# Cálculo de Programas y Desarrollo de Invariantes

**Programación Avanzada**

Pablo F. Castro

# Desarrollando Ciclos

Suponiendo que tenemos el invariante, la postcondición y el variante:

Precondición

$\{P\}$

Inicialización

$S'$

Invariante

$\{I\}$

$\{v : t\}$

Cota o Variante

$do\ B \rightarrow decrementar\ t,\ reestablecer\ Inv\ od$

$\{\neg B \wedge Inv\}$

Cuerpo

$\{Q\}$

Postcondición

1. Encontrar S' tal que:

$$\{P\}S'\{I\}$$

2. Encontrar B, tal que:

$$\neg B \wedge Inv \Rightarrow Q$$

3. Elegir cota t tal que:

$$I \wedge t \leq 0 \Rightarrow \neg B$$

4. Derivar S tal que:

$$\{I \wedge B\}S\{I\}$$

y

$$\{I \wedge B \wedge t = A\}S\{t < A\}$$

# Un Ejemplo

Supongamos la siguiente especificación:

$$\{P : x = X \wedge y = Y \wedge X > 0 \wedge Y > 0\}$$
$$S$$
$$\{Q : x = mcd.X.Y\}$$

mcd cumple:

Usaremos estas propiedades para derivar el programa

$$1. mcd.x.x = x$$
$$2. mcd.x.y = mcd.y.x$$
$$3. x > y \Rightarrow mcd.x.y = mcd.(x - y).y$$
$$4. x < y \Rightarrow mcd.x.y = mcd.x.(y - x)$$

# Encontrando el Invariante

La idea es que x e y posean el mismo mcd que X e Y

$$\{I : x > 0 \wedge y > 0 \wedge mcd.x.y = mcd.X.Y\}$$

No hay inicialización, tenemos que demostrar:

$$P \Rightarrow I$$

Directo usando Leibniz

Tratemos de encontrar B tal que: $I \wedge \neg B \Rightarrow Q$

Un buen candidato es:
x=y

# Derivando el Cuerpo

Es decir: $B \equiv x \neq y$ que podemos dividirlo en dos casos: $x < y \vee x > y$, en el primer caso:

$$I \wedge x < y$$
$$\equiv [\text{def. I}]$$
$$mcd.x.y = mcd.X.Y \wedge x > 0 \wedge y > 0 \wedge x < y$$
$$\Rightarrow [\text{Prop. mcd}]$$
$$mcd.x.(y - x) = mcd.X.Y \wedge x > 0 \wedge (y - x) > 0$$
$$\equiv [\text{substitución}]$$
$$I[y := y - x]$$
$$\equiv$$
$$wp.(y := y - x).I$$

Es decir:

$$\{I \wedge x < y\} y := y - x \{I\}$$

De la misma forma:

$$\{I \wedge x > y\} x := x - y \{I\}$$

# Ejemplo (cont.)

Es decir, obtenemos:

$$
\begin{array}{l}
do\ x < y \rightarrow y := y - x \\
\Box x > y \rightarrow x := x - y \\
od
\end{array}
$$

Falta determinar la cota o variante!

Un buen candidato es: $t = x + y$, demostremos:

