

# Tutorial de manejo de la herramienta de verificación formal de programas Verification Tools 1.0

En este apartado, especificaremos un tutorial de manejo de la herramienta de verificación de programas iterativos Verification Tools 1.0.

El programa a verificar es un algoritmo de división, y su código en C++ es el siguiente:

```
c=0;  
r=a;  
while(r>=b){  
    c=c+1;  
    r=r-b;  
}
```

Este programa presenta estas aserciones:

- **Precondición:**  $(a \geq 0) \wedge (b > 0)$
- **PostCondición:**  $((a == b * c + r) \wedge (r \geq 0)) \wedge (r < b)$
- **Invariante**  $((a == b * c + r) \wedge (r \geq 0)) \wedge (b > 0)$
- **Cota:**  $r$

A continuación se muestra como se deben cargar las distintas aserciones y código en la herramienta:

Verification Tools 1.0

VerificaciónResultadosTutorialIniciar pruebas

PRECONDICIÓN

(a>=0)&&(b>0)

POSTCONDICIÓN

((a==b\*c+r)&&(r>=0))&&(r<b)

INVARIANTE

((a==b\*c+r)&&(r>=0))&&(b>0)

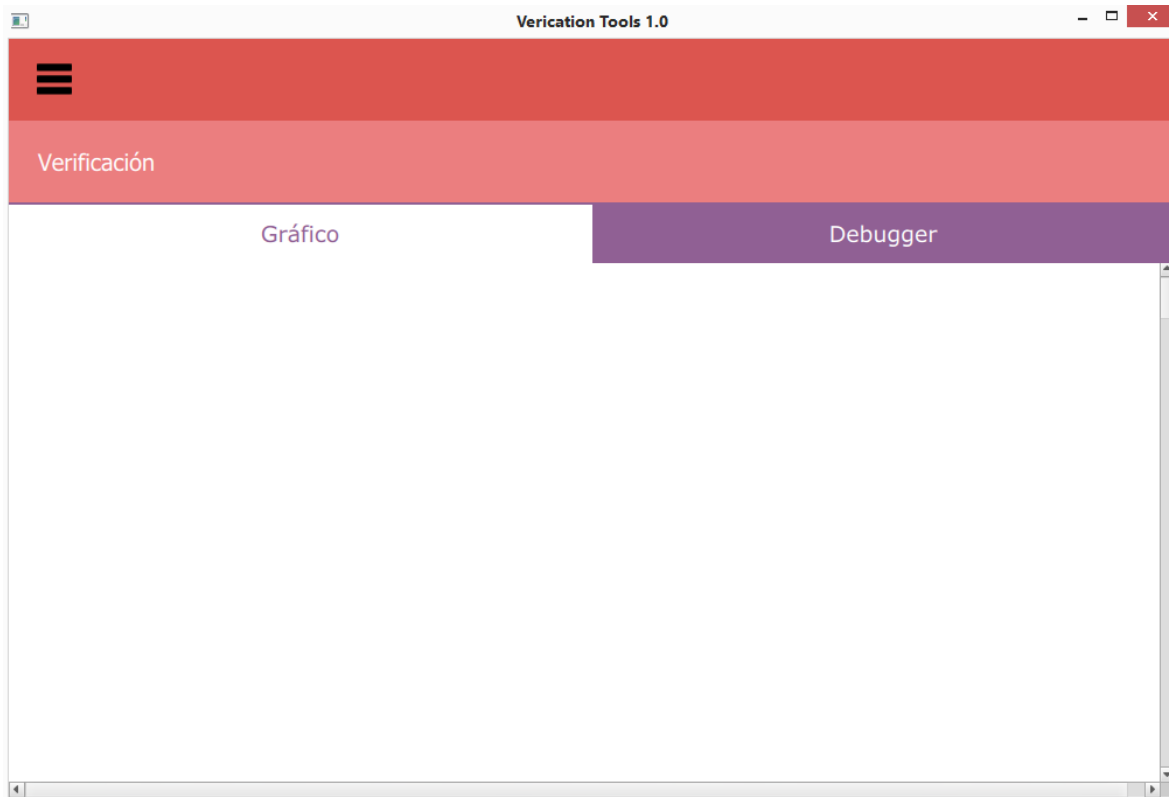
COTA

r

ALGORITMO

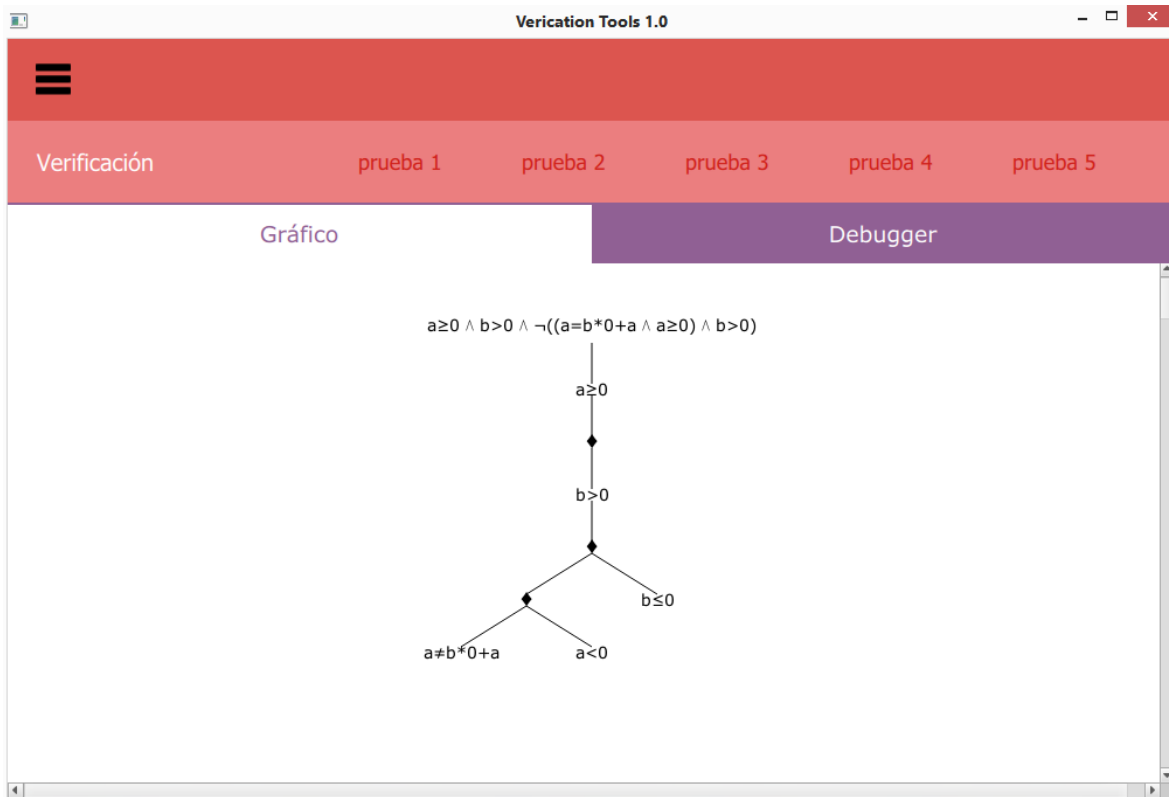
c=0;  
r=a;  
while(r>=b){  
    c=c+1;  
    r=r-b;  
}

Luego presionamos en el *Iniciar pruebas*, y nos trasladamos a la siguiente página, en donde la verificación comienza. En el caso de que se produzca algún error de sintaxis en cualquiera de la entradas, la celda de entrada se sombreara de un color rojo, y no se permitirá el inicio de las pruebas.

