una función que añadimos que evalúa si un argumento es múltiplo de 22.

Se le otorga un argumento n (que pertenece al conjunto de enteros) a la función y esta determina si el resto de la división del argumento por 22 es 0. En cuyo caso, el argumento es un múltiplo de 22.