

Teoría de Lenguajes

Teoría de la Programación

Clase 4: Técnicas de programación declarativa & Concurrencia



Problema 1

Desarrollar un procedimiento que dadas dos listas ordenadas en forma ascendente las mezcle en forma ordenada

Problema 2

Desarrollar un procedimiento que dada una lista la subdivide en 2 listas.

Problema 3

Con lo desarrollado anteriormente, hacer un mergesort.

Tarea para el hogar: Pensar complejidad espacial y temporal del Mergesort

Pensar recursivamente

Una lista es una estructura de datos recursiva. Una función que calcula sobre listas tendrá:

- Caso base
- Caso recursivo

Convertir recursividad a computaciones iterativas! => Tail recursion

- Accumulators!!

High order programming

High order programming

Conjunto de técnicas de programación que se pueden utilizar cuando los procedimientos pueden ser tomados como valores

Un procedimiento cuyos parámetros no son procedimientos, se considera de primer orden

First class citizen

Un “ciudadano de primera clase” es una entidad que soporta todas las operaciones que soporta cualquier tipo básico

En el modelo declarativo, los procedimientos, las funciones, son ciudadanos de primera clase

¿Qué podemos hacer con
high order?

Operaciones básicas

- Abstracción procedural
- Genericidad
- Instanciación
- Embedding

Abstracción procedural

Cualquier statement puede ser convertido a un procedure value

- Statements diferidos
- Parámetros

Genericidad

Enviar procedimientos como parámetro permite realizar procedimientos genéricos.

1. Sumar todos los elementos de una lista
2. Multiplicar todos los elementos de una lista
3. ¿Cómo lo hacemos genérico?

Instanciación

Devolver un procedure value como resultado de una expresión.

1. Revisar el MergeSort. ¿Cómo lo hacemos genérico?
2. Hacer un factory de esos sorts

Embedding

Poder utilizar procedure values como parte de una estructura de datos

Ej: Lo que vimos como clases en el capítulo 1

Funciones anónimas

Definición de una función sin necesidad de usar un identificador de variable

A veces llamadas funciones lambda

Currying

Técnica mediante la cual todos los procedimientos reciben solo un parámetro.

Si tienen más de un parámetro, en realidad es tomado como una función que tiene a su vez un solo parámetro

Currying

Algunos lenguajes soportan curry en forma natural

Permite evaluar funciones en forma parcial



Haskell Curry (1900 -1982)

Loops

Nueva abstracción lingüística: Declarative loops

- Integer loop : {For 1 10 1 Browse}
- List loop: {ForAll [a b c] Browse}

Imperative vs Declarative loops

Cada iteración declara una nueva variable

Cada iteración es independiente de la otra => Podrían ejecutarse en forma concurrente

Concurrencia

Concurrencia

Actividades independientes

La ejecución no es secuencial, sino que se va solapando

No es en paralelo => otro procesador

Concurrencia declarativa

Determinístico

Se hacen cálculos en forma incremental

Ejemplo

```
local X0 X1 X2 X3 in
  thread X1=1 + X0 end
  thread X3=X1 + X2 end
  X0=4
  X2=2
  {Browse [X0 X1 X2 X3]}
end
```

Threads

Hilo independiente de ejecución

Tipos de threads dependiendo el contexto

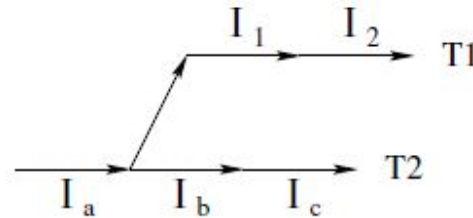
- Heavyweight: diferente proceso de SO
- Middleweight: mismo proceso, kernel level
- **Lightweight: User level - green threads**

Interleavings - Order

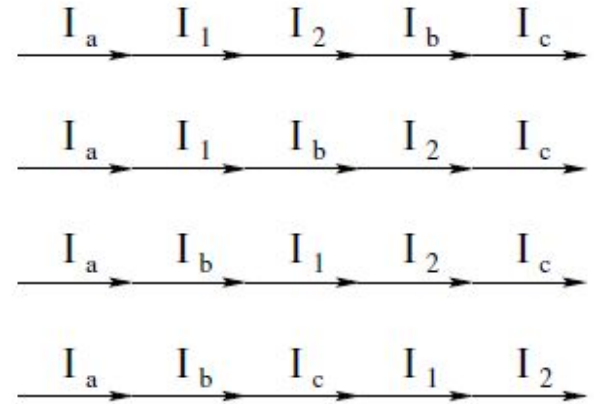
Los threads se ejecutan en forma intercalada (interleaving). No se solapan los computos

Total order

Causal order



Causal order



Some possible executions

Scheduler

Un scheduler define cual es el próximo thread a ejecutarse

Un thread al igual que un programa puede tener estados

- Listo
- Suspendido

El scheduler debe ser justo: Evitar starvation

Sintaxis

Tenemos un nuevo statement válido

```
thread <s> end
```

Máquina abstracta

(ST, σ)



(MST, σ)

$$MST = \{ST_1, ST_2, \dots, ST_N\}$$

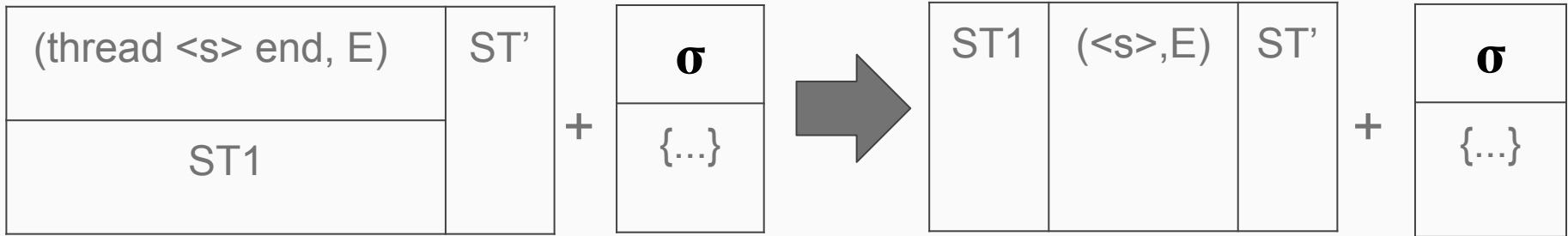
statement
 $(\{ [\underbrace{(\langle s \rangle, \phi)}_{\text{stack}}] \}, \phi)$
stack
multiset

Semántica

En el tope de un ST tenemos el siguiente semantic statement (thread <s> end, E)

Se crea un nuevo ST que se agrega al MST con el tope de pila (<s>,E)

El σ sigue siendo único



Estados - Manejo de memoria

Un ST finalizado puede ser eliminado

Un ST bloqueado puede ser eliminado si la condición de activación es inalcanzable

Cheap concurrency & dataflow

Ejercicio

Hacer un Map pero que funcione en forma concurrente

Bibliografía

- **Concepts, Techniques, and Models of Computer Programming - Capítulo 3**, Peter Van Roy and Seif Haridi
- **Concepts, Techniques, and Models of Computer Programming - Capítulo 4**, Peter Van Roy and Seif Haridi
- **Extras:**
 - **Módulos e interfaces de Oz**

<https://mozart.github.io/mozart-v1/doc-1.4.0/tutorial/node7.html>
 - **Principles of Concurrent and Distributed Programming**, M. Ben-Ari.
 - **High order & Currying con Haskell**

<http://learnyouahaskell.com/higher-order-functions>