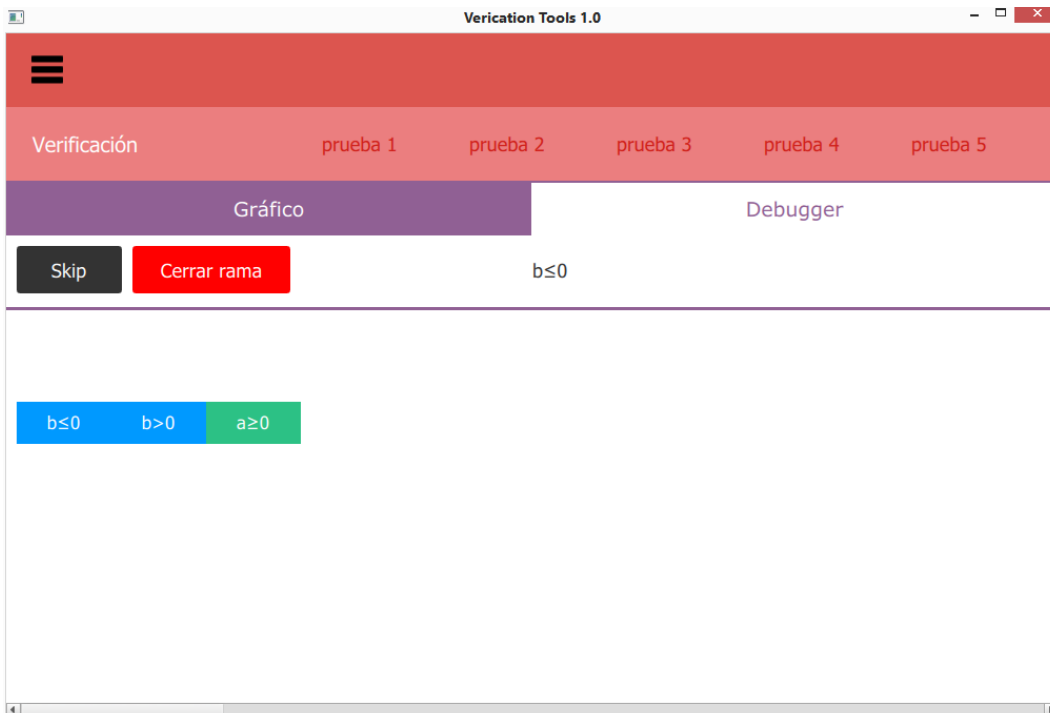


Esta es la pantalla de verificación, el botón *Grafico* imprime por pantalla el árbol a realizar el análisis. Al presionar *Debbuger* se comenzara con el análisis del árbol correspondiente, determinando que ramas del árbol se cierran y cuáles no.

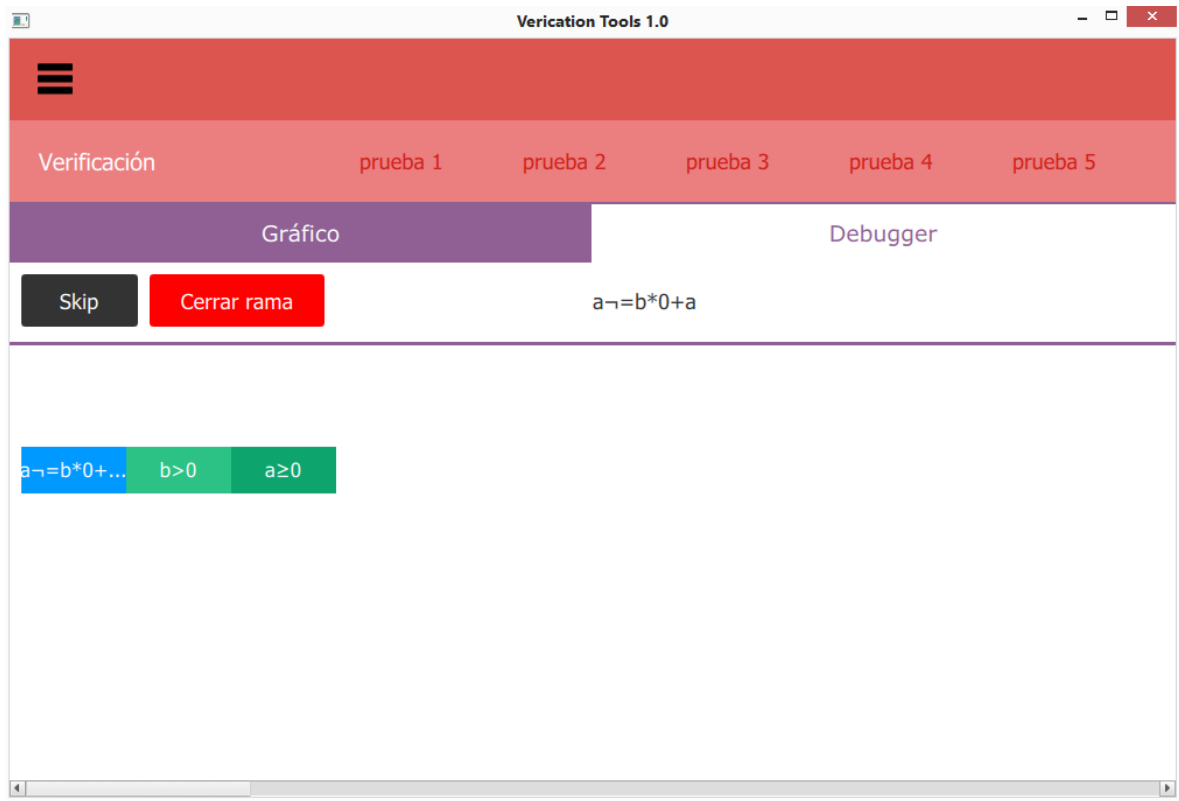
El árbol de la primera prueba ( $\{P\}A\{I\}$ ) a comprobar es el siguiente:



