

PRUEBA DE PATRONES

El problema de parar es fundamentalmente indecidible, lo que significa que no existe un algoritmo que pueda determinar de manera general y para cualquier programa si este se detendrá o continuará ejecutándose indefinidamente. Simular esto requiere trucos o suposiciones que no capturan completamente la naturaleza del problema real.

SIMULACIÓN DE “HALTCHECKER”:

Mi HaltChecker utiliza una estrategia de tiempo “TimeHaltStrategy” para decidir si un programa se detiene. Un programa que se ejecute por más de 5 segundos es probable que no termine. Sin embargo esto es una simple simulación y claramente puede haber programas que terminen pero duren más.

STRATEGY:

Implementar un HaltChecker que funcione con cualquier programa es un desafío debido a la variedad de comportamientos que pueden exhibir los programas. Por lo tanto, he decidido usar el patrón strategy para poder cambiar la estrategia de analizar el código sin tener que modificar todo el programa.

REVERSER:

Actúa de manera contraria basándose en la salida de “HaltChecker”. Imita la paradoja de intentar utilizar la lógica contra sí misma en computación.