Motivation

The present and the future: In the near future, manufacturers will move to a manycore paradigm, where hundreds, or even thousands, of cores on a single chip are expected.

Multicores everywhere!
Quadcore in your pocket!

Writing programs that scale with increasing number of cores should be as easy as writing programs for sequential computers.

Asanovic et al. [1]

Our aim

Design and implement a new object-oriented programming language that combines binary session types, ownership types and size annotations to calculate communication costs.

Programming with STOPS: Session Types, Ownership & Sizes

Juliana Franco, Sophia Drossopoulou and Nobuko Yoshida | {j.vicente-franco,s.drossopoulou,n.yoshida} @imperial.ac.uk | Department of Computing

Background: session types [3, 5]

- Session types describe message-passing communications.
- Session types make possible to ensure communication safety, absence of deadlocks and race conditions through static checking.

Background: ownership types [2, 4]

- In object oriented languages, aliasing is a dangerous and powerful feature.
- Ownership types were first introduced as flexible alias protection in order to limit the changes of objects via aliases.
- Each object owns zero or more objects and each object has a single owner.
- Applications: concurrency without race conditions, parallelism without locks, object cloning, garbage collection . . .

The heap topology for StudentList<root | 3> is:

(1) Extend the ownership types in order to express the size of data structures

```java
class StudentList{o} {  
   @Has up_to N: Node<this> | head;  
   class Node<o> {  
      Node<o> | next;  
      class Student<o> {  
         String name; integer age;  
      }  
   }  
}
```

(2) Extend the session types to express the number of repetitions of a given behaviour

```java
session Professor<o | u, n> =  
   rec a, l:integer. ?StudentList<o | n>. a[u]  
```

(3) The communication cost of a channel governed by the Professor Session type is:

- The cost of sending $u$ values of type integer
- plus the cost of receiving $u$ objects of type StudentList<o | n> (the cost of sending $n$ objects of type Node).

(4) Further questions:
Do we copy the students or do we send the reference? How many students does the professor receive?

References


Multicores everywhere!
Quadcore in your pocket!

Nexus 5 equipped with quadcore CPU (http://www.google.com/nexus/5/)

Tianhe-2, developed by China’s National University of Defense Technology, is the world’s no 1 super computer since June 2013, with a total of 3,120,000 cores according to http://www.top500.org

http://www.upscale-project.eu/