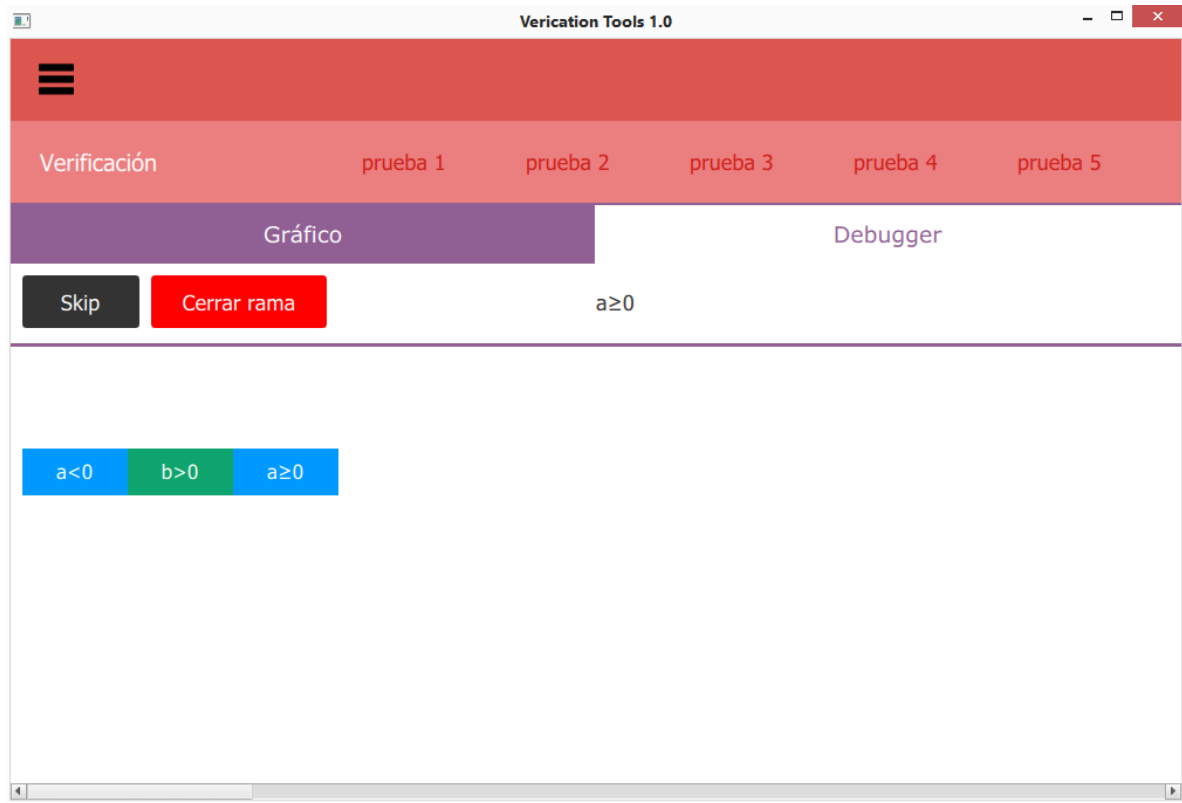
Presionamos *Debugger* y analizamos la primera rama del árbol. En esta encontramos la contradicción  $b \leq 0$  y  $b > 0$ , la cuales seleccionamos y presionamos el botón *Cerrar rama*.



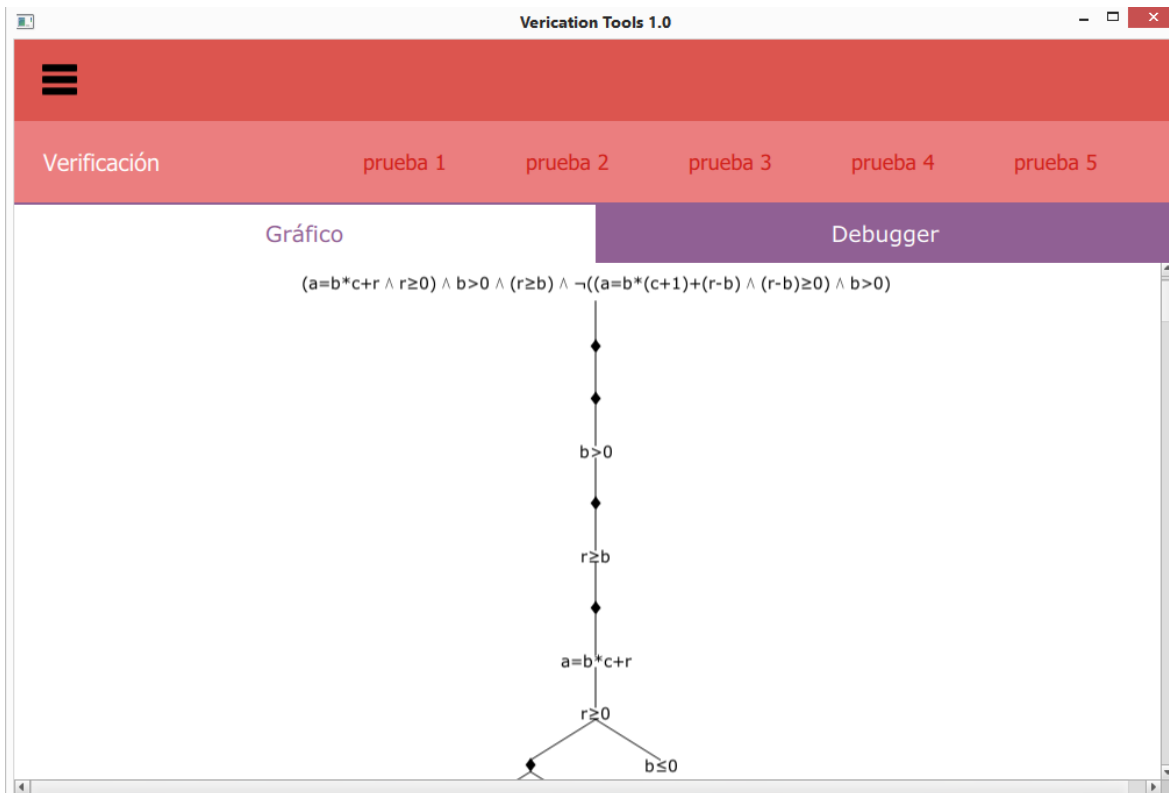
En la segunda rama, la contradicción es la siguiente:  $a \neq a$ , con lo cual la rama se cierra.



