CS-377: Parallel Programming
Assignment 6 Lab Report

Eliot Cowley

May 12, 2015

I had a lot of trouble with this assignment at first. Besides not really knowing much Ruby, I had never used Rinda or any implementation of Linda before, and I was very confused by the examples that Professor Smith had given us. It took me a while to really understand what tuple space is and how writing, reading, and taking work.

The syntax threw me off as well. The multitude of brackets and parentheses and lack of curly braces was a change from all the Java programming I'd been doing recently. How to access values in tuples, getting the correct one by searching for a type such as Numeric or using a subscript, was something I had to get used to.

But once I started to understand these concepts and get a grasp on the language, the exercises became much smoother. Looking back, I can see why Rinda might be a popular tool that programmers use to run their programs in parallel. Its usage of take and write correspond almost perfectly with waiting and signaling semaphores.

I thought these exercises were a good introduction to Rinda. It helped that we were given pseudocode to model our programs after, because without it I would have been even more lost. The producer/consumer problem is also one that we're already familiar with, so it wasn't hard to conceptualize. However, I would have liked to see some more sample Rinda code other than the code from the Wikipedia page that we were given. It wasn't all that helpful for me, and I ended up having to research on my own and collaborate with Henry to try and figure out the correct syntax.

I'm still not sold on Ruby. The syntax isn't nearly as intuitive or easy for me as the developers like to proclaim it is. But it is a popular and useful programming language, and I can see the need for something like Rinda. I think for now, I will be sticking with synchronized methods and code blocks in Java.