1 Monadic second order logic

1.1 DFA to MSO (6=3+3 points)

Consider the DFAs $A$ and $B$ in Figures 1 and 2 over the alphabet $\{a,b\}$.

Provide a description in Monadic Second Order logic of the languages described by the DFAs $A$ and $B$. Before writing your logical formulas, describe in plain English the languages accepted by the two automata.
1.2 MSO$_0$ to DFA (6=3+3 points)

In this section you’ll be given MSO$_0$ formulas over the alphabet \{a, b\} with \(n\) free (second-order) variables and you have to build the corresponding DFA over the alphabet \(\Sigma \times \{0, 1\}^n\). If the free variables are \(X_1, \ldots, X_n\), please follow the convention that a symbol \(\alpha = a, b_1, \ldots, b_n\) is such that the bit \(b_i\) denotes whether the symbol \(\alpha\) belongs to the set \(X_i\).

Please provide the DFA equivalent to the following two formulas.

A \(\exists X_2. X_1 \subseteq X_2 \land a(X_1) \land \text{Sing}(X_2)\)

B \(a(X_1) \land b(X_2) \land X_2 = X_1 + 1\)

1.3 Succinct MSO formulas (8=6+2 points)

We saw in class how it is possible to use MSO to check that a string has even length using two second order variables \(X_{\text{odd}}\) and \(X_{\text{even}}\) that are forced to respectively contain the odd and even positions in the string.

\[
\exists X_{\text{odd}} X_{\text{even}}. \\
(\forall z. (\text{first}(z) \rightarrow z \in X_{\text{odd}}) \land (\text{last}(z) \rightarrow z \in X_{\text{even}})) \\
(\forall x, y. y = x + 1 \rightarrow (x \in X_{\text{odd}} \leftrightarrow y \in X_{\text{even}}))
\]

However, we also observed that the same transformation can be done with only 1 second-order variable

\[
\exists X. \\
(\forall z. (\text{first}(z) \rightarrow z \in X) \land (\text{last}(z) \rightarrow z \not\in X)) \\
(\forall x, y. y = x + 1 \rightarrow (x \in X \leftrightarrow y \not\in X))
\]

Part 1 Show that 2 second-order variables (EDIT: and at most 2 first-order variables) are enough for checking whether the length of a string is divisible by 4;

Part 2 What general principle can you infer from your construction to check that the length of a string is divisible by \(n\)?
2 Reduced Ordered Binary Decision Diagrams (12 points)

2.1 Part 1 (6=4+2 points)

1) Let \((z_1, z_2, z_3, z_4, z_5)\) be Boolean variables (in this given order). Depict the ROBDD for the majority function

\[ MAJ([z_1 = b_1, z_2 = b_2, z_3 = b_3, z_4 = b_4, z_5 = b_5]) = 1 \iff b_1 + b_2 + b_3 + b_4 + b_5 \geq 3 \]

2) Is there an optimal order for the variables? If so which one? If not, explain why.

2.2 Part 2 (6 points)

Given two ROBDDs \(A\) and \(B\) over the same variables and with the same variable ordering, give a linear time algorithm to check whether \(A\) and \(B\) describe the same predicates. Please provide a correctness argument and also explain your reasoning in words.

3 Boolean algebra (10 points)

Prove that the theory of union of intervals forms a decidable Boolean Algebra. Concretely, the domain of the algebra is the set of (positive and negative) integer numbers \(Z\) and each predicate in the algebra is a finite union of intervals \(\varphi = [l_0, u_0] \cup \ldots \cup [l_n, u_n]\) such that for every \(0 \leq i \leq n\), \(l_i \in Z \cup \{-\infty\}, \) and \(u_i \in Z \cup \{\infty\}\). A number \(k \in Z\) is accepted by a predicate \(\varphi = [l_0, u_0] \cup \ldots \cup [l_n, u_n]\) \((k \in [\varphi])\), iff there exists an \(i\) such that \(l_i \leq k \leq u_i\).

Formally define this algebra and prove that it is indeed a decidable Boolean algebra:

1. Closed under complement (for every predicate \(\varphi\) the complement is also in the algebra);

2. Closed under intersection (for every two predicates \(\varphi_1, \varphi_2\) the complement is also in the algebra);

3. Decidable (provide an algorithm to check satisfiability).

You can take inspiration from definition 2 of the paper Applications of Symbolic Finite Automata, from Veanes CIAA 13.
4 Symbolic finite automata (8 points)

A symbolic finite automaton (SFA) is complete when for each symbol in the domain there is a transition out of each state of the automaton.

Formally define complete SFAs and show that deterministic symbolic finite automata are closed under completion. Formally, given a deterministic Symbolic Finite automaton $A = (Q, q_0, \Delta, F)$ construct a complete SFA $A' = (Q', q'_0, \Delta', F')$ such that $L(A) = L(A')$. 