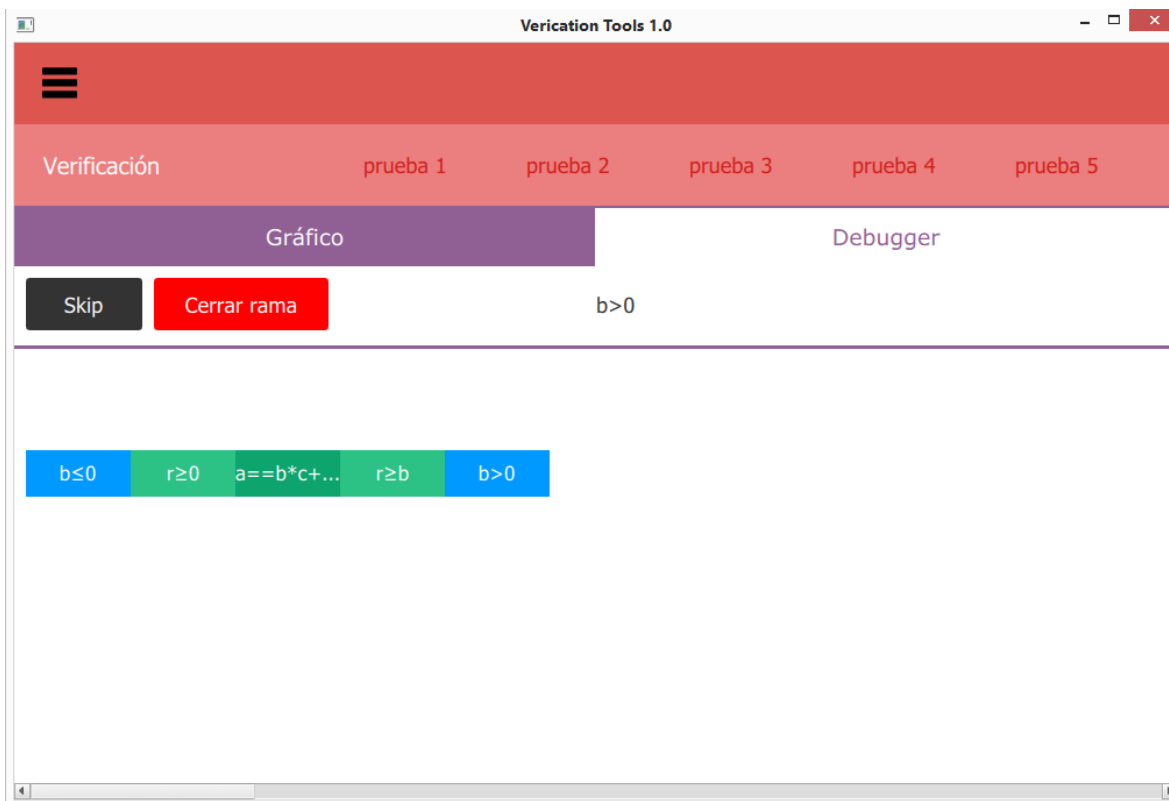
En la tercer rama la contradicción radica en los nodos  $a < 0$  y  $a \geq 0$ , con lo cual la rama se cierra.



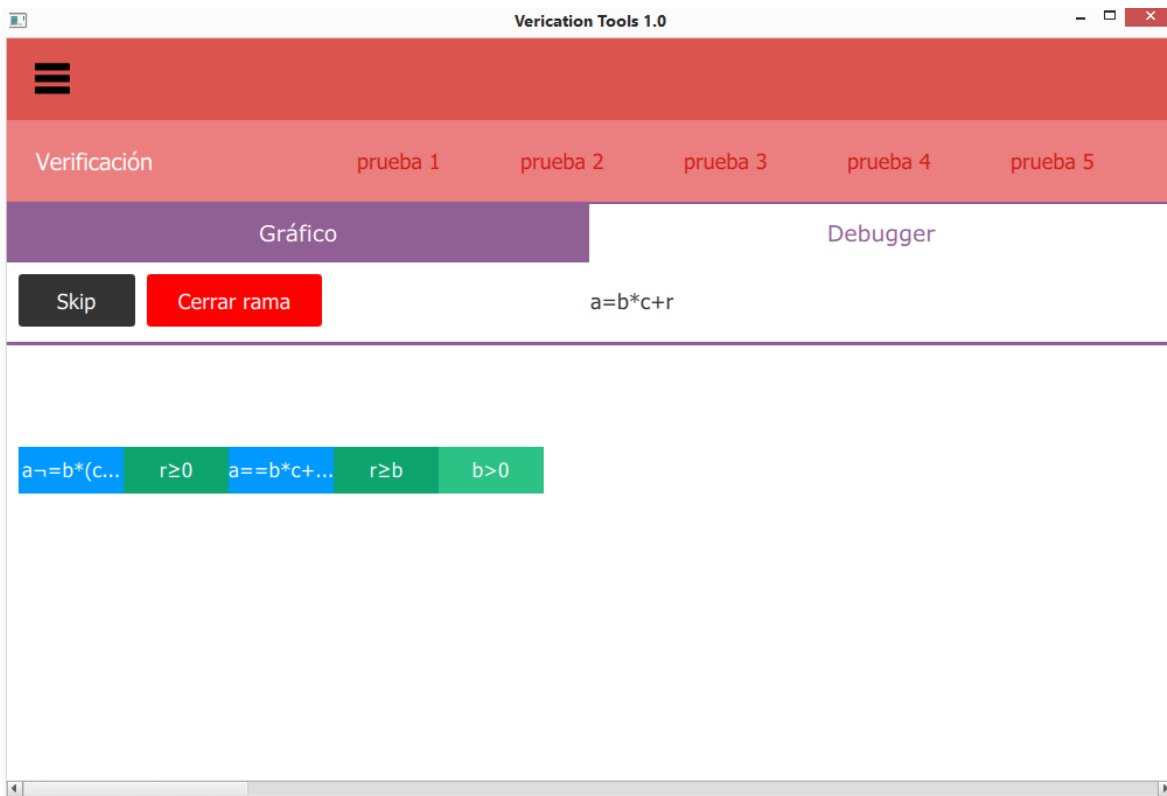
Ahora vamos a la segunda prueba a verificar:  $\{I \wedge b\}A\{I\}$ .



En esta rama la contradicción radica en las formulas  $b \leq 0$  y  $b > 0$ , con cual esta se cierra.

