

## Tarea 06

Navarro Meléndrez Erick Joel

October 2022

**4.8.** Mostrar exactamente en donde nuestra implementación del store de operaciones toma tiempo lineal en lugar de constante.

*En newref, las funciones append y length tienen una complejidad del tiempo lineal y por extensión, newref también.*

*En deref, la función list-ref tiene una complejidad del tiempo lineal y por extensión deref también.*

*En setref, setref-inner devuelve una lista por lo que su complejidad del tiempo debe ser lineal.*

**4.9.** Implementar un store en tiempo constante representándolo con un vector esquema. ¿Cuál es la pérdida por usar esta implementación?

Revisar archivo "lang-4-9.rkt"

**4.10.** Implementa la expresión begin como se especifica en el ejercicio 4.4.

Revisar archivo "lang-4-10.rkt"

**4.11.** Implementa list del ejercicio 4.5.

Revisar archivo "lang-4-11.rkt"

**4.12.** Nuestro entendimiento del store, como se expresa en este intérprete, depende del significado de los efectos del esquema. En particular, depende de nosotros saber cuando estos efectos toman lugar en el programa esquema. Podemos evitar esta dependencia escribiendo un intérprete que imite más cercanamente a la especificación. En este intérprete, value-of devolverá tanto el valor como el store, justo como en la especificación. Un fragmento de este intérprete aparece en la figura 4.6. A esto le llamamos store-passing interpreter. Extiende este intérprete para cubrir todo el lenguaje EXPLICIT-REFS.

Cada procedimiento que podría modificar el store devuelve no solo su valor sino también un nuevo store. Estos son empaquetados en un tipo de dato llamado answer. Completa esta definición de value-of.

Revisar archivo "lang-4-12.rkt"

**4.13.** Extiende el intérprete del ejercicio precedente para tener procedimientos con múltiples argumentos. Revisar archivo "lang-4-13.rkt"

Ejercicio no finalizado