COMS W4995-02
Languages for Embedded System Design
Homework 3

Prof. Stephen A. Edwards
Columbia University

Assigned October 31th, 2001
Due November 14, 2001

You may submit the solutions either on paper or electronically, but not both. If you submit it electronically, send a single file that is text, PostScript, PDF, or Word. Don't send multiple files or an archive such as tar or zip. I just print out whatever you submit and read it; I don't try to run the programs. Make sure your name appears at the beginning of the file you send.

As always, I want the paper or electronic versions at the beginning of class (9:40 EDT) on the due date. This applies to both on-campus and CVN students.

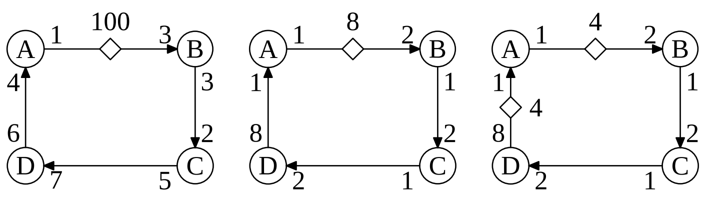
1. **(10 points)** Book, Exercise 11-1: Would adding an empty() command that reported whether a channel had waiting tokens change the semantics of Kahn Process Networks?

2. **(10 points)** Book, Exercise 11-3: The diagram in Figure 11.2 has a finite number of states. Is this true for every Kahn Process Network, even those that do not require unbounded buffer space?

3. **(5 points)** Book, Exercise 12-3: (a) Can an SDF system satisfy the balance equations and not be able to run forever? (b) Can an SDF system violate the balance equations and still run forever?

4. **(10 points)** Can a tree-structured SDF system ever not have a schedule? Why or why not?

5. **(25 points)** For each of the following SDF graphs, find a looped single appearance schedule if it exists, otherwise find a looped schedule if it exists, otherwise explain why the system cannot be scheduled.



(a) (b) (c)

6. **(40 points)** Esterel programming.

   I’ve installed the Esterel compiler on labdien.cc.columbia.edu, part of the cunix cluster (it core dumps on the others because it requires Solaris 2.7, not 2.5). The distribution is in my home directory, i.e., ˜se2007/esterel. For those who prefer to run their own copy, the compiler can be downloaded from www.esterel.org. It is available for Linux, Solaris, and Windows NT.

   Here’s how to compile and run an Esterel program:

```
$ uname -r          # Check OS version, Solaris only
5.7                 # Can’t be 5.5.1
$ cat > hello.strl
module Hello:
input A; output B;
await A; emit B
end module
ˆD
$ ˜se2007/bin/esterel -simul hello.strl
$ cc -o hello hello.c ˜se2007/lib/libcsimul.a
$ ./hello
Hello> ;            # Clock tick with no inputs
--- Output:
Hello> ;
--- Output:
Hello> A;           # Make input A present and tick
--- Output: B
Hello> ;
--- Output:
Hello>
```

   (a) Write an Esterel module that implements a two-bit grey-code up counter. It should have two inputs, C, which increases the count, and R, which resets the counter; and two outputs B0 and B1 that should always indicate the current count. The new value should appear when C is present, and R should take precedence over C.

   (b) Add a user interface to your little counter: Make it advance once every 5 arrivals of a “T” signal, and add an alarm signal “A” that is set off when the count reaches a given setting. Set the alarm with three environmental signals: S, S0, and S1. When S is present, S0 and S1 contain the new setting. Thus, your program should behave like this:

```
S S1;          # Set the alarm to 10
T; T; T; T; T; # Count is now 01
T; T; T; T; T; # Count is now 10 and A is present
```

   For both programs, show me the source and an example run that tests it.