1 Pair & Halt

Instruction: If you have an "if-block", find someone with an "else-block" (and vice versa) so that, when combined, your code snippets ensure this program *always* halt *regardless of user's input.*

$x \leftarrow$ user's input
$y \leftarrow$ user's input
while $x > 0$ and $y > 0$ **do**
    $z \leftarrow$ user's input
    [**if-block**]
    [**else-block**]

<u>**if-blocks**</u>	<u>**else-blocks**</u>
A **if** z is even **then** $x \leftarrow x - 1$ $y \leftarrow y + 1$	A **else** $y \leftarrow y - 1$
B **if** x is odd **then** $x \leftarrow x + 1$	B **else** $x \leftarrow x + 1$ $y \leftarrow y - 1$
C **if** $z < 5$ **then** $x \leftarrow x - 2$ $y \leftarrow y + 1$	C **else** $x \leftarrow x - 2$

Solution: All pairs halt except if-block A + else-block B. For that pair, The sequence $x \leftarrow 2$, $y \leftarrow 2$, and $z_i \leftarrow i$ (where z_i denotes the i-th input for z) causes the program to run forever. Notice that z_i at each i-th input alternates between even and odd. This causes the program to alternate between incrementing and decrementing x and y at each iteration of the while loop. Concretely, in our example (x, y) alternates between $(2, 2)$ and $(3, 1)$, never terminating.

2 Potential Function

Instruction: Define and analyze a potential function (in terms of x and/or y) for your algorithm to prove that it halts on all inputs.

If-block A + Else-block A

Solution: Define a unit of work to be one iteration of the while loop and the potential $s = 2x+y$. Let x_i, y_i, and z_i denote the value of x, y, and z on the i-th iteration, respectively. Consider the following two cases:

- When z_i is even,

$$\begin{aligned} s_{i+1} &= 2(x_i - 1) + (y_i + 1) \\ &= 2x_i + y_i - 1 \\ &= s_i - 1 \end{aligned}$$

- When z_i is odd,

$$\begin{aligned} s_{i+1} &= 2x_i + y_i - 1 \\ &= s_i - 1 \end{aligned}$$

In both cases, the potential decreases by 1 with every unit of work. Moreover, $s = 2x + y > 0$ otherwise the program must terminate.

It is instructive to see the thought process in deriving $s = 2x + y$. First, notice that $s = x$ doesn't work because the user might always input odd z, resulting in no change in the potential. $s = y$ also doesn't work because the user might always input even z, causing the potential to increase on every iteration. This lead us to a linear combination of x and y, say $s = ax + by$. We want s to decrease regardless of whether z is even, which means we want the following to happen:

$$\begin{cases} a(x-1) + b(y-1) & < ax + by \\ ax + b(y-1) & < ax + by \end{cases} \implies \begin{cases} -a + b & < 0 \\ -b & < 0 \end{cases}$$

So any pairs of a, b satisfying $a > b > 0$ would work. For simplicity, we choose $a = 2$ and $b = 1$, yielding $s = 2x + y$.

If-block A + Else-block C

Solution: Define a unit of work to be one iteration of the while loop and the potential $s = x$. Let x_i and z_i denote the value of x and z on the i-th iteration, respectively. Consider the following two cases:

- When z_i is even,

$$s_{i+1} = x_i - 1 = s_i - 1$$

- When z_i is odd,

$$s_{i+1} = x_i - 2 = s_i - 2$$

In both cases, the potential decreases by at least 1 with every unit of work. Moreover, $s = x > 0$ otherwise the program must terminate.

If-block B + Else-block A

Solution: The key observation here is the algorithm will always enter the else block after at most one iteration. Specifically,

- If the initial x is odd, it increments by 1 on the first iteration and becomes even on the next iteration.
- If the initial x is even, the algorithm enters the else-block which does not change x (so x will still be even on the next iteration.

Define a unit of work to be **two** iterations of the while loop and the potential $s = y$. Let y_i denotes the value of y on the i-th iteration,

$$s_{i+1} = \begin{cases} y_i - 1 & \text{if } i = 1 \text{ and initial } x \text{ is odd} \\ y_i - 2 & \text{otherwise} \end{cases}$$

Therefore, the potential decreases by at least 1 with every unit of work. Moreover, $s = y > 0$ otherwise the program must terminate.