$$1. I \wedge (x < y \vee x > y) \rightarrow x + y > 0$$

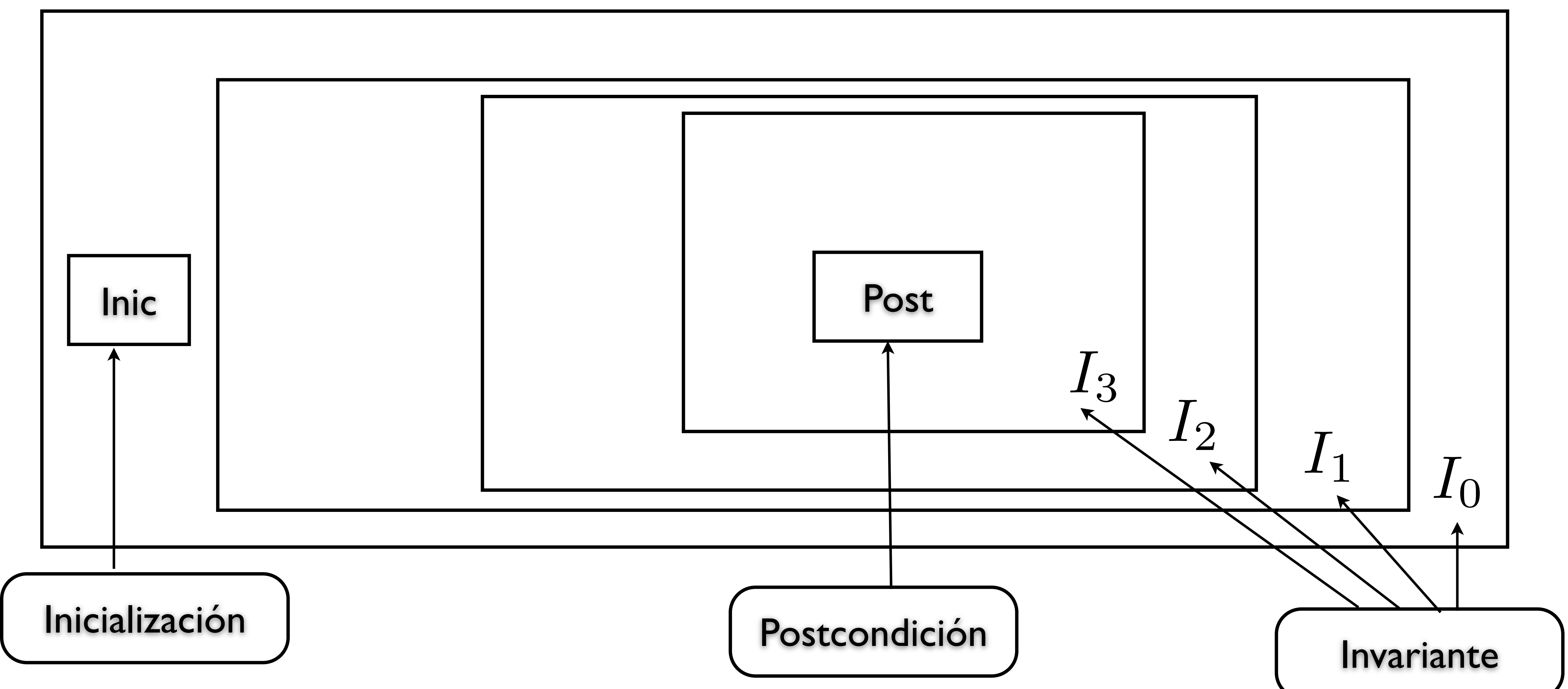
Por invariante: $x > 0$ e $y > 0$

$$2a. \{I \wedge x < y \wedge x + y = A\} y := y - x \{x + y < A\}$$

$$2b. \{I \wedge x > y \wedge x + y = A\} x := x - y \{x + y < A\}$$

# Técnicas de Desarrollo de Invariantes

Podemos graficar la noción de *invariante* de la siguiente forma



# Debilitando la postcondición

En general, podemos obtener el invariante si debilitamos la postcondición:

- Tomar partes de una conjunción:

$$A \wedge B \wedge C \Rightarrow A \wedge B$$

- Reemplazar constante por variable:

$$x \leq 10 \Rightarrow x \leq i \text{ en donde: } i \leq 10$$

- Agregar una diyunción: (generalmente no se usa)

$$P \Rightarrow P \vee Q$$

# Tomar términos de una conjunción

Supongamos la siguiente especificación:

$\{n \geq 0\}$

$S$

$\{0 \leq a \wedge a^2 \leq n \wedge n < (a+1)^2\}$

Aproxima la raíz cuadrada de n

Podemos proponer: $Inv = 0 \leq a \wedge a^2 \leq n$

Debido a la condición II, tenemos:

$$B \equiv n \geq (a+1)^2$$

# Ejemplo (cont)

Es decir, obtenemos:

$$
\begin{array}{l}
a := 0; \\
do\ n \geq (a+1)^2 \rightarrow S \\
od
\end{array}
$$

