Homework 1, Finite Model Theory

1. Consider finite strings over the alphabet $\Sigma = \{a, b\}$. Express the following properties using Monadic Second Order Logic over strings. Construct the corresponding automata and compare them to the automata you would have constructed directly by looking at the properties.

   (a) There is at least one occurrence of a.

   (b) There is at least one occurrence of b such that an a occurs before.

2. Let a *path-like* graph of length p be a graph over the set of vertices $\{u_1, v_1, \ldots, u_p, v_p\}$ such that the only possible edges are (u_i, v_i), (u_i, u_{i+1}), (u_i, v_{i+1}), (v_i, u_{i+1}) and (v_i, v_{i+1}) for $i \in [p]$. For such a graph G, let G_i be the subgraph induced by G on $\{u_1, v_1, \ldots, u_i, v_i\}$. Design a dynamic programming algorithm to check if such a graph G is 3-colorable. The algorithm would build a table indexed by $i = 1, \ldots, p$. The i^{th} entry of the table would contain information needed to check if G_i is 3-colorable as well as information needed to compute the $(i+1)^{\text{th}}$ entry in the next round.

   Next suppose that a path-like graph of length p is represented by a string of length $(p-1)$. The i^{th} letter a_i of the string is a subset of $\{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$.

a_i has this	if this is an edge in G
$(1,2)$	(u_i, v_i)
$(1,3)$	(u_i, u_{i+1})
$(1,4)$	(u_i, v_{i+1})
$(2,3)$	(v_i, u_{i+1})
$(2,4)$	(v_i, v_{i+1})
$(3,4)$	(u_{i+1}, v_{i+1})

   Construct an automaton that accepts a string of such letters exactly when it satisfies the following conditions.

   (a) Successive letters are consistent (if a_i says there is an edge between u_{i+1} and v_{i+1}, then a_{i+1} should not contradict this).

   (b) The path-like graph represented by the string is 3-colorable.

   The states of the automaton would correspond to the possible entries of the table constructed by the above dynamic programming algorithm. The transitions of the automaton would correspond to the way the algorithm constructs the $(i+1)^{\text{th}}$ entry by looking at the i^{th} entry.

3. Given a 3-CNF formula, consider the following *incidence graph* associated with it. There is one vertex for every clause and one vertex for every Boolean variable occurring in the formula. There is an edge labeled O

from a clause vertex C to a variable vertex x if the variable x occurs in the clause C. There is an edge labeled $\overline{O}$ from a clause vertex C to a variable vertex x if the negation of x occurs in the clause C. Following is the syntax for Monadic Second Order Logic on such graphs.

$$\phi ::= x = y \mid \text{Clause}x \mid \text{Variable}x \mid Xx \mid Oxy \mid \overline{O}xy \mid \neg\phi \mid \phi \vee \phi \mid \exists x \phi \mid \exists X \phi$$

Intuitively, Clause$x$ means that the vertex x represents a clause and Variable$x$ means that the vertex x represents a Boolean variable. Likewise, Oxy means that there is an edge labeled O from the vertex x to the vertex y. Similarly, $\overline{O}xy$ means that there is an edge labeled $\overline{O}$ from the vertex x to the vertex y.

Write a MSO sentence ϕ such that any 3-CNF formula F is satisfiable iff its incidence graph G_F satisfies ϕ.