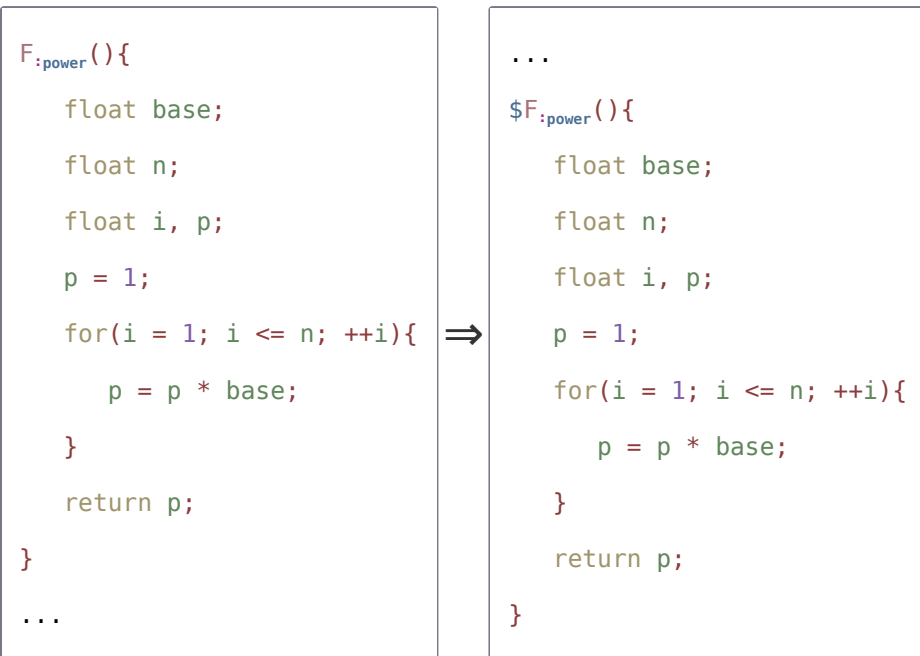


Ejemplo

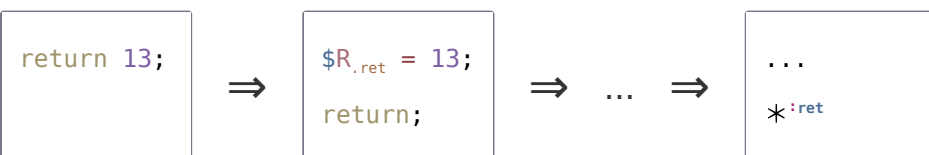


$\langle retvp \rangle \rightarrow lit \mid V_n'$
 $\langle retv \rangle \rightarrow \langle sign \rangle \langle retvp \rangle$

return; \Rightarrow $\ast :ret$

return $\langle retv_1 \rangle$; \Rightarrow $\&R_{:ret} = \langle retv_1 \rangle$;
return;

Ejemplo



←:⟨id⟩

Señales que pone el expansor.

$W \rightarrow V \mid N$

```
$W_{:nombre}() {  
    texto1  
}
```

→

```
:def_subp:{  
    #definec $R W  
    texto1  
    Z:i = *Y:top;  
    :rest_vals:  
    *  
    1:np  
}
```

←:ret

←:i

←:npos

Ejemplo

```
$F:power(){  
    float base;  
    float n;  
    float i, p;  
    p = 1;  
    for(i = 1; i <= n; ++i){  
        p = p * base;  
    }  
    return p;  
}
```

⇒

```
:def_subp:{  
    #definec $R F  
    float base;  
    ...  
    p = p * base;  
}  
return p;  
Z:67 = *Y:top;  
:rest_vals:  
*  
1:np  
}
```

←:ret

←:67

←:npos

:rest_vals:

```
:rest_vals: ➔ Z_:top--;  
               Z_:r1 = *Y_:top;  
               Z_:top--;  
               Z_:r2 = *Y_:top;  
               Z_:top--;  
               Z_:and = *Y_:top;  
               Z_:top--;  
               Z_:or = *Y_:top;  
               Z_:top--;  
               Z_:obl = *Y_:top;  
               Z_:top--;  
               Z_:opl = *Y_:top;  
               Z_:top--;  
               Z_:brk = *Y_:top;  
               Z_:top--;  
               Z_:sw = *Y_:top;  
               Z_:top--;  
               Z_:top = *Y_:top;
```

:def_subp:{ *texto*₁ }

Sea m el $\langle \text{natural} \rangle$ asociado a **:num_position:**. El expansor:

- asocia 1 a **:num_position:**.
- expande las macroinstrucciones en *texto*₁ hasta obtener $\langle \text{preinstrucciones}_1 \rangle$.
- sustituye \leftarrow **:return** por una señal, sea \leftarrow **:k**, y todas las apariciones de **:return** en $\langle \text{preinstrucciones}_1 \rangle$ por **:k**.
- sustituye la última $\langle \text{preinstrucción} \rangle$, 1^{np} , por 1^n , donde n es el $\langle \text{natural} \rangle$ asociado a **:num_position:**.
- sustituye \leftarrow **:npos** por una señal, sea \leftarrow **:p**, y todas las apariciones de **:npos** en $\langle \text{preinstrucciones}_1 \rangle$ por **:p**.
- asocia m a **:num_position:**.

Expansiones finales 1''

Si se usan macroinstrucciones **:def_subp:**, el expansor, en las expansiones finales:

- antes de añadir la *⟨preinstrucción⟩* **1^{top}** añade una *⟨preinstrucción⟩* **1ⁿ**, donde *n* es el *⟨natural⟩* asociado a **:num_position:**.
- añade una señal **←:\$npos.** apuntando a esta *⟨preinstrucción⟩*.
- sustituye las apariciones de **:npos** en cualquier *⟨preinstrucción⟩* por **:\$npos**.
- por cada señal **←:⟨id_i⟩** presente:
 - la sustituye por una señal **←:⟨natural⟩**; sea ésta **←:i**.
 - sustituye las apariciones de **:⟨id_i⟩** en cualquier *⟨preinstrucción⟩* por **:i**.

Llamada a subprogramas

$$\begin{aligned} \langle argp_y \rangle &\rightarrow \langle natural \rangle \mid \mathbf{0} \mid \mathbf{Y}'_n \mid \&V_n \langle indf \rangle \\ \langle arg_y \rangle &\rightarrow \langle argp_y \rangle \mid + \langle argp_y \rangle \\ \langle argp_f \rangle &\rightarrow \langle racional \rangle \mid \mathbf{F}'_n \\ \langle arg_f \rangle &\rightarrow \langle signo \rangle \langle argp_f \rangle \\ \langle arg \rangle &\rightarrow \langle arg_y \rangle \mid \langle arg_f \rangle \\ \langle rest_args \rangle &\rightarrow \varepsilon \mid , \langle arg \rangle \langle rest_args \rangle \\ \langle args \rangle &\rightarrow \varepsilon \mid \langle arg \rangle \langle rest_args \rangle \end{aligned}$$