

Compiladores 2020-1

Facultad de Ciencias UNAM

Práctica 7: Generación de Código en C

Lourdes del Carmen González Huesca Alejandra Krystel Coloapa Díaz
Javier Enríquez Mendoza

Integrantes del equipo

Ángeles Martínez Ángela Janín	314201009
García Landa Valeria	314033008
Martínez Monroy Emily	314212391
Rebollar Pérez Ailyn	314322164

Desarrollo de los Ejercicios

Ejercicio 1

Para este ejercicio lo que hicimos fue definir un nuevo lenguaje que se llama L13, donde eliminamos las listas y a continuación el agregamos dos nuevos constructores, uno de ellos es el de array y en los terminales agregamos el constructor len. Nuestro lenguaje L13 quedó de la siguiente manera:

```
1 (define-language L13
2   (extends L12)
3   (terminals
4     (+ (len (len))))
5   (Expr (e body)
6         (- (list e* ...))
7         (+ len
8           (array len t [e* ...]))))
```

Después realizamos el proceso de *list-to-array* pero para ello, volvimos a definir el algoritmo J de tal manera que ahora recibiera expresiones del lenguaje L12 y se modificó el caso de las lambdas donde ahora no están currificadas, entonces se hizo una función auxiliar donde fuimos construyendo los pares que se metían al contexto porque ahora tenemos una lista de variables y de tipos, a esa función se llama *pares*.

Sucesivamente, como no cambian las demás expresiones con unparse y parser adaptamos para ahora tener una expresión en L13.

Ejercicio 2

Para éste ejercicio decidimos regresar cadenas en lugar de s-expressions, por lo que declaramos una nueva función que se llama c y usamos nanopass-case y aún seguimos trabajando en el ejercicio.