

## Anexo I - Etapa IV: Completando la Máquina Virtual de Stókhos

La sesión de REPL en la página siguiente pone en evidencia como los ciclos de cómputo afectan la evaluación de variables. Entender el rol de la memoización, junto con los conceptos de l-value, c-value, y r-value, es *clave* para entender la semántica del lenguaje Stókhos, y por lo tanto la traza. Por esto, Ustedes deben *primero entender el material* provisto en las láminas del 8 de abril y el enunciado de la etapa IV. Después de hacerlo, pueden seguir la traza y verificar su comprensión. Si su reacción es “Si, claro que se debe comportar así” están bien, de otra forma vuelvan a leer y discutir el material con cuidado.

El “prompt” del REPL tiene la forma  $\langle K: N/M \Rightarrow R \rangle$ , donde K, N, M, y R son números enteros. El significado de esos números solo tiene que ver con la auditoría de mensajes síncronos y asíncronos entre el REPL y la VM. La forma del “prompt” es *irrelevante* para entender la traza y para Ustedes. En su proyecto, el REPL y la VM comparten de el mismo proceso: no hay comunicación entre dos procesos, posiblemente corriendo en computadoras distintas.

La función **sys.tick** es *igual* a la función **tick** que Ustedes deben implementar. La única diferencia es que para Ustedes esa función es el estado una variable global (predefinida) llamada **tick**, en vez de ser el estado del campo **tick** de una tupla con campos nombrados llamada **sys**. La tupla **sys** es obviamente es una variable predefinida global.

La prueba más fuerte de que han entendido el material es saber explicar no sólo el “mecanismo” – es decir, cómo la memoización hace que Stókhos se comporte de la manera observada en la traza - sino la “razón” – es decir, por qué esa semántica es deseable.

Las dos formas de implementar un lenguaje son traducirlo a otro o interpretarlo. Este curso se llama “Traductores e Interpretadores” porque la idea es que entiendan, a un nivel básico, pero no superficial, como se implementan estas herramientas de programación. El objetivo principal del laboratorio es adquirir la capacidad de entender la definición de un lenguaje e implementarlo correctamente, respetando su sintaxis y semántica.

La gran mayoría de Ustedes nunca van a tener que implementar un lenguaje de programación en su carrera profesional. Pero puede que varios tengan que implementar interfaces avanzadas y complejas entre sistemas de cómputo o entre humanos y computadoras. Cada vez que hacen esto están definiendo e implementando un lenguaje.

Este laboratorio es clave en la carrera: hacerlo bien les ayuda a desarrollar su conocimiento de la ciencia de la computación y mejores prácticas de ingeniería de software.

<12: 5/5 => 5> num x := 'floor(100 \* uniform())';

ACK: num x := 'floor(100 \* uniform())';

<14: 6/6 => 6> x

OK: x ==> 27

<17: 7/7 => 7> x

OK: x ==> 27

<18: 8/8 => 8> x

OK: x ==> 27

<19: 9/9 => 9> sys.tick();

ACK: sys.tick();

<20: 10/10 => 10> x

OK: x ==> 38

<21: 11/11 => 11> x

OK: x ==> 38

<22: 12/12 => 12> x

OK: x ==> 38

<23: 13/13 => 13> sys.tick();

ACK: sys.tick();

<24: 14/14 => 14> x

OK: x ==> 2

<25: 15/15 => 15> x

OK: x ==> 2

<26: 16/16 => 16> >>

ACK: sys.tick();

<27: 17/17 => 17> x

OK: x ==> 5

<28: 18/18 => 18> x

OK: x ==> 5

<29: 19/19 => 19> >>

ACK: sys.tick();

<30: 20/20 => 20> x

OK: x ==> 57

<31: 21/21 => 21> x

OK: x ==> 57