Massachusetts Institute of Technology Handout 8
6.852: Distributed Algorithms
Prof. Nancy Lynch February 28, 2008

The Tempo Toolkit

Tempo Basics

To precisely model and prove properties of distributed algorithms in an asynchronous network, we introduced the Input/Output Automata (IOA) formalism. The Tempo Toolkit is a collection of tools to simplify the process of developing and validating IOA systems. (Tempo actually works with Timed Input/Output Automata, a more powerful version of IOA that includes timing constraints. For our purposes, we will not make use of the timing features.)

The toolkit consists of the Tempo language, which closely matches the format of the pseudocode used in the book to describe IOA, a syntax checker to validate your IOA code, as well as connections to a homegrown simulator, the PVS theorem prover, and the UPAAL model-checker.

Your Requirements

When specified by the problem set, you will be required to write your distributed algorithm in the Tempo language and validate it with the syntax checker. You will *not* be required to use the other tools included with the toolkit—simulation, theorem proving, model-checking—but are welcome to try if you have some experience with these techniques. We recommend that you experiment with writing your code in Tempo for the relevant questions in part b of problem set 2. We will *require* you to hand in syntax-checked Tempo code starting with problem set 3.

Installing the Tempo Toolkit

Windows, Mac, and Linux version of the toolkit can be found at http://www.veromodo.com/tempo/. We recommend that you download the version including the Eclipse-based user interface. For this to work, you must have Java version 1.5 or better. The latest versions of Java can be download at: http://www.java.com/en/download/index.jsp.

Within the install directories see `doc/Tempo_Guide.pdf` for an overview of both the TIOA formalism and the Tempo language. Section 3 provides some toy examples of TIOA algorithms implemented in Tempo. (If you run a 64-bit machine, contact Calvin to see if a 64-bit compile can be made available.)