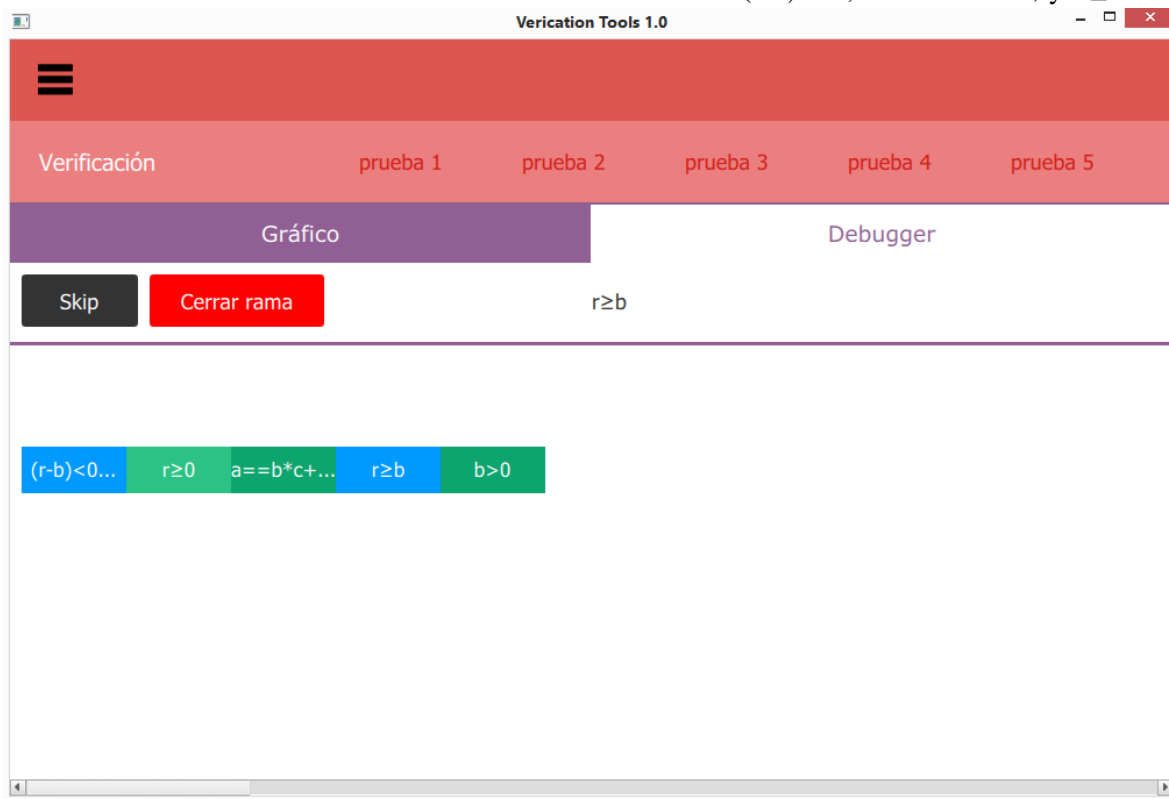


La rama siguiente, se encuentran dos contradicciones:  $a \neq b*(c+1) + (r-b)$  y  $a = b*c+r$ , con lo que concluimos que rama se cierra.



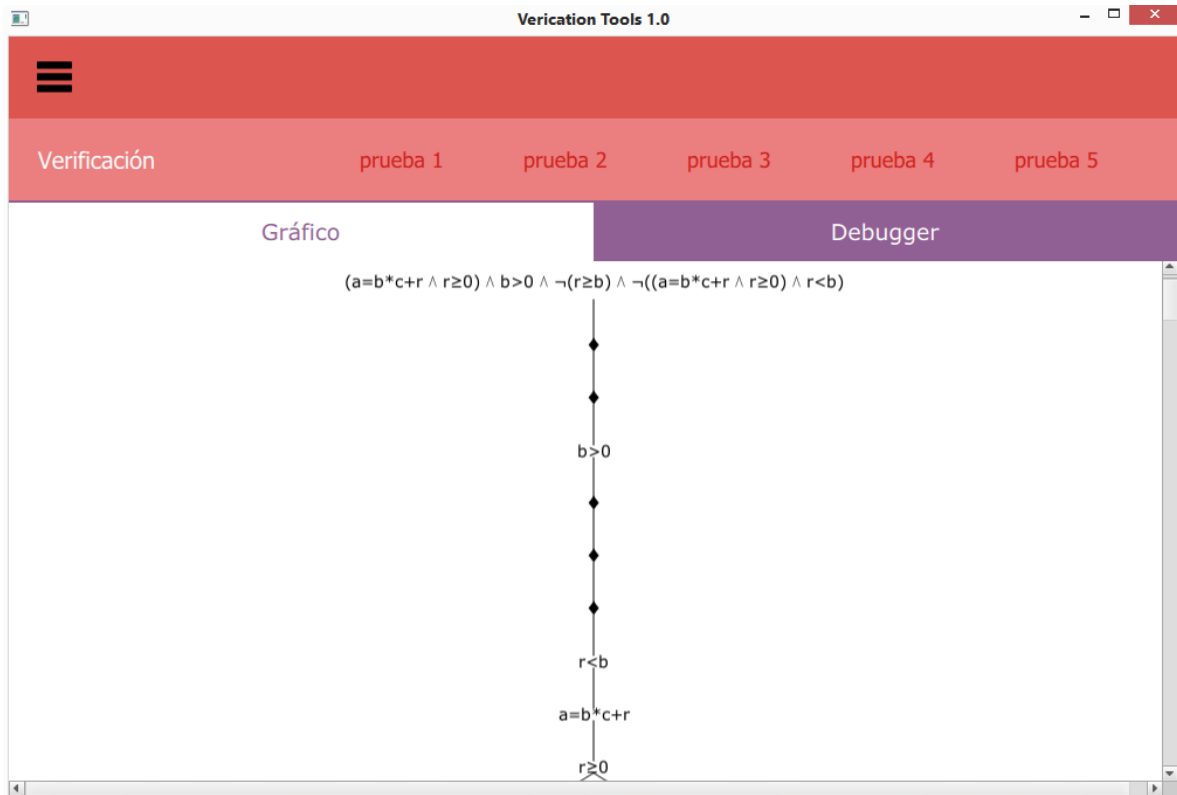


En esta rama las inconsistencias se encuentran en las formulas  $(r-b) < 0$ , es decir  $r < b$ , y  $r \geq b$ .