Derivemos S:

$$
\begin{array}{l}
I \wedge (a+1)^2 \leq n \\
\equiv [\text{def.inv}] \\
0 \leq a \wedge a^2 \leq n \wedge (a+1)^2 \leq n \\
\Rightarrow [\text{Aritmetica}] \\
0 \leq a+1 \wedge (a+1)^2 \leq n \\
\equiv [\text{def wp}] \\
wp.(a := a+1).I
\end{array}
$$

Entonces:

$$
\begin{array}{l}
a := 0; \\
do\ n \geq (a+1)^2 \rightarrow a := a+1 \\
od
\end{array}
$$

# Ejemplo (cont.)

Necesitamos encontrar una cota:

$$t = \sqrt{n} - a$$

Debemos probar:

$i.\ 0 \leq a \wedge (a+1)^2 \leq n \Rightarrow \sqrt{n} - a > 0$

$ii. \{0 \leq a \wedge (a+1)^2 \leq n \wedge \sqrt{n} - a = A\} a := a + 1 \{\sqrt{n} - a < A\}$

El tiempo de ejecución es proporcional a:

$$\sqrt{n}$$

# Cambiando Constantes por Variables

Consideremos de nuevo el problema de la raíz cuadrada:

$$\{P : n \geq 0\}$$
$$S$$
$$\{0 \leq a \wedge a^2 \leq n < (a+1)^2\}$$

Podemos reemplazar a+1 por una variable:

$$\{a^2 \leq n < b^2\}$$

Una cota para b puede ser n+1, el invariante es:

$$\{I : (0 \leq a) \wedge (a < b \leq n+1) \wedge (a^2 \leq n < b^2)\}$$

# Ejemplo (cont)

- La asignación: $a, b := 0, n+1;$ hace true el invariante, es un buen candidato para inicialización
- Para la guarda se debe cumplir: $I \wedge \neg B \Rightarrow Q$ , es decir: $B \equiv a+1 \neq b$

Es decir, obtenemos:

$$
\begin{aligned}
& a, b := 0, n+1; \\
& do\ b \neq a+1 \rightarrow S \\
& od
\end{aligned}
$$

Falta encontrar S

# Ejemplo (cont.)

- El ciclo terminará cuando *a+1=b*, es decir, cada paso del ciclo debe achicar *b-a*.
- Para achicar *b-a*, podemos mover *a* ó *b*.
- Para hacer esto podemos mover a o b al medio: (a+b)/2

$$
\begin{array}{l}
do\ a+1 \neq b \wedge B_0 \rightarrow a := (a+b)/2 \\
\Box\ a+1 \neq b \wedge B_1 \rightarrow b := (a+b)/2 \\
od
\end{array}
$$

Falta averiguar las guardas

# Ejemplo (cont)

Cada rama debe hacer true el invariante, podemos usar esto para calcular las guardas.

$$I \wedge (a+1) \neq b \wedge B_0 \rightarrow wp.a := (a+b)/2.(a \leq b \leq n+1 \wedge a^2 \leq n < b^2)$$
$$\equiv [\text{def.wp}]$$
$$I \wedge (a+1) \neq b \wedge B_0 \rightarrow (a+b)/2 \leq b \leq n+1 \wedge ((a+b)/2)^2 \leq n < b^2)$$

Un candidato es: $((a+b)/2)^2 \leq n$

De la misma forma la otra guarda es: $((a+b)/2)^2 > n$

# Ejemplo (cont)

Es decir, nos queda:

la cota es: b-a-1

$$a, b := 0, n+1;$$

$$do\ (a+1) \neq b \wedge ((a+b)/2)^2 \leq n \rightarrow a := (a+b)/2$$

$$\Box\ (a+1) \neq b \wedge ((a+b)/2)^2 > n \rightarrow b := (a+b)/2$$

$$od$$

Es una búsqueda dicotómica. Es más rápido que el anterior.

