

Ejemplo 2. La definición de funciones en cierto lenguaje es de la siguiente manera:

```
tipo function nombre (tipo: nombre [: tipo: nombre]*)  
declare {  
    [tipo: nombre [: nombre]* ; | decl-function ]+  
}  
begin {  
    [sentence ]+  
}
```

donde tipo puede ser **integer** o **char**, nombre comienza por una letra y puede ir seguida por cualquier cantidad de letras o enteros y una subcadena puede ser op1, op2, op3. La declaración de funciones puede ser recursiva (funciones dentro de funciones) por lo que decl-function representa la declaración de una función dentro de la actual.

id/var

$S \rightarrow T \text{ function } N (\Lambda; L) \text{ declare } \{ D \} \{ \text{begin } \{ E \} \}$

$T \rightarrow \text{int} / \text{char}$

$\Lambda \rightarrow T : N$

$L \rightarrow ; \Lambda L / \Lambda$

$D \rightarrow \underline{\Lambda M} / \underline{SM}$

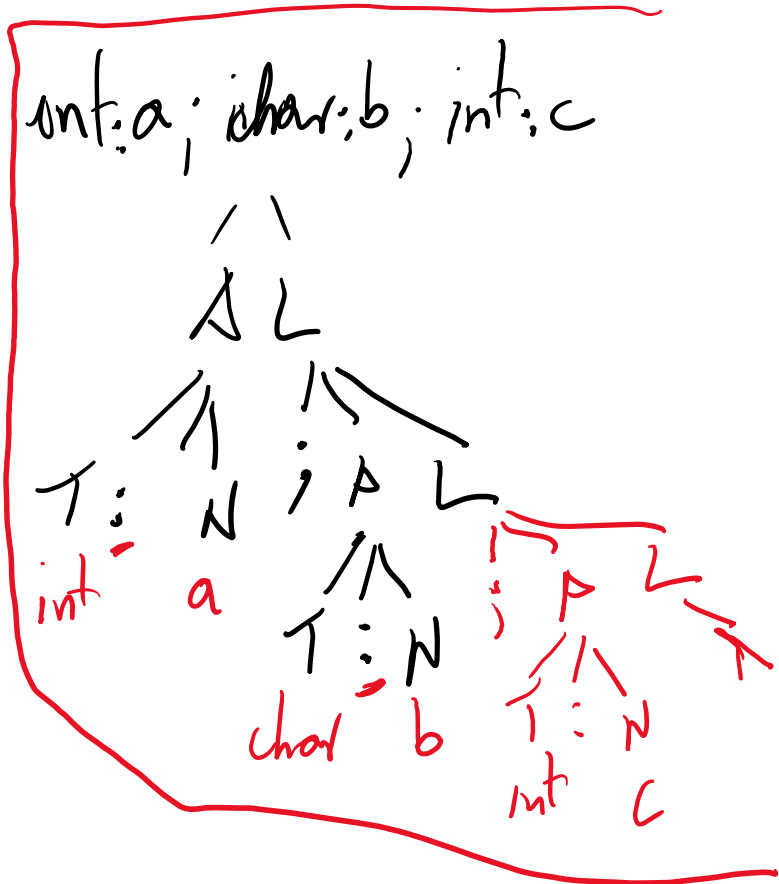
$M \rightarrow ; \underline{NM} / \cancel{DM} / \underline{\Lambda}$

$N \rightarrow \text{id}$

$E \rightarrow E \text{ op } E / (E) / \Lambda$

• declare { int:a; b; c } {

• declare { int function k... }



declare { { { ...



id  
(a)

declare h int function h (char c) declare.

declare h D h