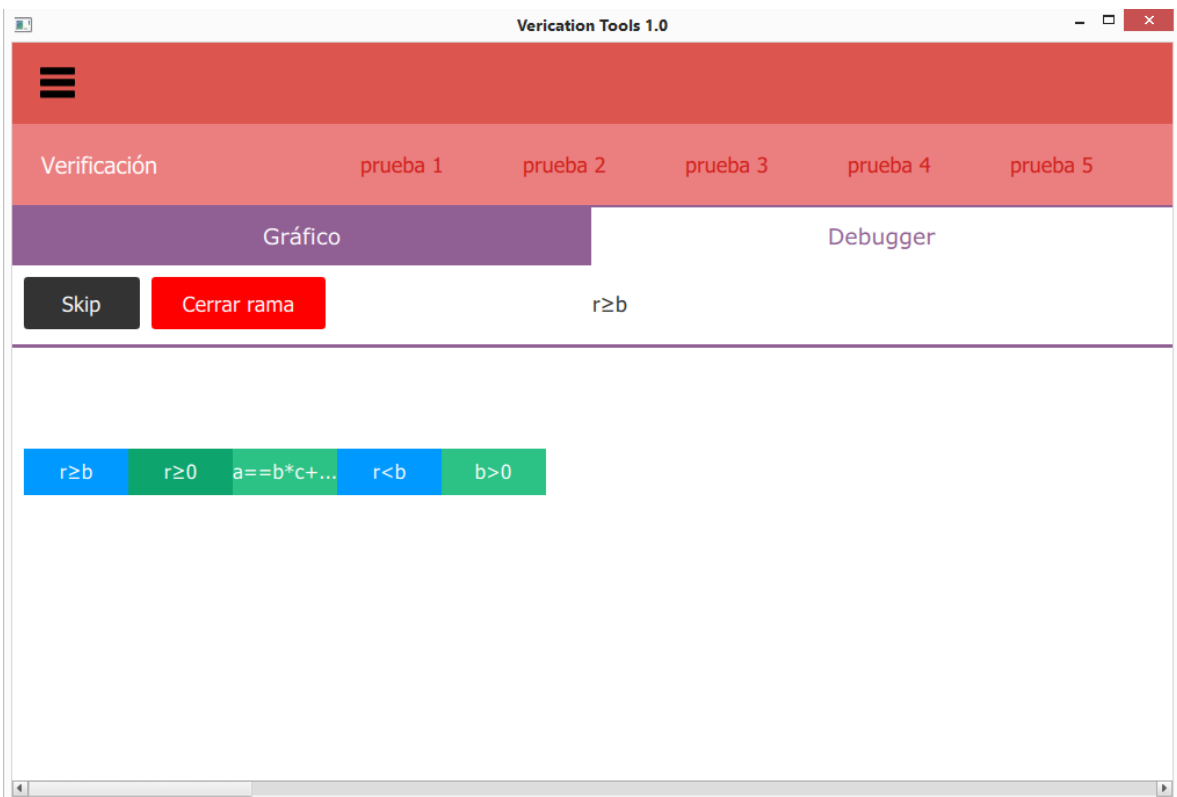


Con esto finaliza la segunda prueba.

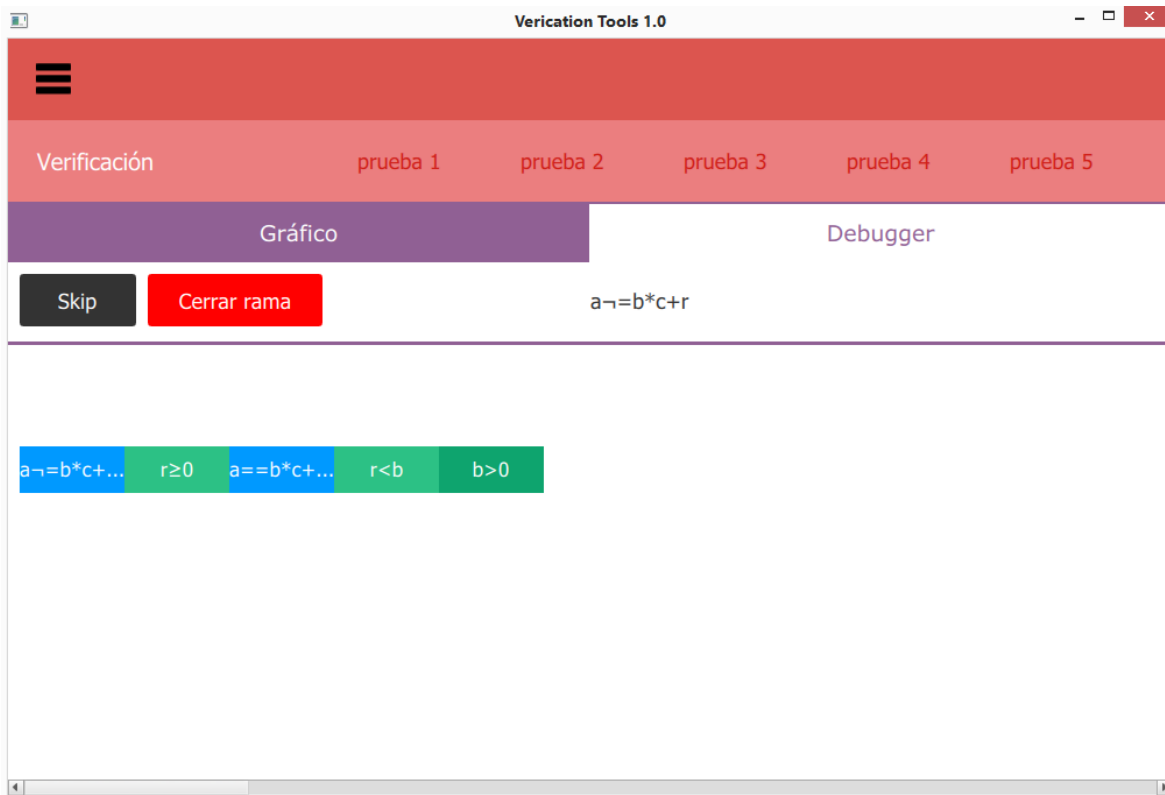
La siguiente prueba es  $I \wedge \neg B \rightarrow Q$ , el árbol generado es el siguiente:



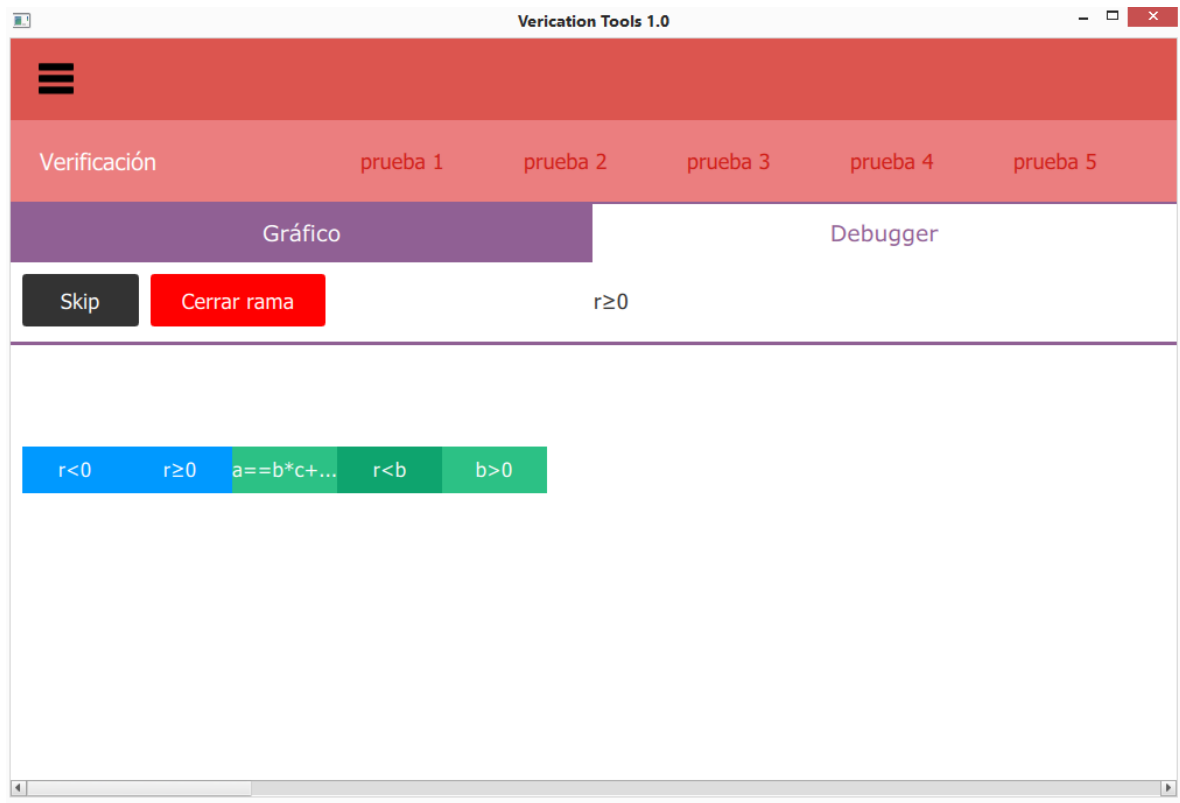
En la primera rama a analizar, con la misma metodología que venimos desarrollando, encontramos estas 2 inconsistencias  $r \geq b$  y  $r < b$ , las cuales cierran la rama.