# Ejemplo:División Entera

Supongamos la siguiente especificación:

$$\{P : x \geq 0 \wedge y > 0\}$$

$$S$$

$$\{Q : x = q * y + r \wedge 0 \leq r \wedge r < y\}$$

Calcula el cociente y el resto de la división

siempre debería valer durante el

siempre debería valer durante el ciclo

Solo vale al final de la división

Es decir podemos tomar las dos primeras partes como invariante, agregandole que y>0

# Ejemplo (cont)

Es decir el invariante es: $\{I : x = q * y + r \wedge 0 \leq r \wedge y > 0\}$

Entonces la guarda es: $r \geq y$

Para la inicialización se debe cumplir:

$$\{x \geq 0 \wedge y > 0\} q, r := F, E \{x = q * y + r \wedge 0 \leq r\}$$
$$\equiv [\text{prop.wp}]$$
$$x \geq 0 \wedge y > 0 \Rightarrow wp.(q, r := F, E).(x = q * y + r \wedge 0 \leq r)$$
$$\equiv [\text{def.wp}]$$
$$x \geq 0 \wedge y > 0 \Rightarrow x = F * y + E \wedge 0 \leq E$$
$$\Leftarrow [\text{Arit.}]$$
$$F = 0 \wedge E = x$$

Es decir, la inicialización es:

$$q, r := 0, x$$

# Ejemplo (cont.)

Hasta ahora sabemos:

$$
\begin{aligned}
&q, r := 0, x \\
&do\ r \geq y \to S \\
&od
\end{aligned}
$$

Para la cota sabemos que r tiene que decrecer. Se tiene que cumplir:

$$
\begin{aligned}
&x = q * y + r \wedge 0 \leq r \wedge r \geq y \Rightarrow wp.(q, r := E, r - F).(x = q * y + r \wedge 0 \leq r) \\
&\equiv [\text{def. wp}] \\
&x = q * y + r \wedge 0 \leq r \wedge r \geq y \Rightarrow (x = E * y + r - F \wedge 0 \leq r - F) \\
&\equiv [\text{Arit.}] \\
&x = q * y + r \wedge 0 \leq r \wedge r \geq y \Rightarrow (q * y + r = E * y + r - F \wedge 0 \leq r - F) \\
&\equiv [\text{Arit.}] \\
&x = q * y + r \wedge 0 \leq r \wedge r \geq y \Rightarrow (E = q + F/y \wedge F \leq r) \\
&\Leftarrow [Arit.] \\
&E = q + 1 \wedge F = y
\end{aligned}
$$

# Ejemplo (cont.)

Es decir nos queda:

$$\{P : x \geq 0 \wedge y > 0\}$$
$$q, r := 0, x;$$
$$do\ r \geq y \rightarrow q, r := q + 1, r - y;$$
$$od$$
$$\{Q : x = q * y + r \wedge 0 \leq r \wedge r < y\}$$

Aquí hemos usado la técnica de tomar partes de una conjunción

En la precondición, invariante y postcondición podemos agregar:

$$x = X \wedge y = Y$$

Para asegurar que los valores de x e y no se pierden

# Un ejemplo más divertido: Bandera Holandesa

Supongamos que tenemos un arreglo, con tres posibles valores: r (rojo), b (blanco), a (azul):

[r,b,a,r,r,a,a]

Queremos ordenarlo como la bandera Holandesa:

[r,r,r,b,a,a,a]

