EECS 638
Spring 2014

Programming Project
Expert system to control traffic lights at an intersection

Due: midnight from April 18 to April 19, 2014

In this project you are to develop a program in CLIPS in which knowledge is coded by rules. Your program should be able to control traffic lights at an intersection. The *intersection* is a crossing of two-way roads with two lanes in both ways. One of the roads will be referred to as *west-east*, the other as *north-south*. Any car, approaching the intersection, will be detected by one of the eight *sensors*, located pairwise (for sensing incoming traffic in both lanes) at four sides of the intersection. It is assumed that sensors are sensitive to the direction of a car, i.e., only cars approaching the intersection will be detected, while cars leaving the intersection are ignored by a sensor. Only the first car in the line to the intersection will be detected. The system controlling the lights is equipped with a *clock*. The clock generates a *pulse* every 10 seconds. The signal from sensors is observed only at moments when clock pulses are generated.

When no cars have approached the intersection, a normal cycle of traffic lights is used: Between the first and fifth pulses (i.e. for 40 seconds), the lights are green for one direction, say west-east, and red for the other direction, i.e., north-south. Between the fifth and sixth clock pulses, i.e., for 10 seconds, lights are yellow for the west-east direction and red for the north-south direction. With the sixth clock pulse, lights are red for the west-east direction and green for the north-south direction for the next 40 seconds, etc.

If in any direction cars are still incoming at the end of the green cycle and there are no cars in the perpendicular direction, the green light should be extended as long as this situation continues, i.e., cars are still incoming and no cars from the perpendicular direction.

When a car from any direction has approached the intersection, say west-east, lights for this direction are red and no cars have been detected from the other direction, then with the first clock pulse the lights for the north-south direction should be yellow. After second clock pulse, the lights for the west-east direction should become green. If after that event no new car is detected, the regular cycle continues, i.e., after the next 40 seconds green lights become yellow, etc. With a new car approaching the intersection, the regular cycle may be interrupted again, as previously described.

If two cars are detected from the same direction, e.g., the first car has approached the intersection from the west and the second from the east, the situation is reduced to the case of a single car that has approached the intersection from either west or east.

When from both different directions cars are detected (e.g., eastbound and southbound), the cycle will continue as usual, without any interruption.

The input data file will have eight binary digits (each digit has value either 0 or 1) per line. Digits will be separated by one or more spaces. The first pair of digits corresponds to the westbound cars approaching the intersection on two lanes (0 means no car, 1 means that car has approached), the second pair of digits corresponds to eastbound cars, the third pair of digits corresponds to northbound cars, and the fourth pair of digits corresponds to southbound cars. Values in every line are synchronized with a clock pulse, i.e., it is assumed that the line represents a snapshot at the intersection taken every 10 seconds.

The output file (that should be created by your program) will have six binary digits per line, separated by commas. Digits correspond to (1) green lights, (2) yellow lights, (3) red lights in the west-east direction, and (4) green lights, (5) yellow lights, and (6) red lights in the north-south direction. You should assume that lights are green in the west-east direction initially and that your system starts from the very beginning of the normal cycle. Assume that the output is observed in the middle between time clock pulses

Also, your program should generate automatic comments. It should be explained, after every six output digits of your output, on the basis of all past relevant situations, why your program generated such an output.

General Remarks. You may assume that the input data file does not contain errors. Include all comments, including instructions about compiling and linking in a single file called **read.me**. Do not forget to include your name and KUID#. When you are ready to submit the project, send ALL necessary source files, makefile (if any), and the read.me file by e-mail to **s063s478@ku.edu**. **Use *tar* and send your homework as one file. Do not send object files, executable files, and test data files**. Time stamp of the e-mail will be used to decide late penalty, if applicable. Late projects will be accepted with 10% penalty per day up to five days.