If-block B + Else-block B

Solution: The key observation here is that the algorithm alternates between the if-block and the else-block during iterations of the while loop because x is incremented by 1 regardless of whether the algorithm enters the if-block or the else-block. This means that y decreases by 1 every two iterations. Define a unit of work to be **two** iterations of the while loop and the potential $s = y$. Let y_i denotes the value of y on the i-th iteration,

$$s_{i+1} = y_i - 1 = s_i - 1$$

Therefore, the potential decreases by at least 1 with every unit of work. Moreover, $s = y > 0$ otherwise the program must terminate.

If-block B + Else-block C

Solution: The key observation here is the algorithm will always enter the else block after at most one iteration. Specifically,

- If the initial x is odd, it increments by 1 on the first iteration and becomes even on the next iteration.
- If the initial x is even, $x - 2$ will still be even on the next iteration.

Define a unit of work to be **two** iterations of the while loop and the potential $s = x$. Let x_i denotes the value of x on the i-th iteration,

$$s_{i+1} = \begin{cases} x_i - 1 & \text{if } i = 1 \text{ and initial } x \text{ is odd} \\ x_i - 2 & \text{otherwise} \end{cases}$$

Therefore, the potential decreases by at least 1 with every unit of work. Moreover, $s = x > 0$ otherwise the program must terminate.

If-block C + Else-block A

Solution: Define a unit of work to be one iteration of the while loop and the potential $s = x+y$. Let x_i, y_i, and z_i denote the value of x, y, and z on the i-th iteration, respectively. Consider the following two cases:

- When $z_i < 5$,

$$\begin{aligned} s_{i+1} &= (x_i - 2) + (y_i + 1) \\ &= x_i + y_i - 1 \\ &= s_i - 1 \end{aligned}$$

- When $z_i \geq 5$,

$$\begin{aligned} s_{i+1} &= x_i + (y_i - 1) \\ &= s_i - 1 \end{aligned}$$

In both cases, the potential decreases by 1 with every unit of work. Moreover, $s = x + y > 0$ otherwise the program must terminate.

If-block C + Else-block B

Solution: Define a unit of work to be one iteration of the while loop and the potential $s = 2x+3y$. Let x_i, y_i, and z_i denote the value of x, y, and z on the i-th iteration, respectively. Consider the following two cases:

- When $z_i < 5$,

$$\begin{aligned} s_{i+1} &= 2(x_i - 2) + 3(y_i + 1) \\ &= 2x_i + 3y_i - 1 \\ &= s_i - 1 \end{aligned}$$

- When $z_i \geq 5$,

$$\begin{aligned} s_{i+1} &= 2(x_i + 1) + 3(y_i - 1) \\ &= 2x_i + 3y_i - 1 \\ &= s_i - 1 \end{aligned}$$

In both cases, the potential decreases by 1 with every unit of work. Moreover, $s = 2x + 3y > 0$ otherwise the program must terminate.

It is instructive to see the thought process in deriving $s = 2x + 3y$. Using the same linear combination trick as in if-block A + else-block A, let $s = ax + by$. We want s to decrease regardless of whether $z < 5$, which means we want the following to happen:

$$\begin{cases} a(x-2) + b(y+1) & < ax + by \\ a(x+1) + b(y-1) & < ax + by \end{cases} \implies \begin{cases} -2a + b & < 0 \\ a - b & < 0 \end{cases}$$

So any pairs of a, b satisfying $a < b < 2a$ would work. For simplicity, we choose $a = 2$ and $b = 3$, yielding $s = 2x + 3y$.

If-block C + Else-block C

Solution: Define a unit of work to be one iteration of the while loop and the potential $s = x$. Let x_i denotes the value of x on the i-th iteration,

$$s_{i+1} = x_i - 1 = s_i - 1$$

Therefore, the potential decreases by 1 with every unit of work. Moreover, $s = x > 0$ otherwise the program must terminate.