Primero los rojos, después los blancos y últimos los azules

# Especificación

La especificación es:

$$Array\ a[0,N)\ of\ \{r,w,b\}$$

$$\{R = \text{cant rojos} \wedge W = \text{cant. blancos}\}$$

$$S$$

$$\{$$

$$\langle \forall i : 0 \leq i < R : a.i = r \rangle$$

$$\langle \forall i : R \leq i < W + R : a.i = w \rangle$$

$$\langle \forall i : W + R \leq i < N : a.i = b \rangle$$

$$\}$$

Primero aparecen los rojos, después los blancos y finalmente los azules

# Bandera Holandesa (cont.)

La idea es usar solo intercambios de valores:

$$a.i, a.j := a.j, a.i$$

Intercambia los valores de a.i y a.j

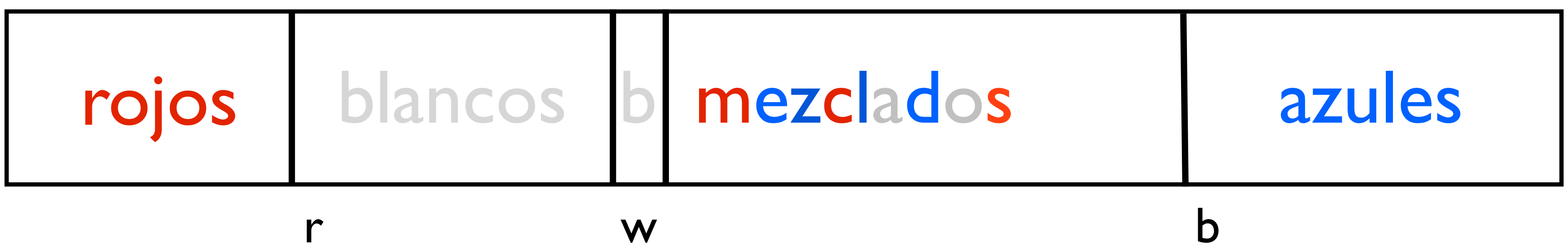
La idea del invariante es considerar una parte no ordenada:


El programa termina cuando w=b

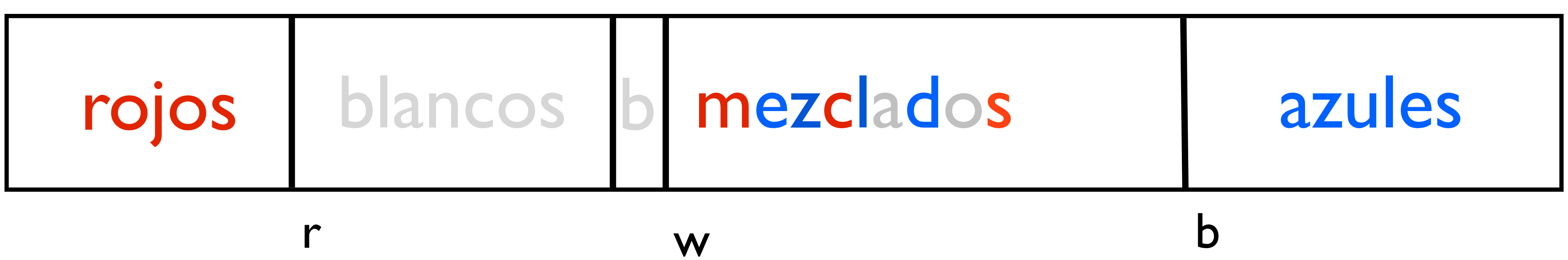
# Bandera Holandesa (cont)

Tenemos tres casos:

- a.w es blanco:

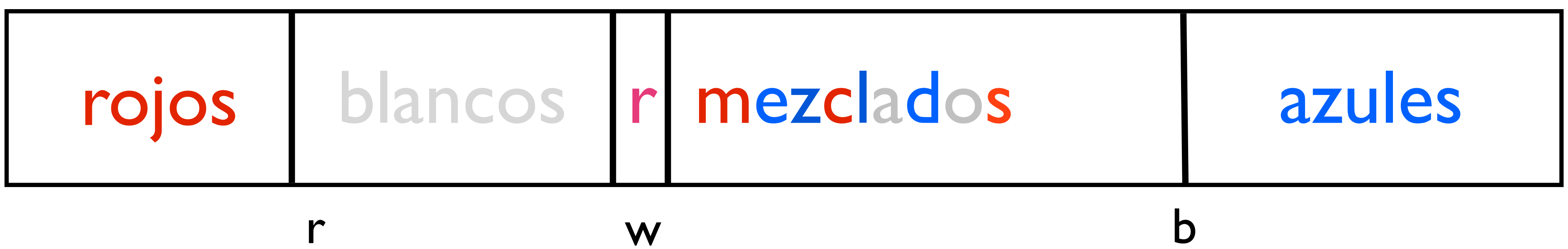



Avanzamos w:

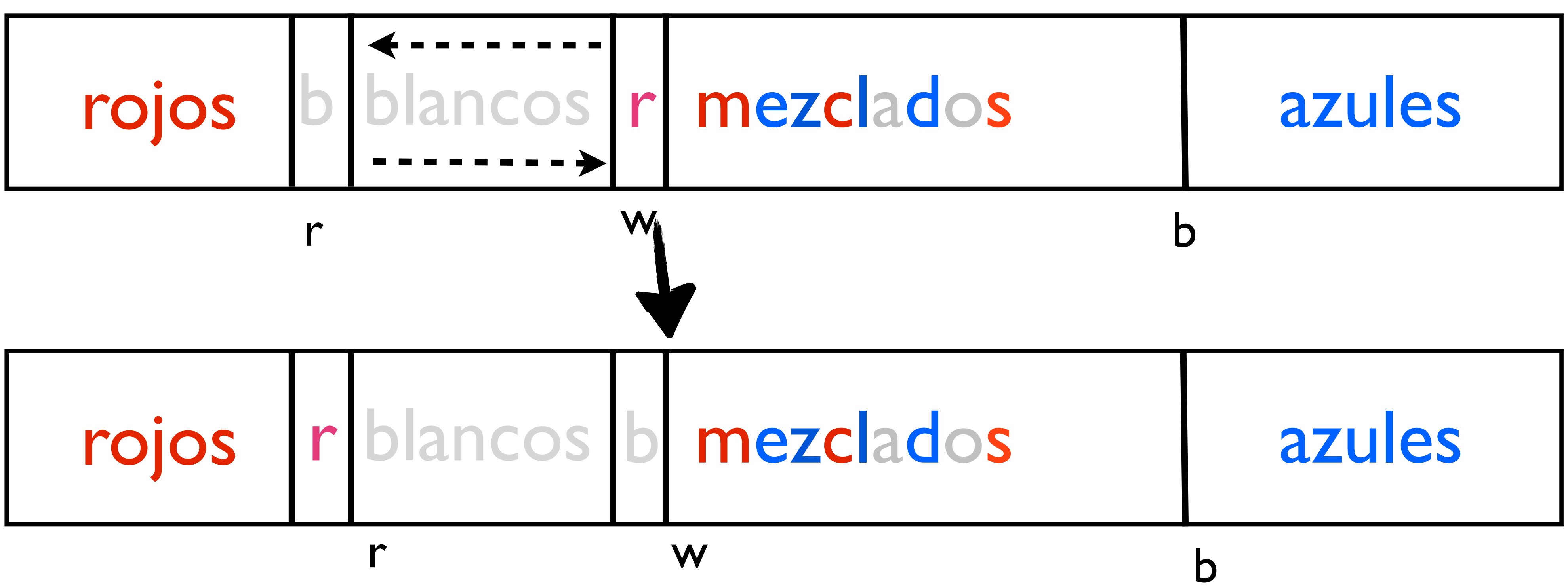


# Bandera Holandesa (cont)

- a.w es rojo:

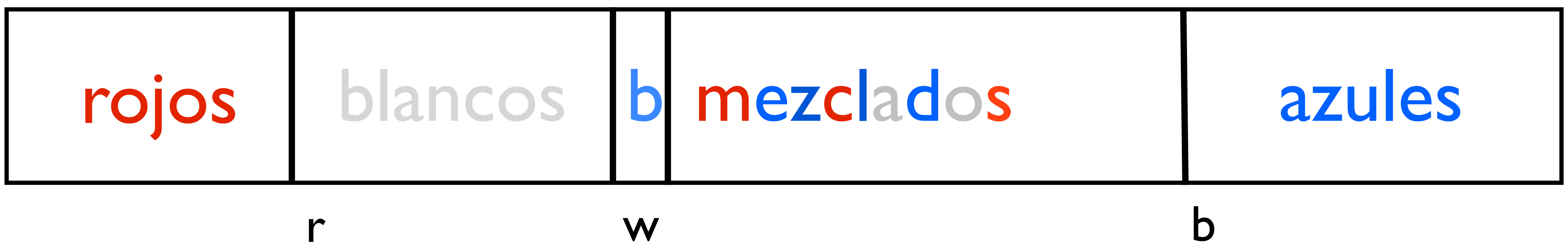



Intercambiamos a.w y a.r y avanzamos ambos

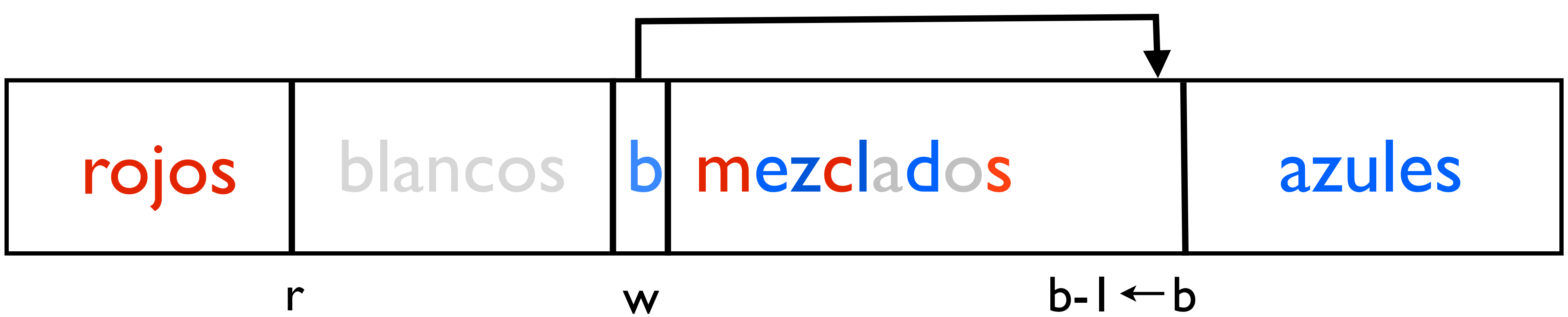


# Bandera Holandesa (cont.)

- a.w es azul:




Intercambiamos a.w y a.(b-1) y decrementamos b



# Bandera Holandesa (cont.)

Es decir, nos queda:

$$
\begin{array}{l}
r, w, b := 0, 0, N; \\
do\ w < b \to \\
\qquad if\ a.w = w \to w := w + 1; \\
\qquad \Box\ a.w = r \to a.w, a.r := a.r, a.w; \\
\qquad\qquad\qquad w, r := w + 1, r + 1; \\
\qquad \Box\ a.w = b \to a.w, a.(b-1) := a.(b-1), a.w; \\
\qquad\qquad\qquad b := b - 1; \\
\qquad fi \\
od
\end{array}
$$

Demostrar la corrección!

# Un Teorema sobre Cotas

Consideremos el siguiente programa:

$$\{0 < m \wedge 0 < n\}$$
$$i, j := m - 1, n - 1;$$
$$do\ j \neq 0 \rightarrow j := j - 1$$
$$\Box\ i \neq 0 \wedge j = 0 \rightarrow i, j := i - 1, n - 1$$
$$od$$
$$\{true\}$$

Cuál es la cota de este programa?

# Un Teorema sobre Cotas

Teorema: Si (i,j) es un par de expresiones tal que en cada paso del ciclo este par se decrementa lexicográficamente, y:

$$min_i \leq i \leq max_i \wedge min_j \leq j \leq max_j$$

Entonces una cota del ciclo es:

$$(i - min_i) * (1 + max_j - min_j) + j - min_j$$

Se puede generalizar para n-uplas. En el ejemplo anterior la cota es:

$$i * n + j$$