Cost Model based on the λ-Calculus
or
The Church Calculus the Other Turing Machine

Guy Blelloch
Carnegie Mellon University

July 9, 2018

1 Parallelism vs. Concurrency

1.1 Parallelism

Provides performance
Goal: Make parallel algorithms map sequential algorithms (no difference)
Don't change the semantics, just integrate a cost model.

2 λ-Calculus with Cost Model

Church/Turing Hypothesis: $(\lambda x.e_1)e_2 \;\Rightarrow_\beta\; e_1[e_2/x]$

2.1 Machine Models and Simulation

2.1.1 Handbook of Theoretical Computer Science

Chapter 1: Machine Models and Simulations [Peter van Emde Boas]
In order to measure time and space, then need to specify which notions of time and space are meant.
Can use abstract machines to do this (i.e.: RAM Machine) $\leftarrow$ General convection. We want to use a language-based cost model (okay because abstract).
However, the arbitrariness of model choice remains. Need a way to translate between the machines to compare **mutual simulations**.
Example: with a machine with unbounded registers can run exponential numbers of calculations simultaneously ($\implies P = NP$). But this is not allowed, bits must be limited.

2.1.2 Machine Models

Random Access Machines:

- SRAM - succ, pred
- RAM - add, sub
- MRAM - add, sub, mult
- LRAM - log length words
- RAM-L - cost of instruction is word length
- SMM, KUM, pure, impure

Parallel Machine Models:

- Circuit models
- PSPACE
- TM with alternation
- Vector models
- PRAM - EREW, CREW, CRCW (priority, arbitrary, . . .)
- SIMDAG
- k-PRAM, MIND-RAM, PTM

2.1.3 The two parts

Part 1: The model
  Well defined semantics
  simple
  close to programming paradigm
Part 2: Simulation
  mapping of costs
  good bounds when simulated on realistic machines

2.1.4 Language-Based Cost Model

A cost model base on a "cost semantics" instead of a machine
Why use the λ-calcusus? Historically the first model, clean, well understood
What costs?
Advantages:

- naturally parallel (turing model not parallel)
- more elegant
- model is closer to code and algorithms

- closer - in terms of simulation costs - to ”practical” machine models such as the RAM

Disadvantages:

- 50 years of history

2.1.5 CBV λ-Calculus

$$e = x|(e_1, e_2)|\lambda x.e \tag{expression}$$

Lambda Calulus Sequential without Cost:

$$e \Downarrow v \tag{relation}$$

$$\lambda x.e \Downarrow \lambda x.e \tag{λ}$$

$$\frac{e_1 \Downarrow \lambda x.e \;\; e_2 \Downarrow v \;\; e[v/x] \Downarrow v'}{(e_1 e_2 \Downarrow v'} \tag{App}$$

Lambda calculus parallel with cost:

$$e \Downarrow v; w, d \tag{relation-par-cost}$$

$$\lambda x.e \Downarrow \lambda x.e; 1, 1 \tag{λ-par-cost}$$

$$\frac{e_1 \Downarrow \lambda x.e; w_1, d_1 \;\; e_2 \Downarrow v; w_2, d_2 \;\; e[v/x] \Downarrow v'; w_3, d_3}{(e_1 e_2 \Downarrow v'; 1 + w_1 + w_2 + w_3, 1 + max(d_1, d_2) + d_3} \tag{App-par-cost}$$

2.1.6 Simulation (sequential)

CEK Machine: a state transition system

$$(C, E, K) \Rightarrow (C', E', K')$$

”control” $C := e_1 e_2|\lambda x.e|x$
”environment” $E := x \to v$
”continuation” $K :=$
Everything is constant time except perhaps look-up and insert.