1. (60 points) (Problem created by David Broman)

**Update (December 2, 2013):** With the given DAE, it turns out that BLT sorting succeeds in step (b). As a consequence, exercise steps (c) and (f) are therefore removed. Your solution for exercise steps (a), (b), (d), and (e) should be the same. Note also that steps (d) and (e) now refers to (b) and not (c).

Consider the following system of differential-algebraic equations

\[
\begin{align*}
    f_1(\dot{x}_2, \dot{x}_4, x_5) &= 0 \\
    f_2(\dot{x}_2, x_3, \dot{x}_4) &= 0 \\
    f_3(x_1, x_2, \dot{x}_4) &= 0 \\
    f_4(\dot{x}_2, x_4) &= 0 \\
    f_5(x_1, x_2, x_3, \dot{x}_4) &= 0
\end{align*}
\]

(a) Write out the incidence matrix and the bipartite graph of the DAE. Note that only unknown variables should be used in the matrix.

(b) Apply BLT sorting. Give a detailed explanation of what is happening (step by step). Does the algorithm succeed? Why? Why not?

(c) Exercise (c) has been removed.

(d) Has the DAE of step (b) any algebraic loops? If so, which equations are involved?

(e) What is the index of the sorted system in step (b)

(f) Exercise (f) has been removed.
2. (60 points)

Consider the game graph shown below:

This game has two players: Player 0, playing from the square states (1, 3, 5, 7, 8), and Player 1, playing from the round states (2, 4, 6). This is a slightly different type of game from the one we studied in class, in the sense that the moves of the two players do not necessarily alternate. For example, if the game is at state 5, Player 0 may choose to move to state 8, in which case it’s again her turn to play.

(a) Can the above game be transformed to an equivalent game where the two players strictly alternate (i.e., from a square state we can only move to a round state, and vice versa), and how?

(b) How would you modify the algorithms to solve safety, reachability, and deterministic Büchi games directly on the above type of games, without having to transform them first to games where the two players strictly alternate?

(c) Which algorithm would you use to compute the winning states for Player 1 to avoid “bad” states 7 and 8? Show the corresponding sequence of sets of states computed by the algorithm, and the strategy for Player 1.

(d) Which algorithm would you use to compute the winning states for Player 1 to reach target state 1? Show the corresponding sequence of sets of states computed by the algorithm, and the strategy for Player 1.

(e) Which algorithm would you use to compute the winning states for Player 1 to keep visiting the set of states \(\{4, 5\}\) infinitely often? Show the corresponding sequences of sets of states computed by the algorithm, and the strategy for Player 1.

This game has two players: Player 0, playing from the square states (1, 3, 5, 7, 8), and Player 1, playing from the round states (2, 4, 6). This is a slightly different type of game from the one we studied in class, in the sense that the moves of the two players do not necessarily alternate. For example, if the game is at state 5, Player 0 may choose to move to state 8, in which case it’s again her turn to play.

(a) Can the above game be transformed to an equivalent game where the two players strictly alternate (i.e., from a square state we can only move to a round state, and vice versa), and how?

(b) How would you modify the algorithms to solve safety, reachability, and deterministic Büchi games directly on the above type of games, without having to transform them first to games where the two players strictly alternate?

(c) Which algorithm would you use to compute the winning states for Player 1 to avoid “bad” states 7 and 8? Show the corresponding sequence of sets of states computed by the algorithm, and the strategy for Player 1.

(d) Which algorithm would you use to compute the winning states for Player 1 to reach target state 1? Show the corresponding sequence of sets of states computed by the algorithm, and the strategy for Player 1.

(e) Which algorithm would you use to compute the winning states for Player 1 to keep visiting the set of states \(\{4, 5\}\) infinitely often? Show the corresponding sequences of sets of states computed by the algorithm, and the strategy for Player 1.
3. (30 points)

Consider the computation of $\text{RecAcc}$ on slide 55 of the synthesis lectures. Consider the inner iteration which computes $\text{Revisit}$ (“framed” part). This is very similar to, but not exactly the same with, the reachability game algorithm on slide 39. What is the difference? Would it be correct to modify the computation of $\text{Revisit}$ to make it identical to the reachability game algorithm? If yes, provide a proof. If not, provide a counter-example.

4. (30 points)

Consider the non-deterministic Büchi game below (this is the same as the example of slide 57 of the synthesis lectures, with controller states numbered 1,2,3, etc.).

Apply the deterministic Büchi game algorithm to this example. Show the sequences of sets $\text{RecAcc}$ and $\text{Revisit}$ computed by the algorithm. Does the algorithm result in a controller?