

EJERCICIOS 3.4 , 3.5 , 3.6

//Guillermo Escobar P cod 256243

3.4

Consider the problem of making Robin the robot back up one tile. The precise preconditions and postconditions for this problem are as follows:

Preconditions:

There are no obstructions within two tiles of Robin, in any direction.

Postconditions:

Robin is one tile behind (relative to its original direction) its original position.

Robin is facing in its original direction.

Prove that each of the following algorithms correctly solves this problem:

Algorithm 1

```
robin. turnLeft();  
robin. turnLeft();  
robin. move();  
robin. turnLeft();  
robin. turnLeft();
```

Robin deberá estar al final mirando hacia la misma dirección que la original, pero un paso atrás. Vemos que al terminar el algoritmo Robin ha mirado primero hacia su izquierda dos veces, luego queda mirando hacia la dirección opuesta que la original. Retrocede un paso, y gira 180°, así queda mirando a la misma dirección original, pero un paso atrás, cumpliéndose así las poscondiciones dadas.

Algorithm 2

```
robin. turnLeft();  
robin. move();  
robin. turnLeft();  
robin. move();  
robin. turnLeft();  
robin. move();  
robin. turnLeft();
```

Robin deberá estar al final mirando hacia la misma dirección que la original, pero un paso atrás.. Para este algoritmo Robin inicia girando hacia la izquierda y da un paso. Seguidamente lo repite dos veces, así queda en una posición atrás de donde inicio. Finalmente gira para quedar mirando hacia la misma dirección original, cumpliéndose así las postcondiciones dadas.

Algorithm 3

```
robin. turnRight();  
robin. turnRight();  
robin. move();  
robin. turnLeft();  
robin. turnLeft();
```

Robin deberá estar al final mirando hacia la misma dirección que la original, pero un paso atrás. En el algoritmo 3 pasa igual que el algoritmo 1, solo que en los dos primeros pasos gira a la derecha en vez de girar hacia la izquierda. De

todas formas queda mirando hacia la dirección contraria a la inicial , aquí también se cumplen las postcondiciones.

3.5. Explain why each of the following is or is not a valid use of modus ponens:

1. Birds have wings. Having wings implies being able to fly. Therefore, birds are able to fly.

2. Sue always wears sunglasses when driving. Sue is driving now. Therefore, Sue is wearing sunglasses now.

3. Dogs do not have wings. Having wings implies being able to fly. Therefore, dogs can't fly.

4. Birds fly. Having wings implies being able to fly. Therefore, birds have wings.

5. If A and B are both positive, and $A < B$, then $\frac{1}{A} > \frac{1}{B}$. $2 < 2.5$, and $2.5 < 3$. Therefore, $\frac{1}{2.5}$ lies between $\frac{1}{2}$ and $\frac{1}{3}$.

6. Any composite number is the product of two or more prime numbers. 18 is composite. Therefore, 18 is the product of two or more prime numbers.

7. Any composite number is the product of two or more prime numbers. 18 is composite. Therefore, $18 = 3 \times 3 \times 2$.

8. If a baseball player drops the ball, then he or she is a bad baseball player. I once saw a college baseball player drop the ball. Therefore, all college baseball players are bad baseball players.

9 Any person who hand-feeds piranhas is crazy. Therefore, this book's first author is crazy.

Las preposiciones [1, 2, 3, 6, 7] utilizan correctamente el modus ponens
En cambio en las preposiciones [4, 5, 8,9] no se utiliza correctamente el modus ponens

3.6 Which steps in the correctness proof for the secondsSinceMidnight method would be invalid if hour or minute included a fractional part? What if hour were greater than 23, or minute or second greater than 59?

PROOF: Since $0 \leq \text{hour} \leq 23$, and hour contains no fractional part, the expression **hour * 60 computes the number of minutes between midnight and time hour:0:0**. Since minute also contains no fractional part and $0 \leq \text{minute} \leq 59$, **hour * 60 + minute is then the number of minutes between midnight and time hour:minute:0**. Since there are 60 seconds in a minute, multiplying that number by 60 yields the number of seconds between midnight and time hour:minute:0. Finally, since $0 \leq \text{second} \leq 59$, **adding second gives the number of seconds between midnight and time hour:minute:second**. The algorithm returns this final value.

Las partes de texto en color rojo no se podrían llevar a cabo si minutos u horas se tomaran como fracciones. Si hubiera

fracciones en las horas. Se sumaran a la unidad de tiempo mas pequeña siguiente, es decir minutos y segundos. Asi la parte en azul , la cual representa la postcondicion nunca cumpliría.

Para el caso de que los minutos fueran mayor que 59, entonces se sumaria sumar una hora. Si las horas fuesen mayores que 23 entonces el se contaría hasta el otro día .