Ejercicio 3. Decimos que un programa P es *autocontenido* si en cada instrucción *IF* $V \neq 0$ *GOTO* L que ocurre en P, L es una etiqueta definida en P.

a. Demostrar que todo programa P tiene un programa autocontenido P' equivalente (P y P' son programas equivalentes si $\Psi_P^{(n)} = \Psi_{P'}^{(n)}\ \forall n \geq 1$).

b. Sean P y Q dos programas autocontenidos con etiquetas disjuntas y sea $r : \mathbb{N}^n \to \{0,1\}$ un predicado primitivo recursivo. Definir macros para las siguientes pseudo-instrucciones (con su interpretación natural):

- IF $r(V_1, \ldots, V_n)$ GOTO L
- IF $r(V_1, \ldots, V_n)$ THEN P ELSE Q
- WHILE $r(V_1, \ldots, V_n)$ P

c. Dadas las funciones $f, g : \mathbb{N} \to \mathbb{N}$ definidas por

$$f(x) = \begin{cases} 1 & \text{si } x = 3 \\ \uparrow & \text{en otro caso} \end{cases} \qquad \text{y} \qquad g(x) = 2x$$

Demostrar que es $\mathcal{S}$-parcial computable la función

$$h(x) = \begin{cases} f(x) & \text{si } x \geq 5 \ \vee \ x = 3 \\ g(x) & \text{en otro caso} \end{cases}$$

a

Se puede hacer un programa P' que recibe un programa P y le busca las instrucciones de salto condicional que tenga etiquetas que no sean de instrucciones validas. Cuando se encuentra, define una instrucción con dicha etiqueta talque la operación sea no hacer nada ($V \longleftarrow V$). Luego agrego las nuevas instrucciones debajo del programa original e y obtengo un programa e' que computa la misma función que la original pero con todas las etiquetas definidas. Finalmente la salida de P' es $Y \longleftarrow \phi^n(x_1, \cdots x_n, e')$

Sabemos que $\phi^n(x_1, \cdots x_n, e') = \phi^n(x_1, \cdots x_n, e)$ por como definimos el programa, el comportamiento en ambos es el mismo.

Finalemente $\phi^n(x_1, \cdots x_n, e') = \phi^n(x_1, \cdots x_n, e) \iff \psi^n(x_1, \cdots x_n, P') = \psi^n(x_1, \cdots x_n, P)$

b

> $IF\ r(V_1 \cdots V_n)\ GOTO\ L$

```
Z₁ ← r(V₁  ⋯  Vₙ)
IF Z₁ ≠ 0 GOTO L
```

> $IF\ r(V_1 \cdots V_n)\ THEN\ P\ ELSE\ Q$

```
    Z₁ ← r(V₁  ⋯  Vₙ)
    IF Z₁ ≠ 0 GOTO L
    Y ← ψⁿ(V₁,  ⋯, Vₙ, Q)
    GOTO E
[L] Y ← ψⁿ(V₁,  ⋯, Vₙ, P)
```

1

> $WHILE\ r(V_1, \cdots V_n,)\ P$

```
    Z₁ ← r(V₁  ⋯  Vₙ)
    IF Z₁ ≠ 0 GOTO L
    GOTO E

[L] P
    Z₁ ← r(V₁  ⋯  Vₙ)
    IF Z₁ ≠ 0 GOTO L
```

c

```
    IF x = 3 ∨ x ≥ 5 THEN P ELSE Q
```

> P (programa para f)

```
        IF X₁ = 3 GOTO S
  [L]   IF X₁ ≠ 3 GOTO L
  [S]   Y ← 1
```

Q (programa para g)

```
Y ⟵ X₁ + X₁ // programa ej 1b
```