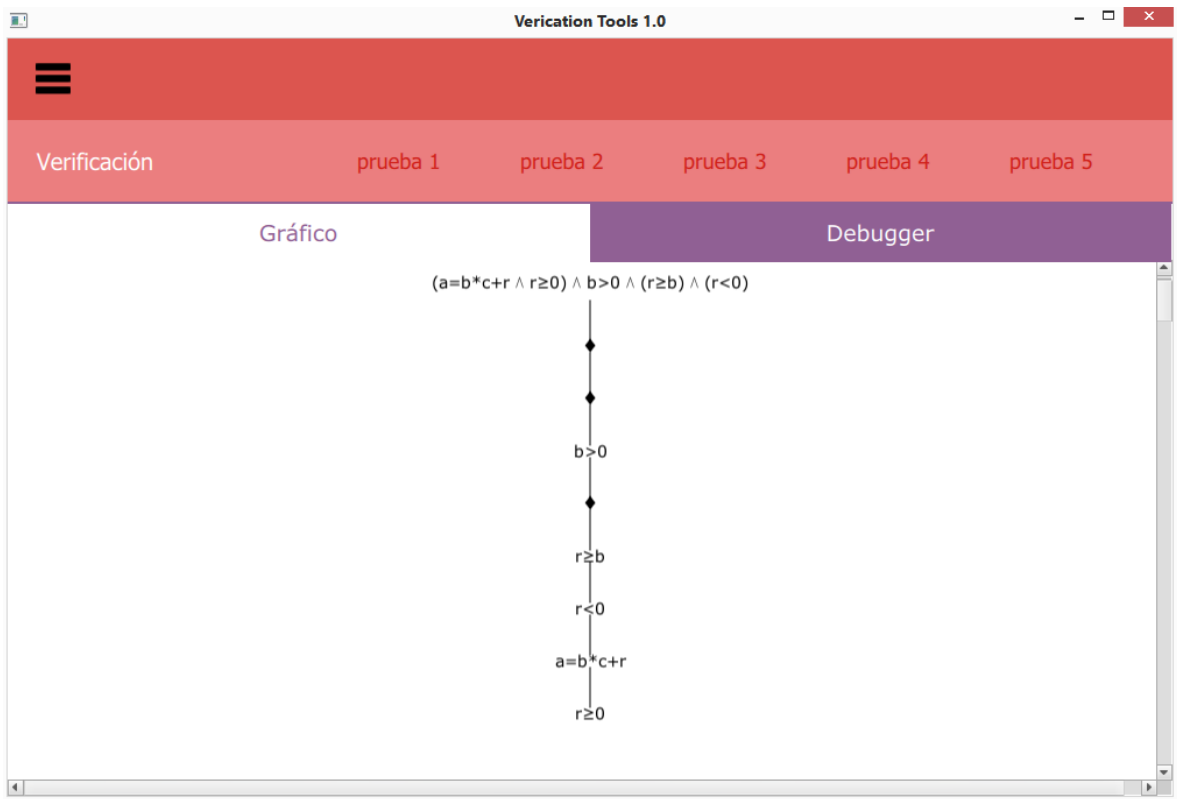
En la siguiente rama, la contradicción radica en las formulas  $a \neq b*c+r$  y  $a=b*c+r$ .



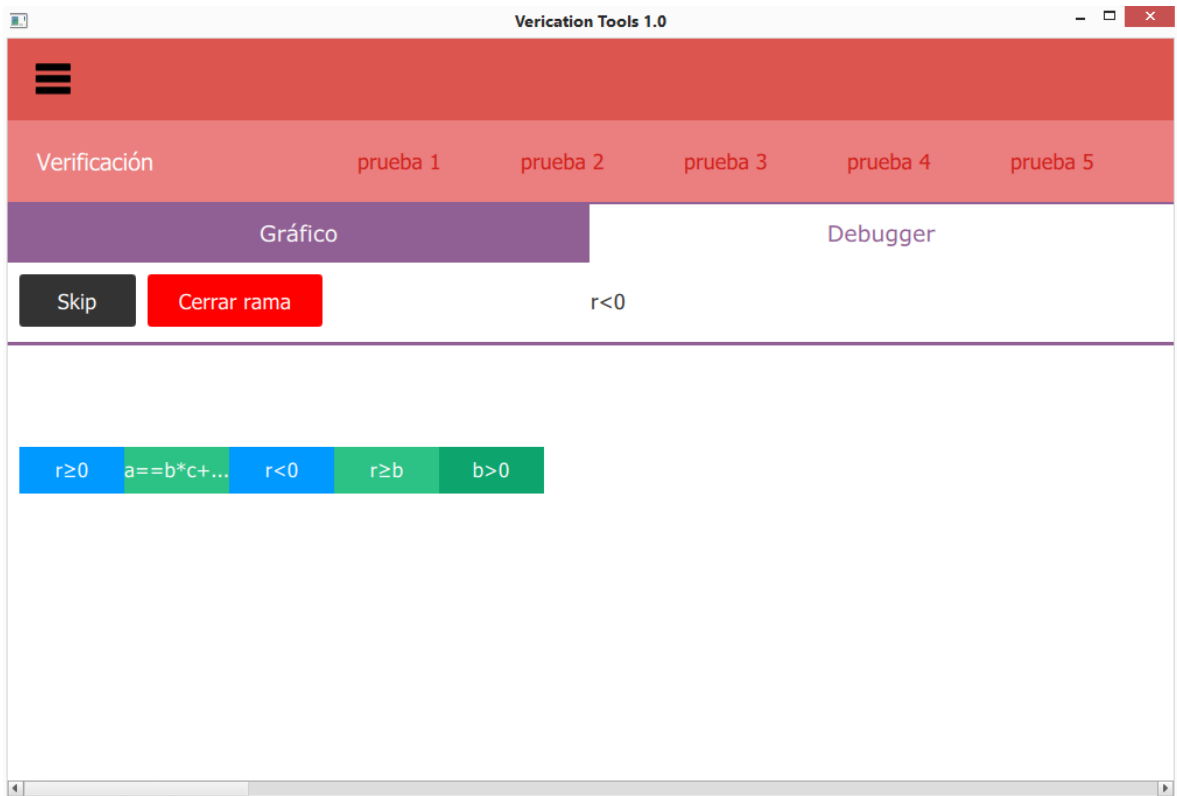
La ultima rama del árbol, se cierra con la inconsistencia  $r < 0$  y  $r \geq 0$ .



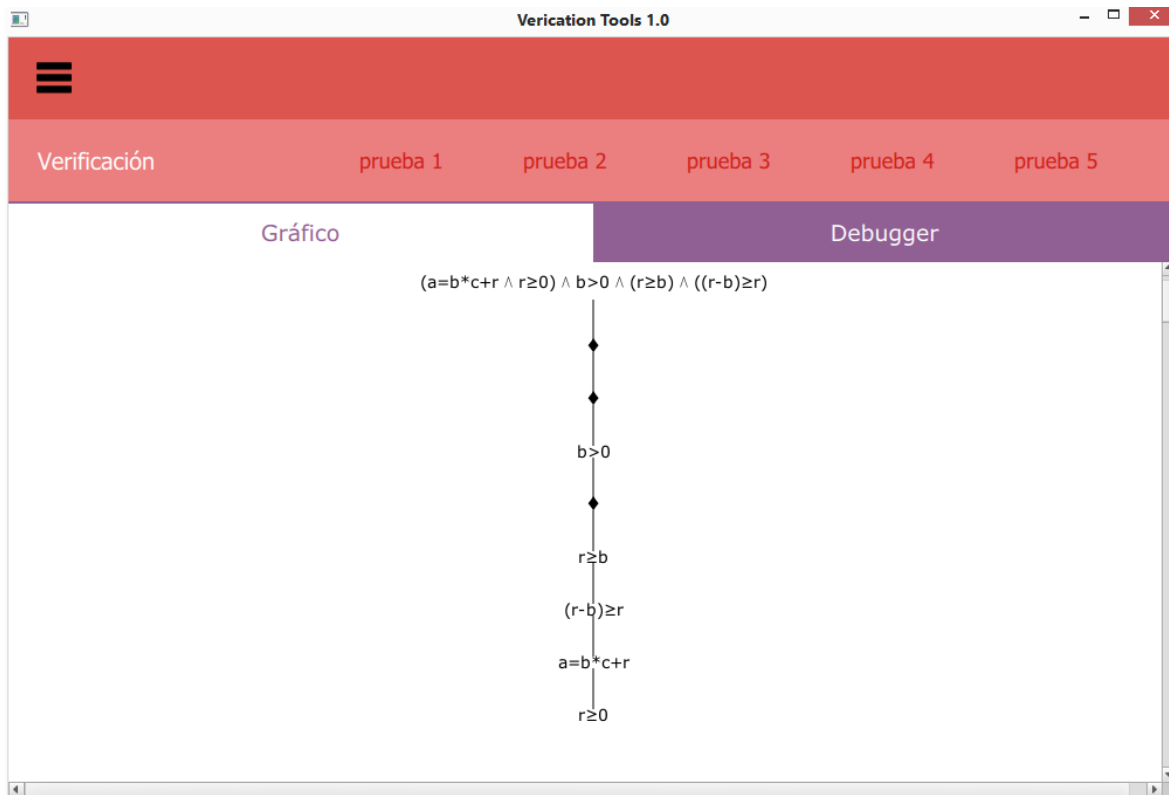
La cuarta prueba  $I \wedge B \rightarrow C \geq 0$ , presenta la el siguiente grafico.



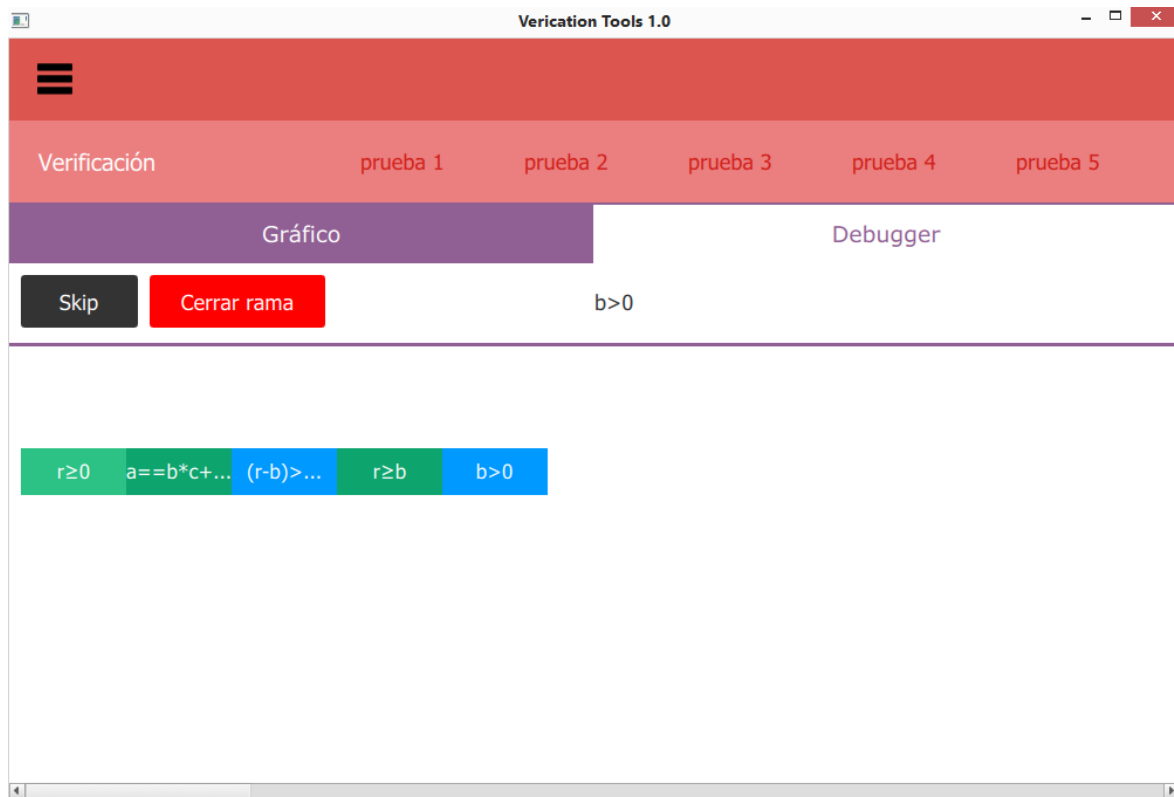
Este árbol contiene una única rama, la cual se cierra con la contradicción  $r < 0$  y  $r \geq 0$ .



La última prueba para verificar el programa, es  $\{I \wedge B \wedge C = T\}A\{C < T\}$ , que cuenta con el siguiente árbol.



Este al igual que el anterior árbol, consta con una sola rama, que se cierra con la inconsistencia  $(r-b) > r$ , simplificando  $b \leq 0$ , y  $b > 0$ .



Al terminar de verificar todas las pruebas formales, volvemos al primer menú de la herramienta, y presionamos el botón *Resultados*, donde se encontraremos el informe y conclusión de la verificación anteriormente realizada.

