Logic, Fagin’s Theorem, and \textbf{NEXPTIME}

Advanced Complexity Homework Assignment: to turn in by Wednesday, Dec. 01, 2010 (formerly, Nov. 24, 2010).

I. Finite Models of First-Order Logic.

The set of \textit{first-order terms} \( s, t, u, v, \ldots \) is defined as the smallest that contains the variables \( x, y, z, \ldots \), and the applications \( f(t_1, \ldots, t_n) \) of a so-called \textit{function symbol} \( f \), of \textit{arity} \( n \in \mathbb{N} \), to \( n \) terms \( t_1, \ldots, t_n \). We assume finitely many function symbols, with given arities. The set \( \text{Var} \) of all variables is assumed countably infinite.

The set of \textit{first-order formulae} \( F, G, \ldots \), is the smallest that contains:

- the \textit{atomic formulae} \( P(t_1, \ldots, t_n) \), where \( P \) is taken among a finite set of so-called \textit{predicate symbols}, of \textit{arity} \( n \in \mathbb{N} \)—each predicate symbol \( P \) comes with a given arity;

- the \textit{conjunctions} \( F \land G \) of two formulae \( F, G \);

- the \textit{negations} \( \neg F \) of formulae \( F \);

- the \textit{universal quantifications} \( \forall x \cdot F \).

We also write \( F \lor G \) for \( \neg(\neg F \land \neg G) \), \( F \Rightarrow G \) for \( \neg(F \land \neg G) \), \( \exists x \cdot F \) for \( \neg\forall x \cdot \neg F \).

A \textit{structure} \( I \) is a non-empty set \( D \) of so-called \textit{values}, together with a total function \( I_f : D^n \rightarrow D \) for each \textit{arity} \( n \) \textit{function symbol} \( f \), and with a subset \( I_P \) of \( D^n \) for each \textit{arity} \( n \) \textit{predicate symbol} \( P \). The \textit{Tarski semantics} \( I[t] \rho \) of terms \( t \) and \( I, \rho \models F \) of formulae \( F \), in an \textit{environment} \( \rho : \text{Vars} \rightarrow D \), is defined as follows.

\[
I[x] \rho = \rho(x)
\]
\[
I[f(t_1, \ldots, t_n)] \rho = I_f(I[t_1] \rho, \ldots, I[t_n] \rho)
\]
\[
I, \rho \models P(t_1, \ldots, t_n) \text{ iff } (I[t_1] \rho, \ldots, I[t_n] \rho) \in I_P
\]
\[
I, \rho \models F \land G \text{ iff } I, \rho \models F \text{ and } I, \rho \models G
\]
\[
I, \rho \models \neg F \text{ iff } I, \rho \models F \text{ does not hold}
\]
\[
I, \rho \models \forall x \cdot F \text{ iff for every } v \in D, I, \rho[x \mapsto v] \models F
\]

We write \( \rho[x \mapsto v] \) for the environment that maps every \( y \) other than \( x \) to \( \rho(y) \), and \( x \) to \( v \).
A sentence is a formula with no free variable, i.e., whose variables are all in the scope of some quantifier. For example, \( \forall x \cdot P(x) \Rightarrow P(x) \) is a sentence, but \( P(x) \Rightarrow \forall y \cdot P(y) \) is not. Whether \( I, \rho \models F \) is true or false is independent of \( \rho \) when \( F \) is a sentence: then we shall simplify the notation to \( I \models F \).

A structure \( I \) is finite, resp. of cardinality 2, iff \( D \) is. The standard representation of a finite structure \( I \) is as follows: first, a natural number \( N \) such that \( D = \{1, 2, \ldots, N\} \); then, each map \( I_f \) is described as a table, where the entry at position \( (v_1, v_2, \ldots, v_n) \) is the value \( I_f(v_1, \ldots, v_n) \); finally, each subset \( I_P \) is described as a truth-table, where the entry at position \( (v_1, v_2, \ldots, v_n) \) is 1 iff \( (v_1, v_2, \ldots, v_n) \in I_P \), 0 otherwise.

1. Show that the following problem FIN-MC is \( \text{PSPACE} \)-complete:
   INPUT: a finite structure \( I \), in its standard representation; a first-order sentence \( F \).
   QUESTION: \( I \models F \)?

   Show that this remains true even if we restrict \( I \) to be of cardinality 2, and \( F \) to have only one predicate symbol \( P \), of arity 1, and where \( F \) does not contain any propositional connective \( (\land, \lor, \neg) \); \( F \) may still use quantifiers \( \forall, \exists \).

2. Given a fixed first-order formula \( F \), the problem FIN-MC(\( F \)) is the following variant:
   INPUT: a finite structure \( I \), in its standard representation.
   QUESTION: \( I \models F \)?

   Can one solve FIN-MC(\( F \)) in polynomial time? If so, for which class of first-order formulae \( F \)?

3. A literal \( L \) is either an atomic formula \( A \) or the negation \( \neg A \) of an atomic formula. A clause \( C \) is a disjunction \( L_1 \lor \ldots \lor L_p \) of literals (or false if \( p = 0 \)). A clausal form \( S \) is a conjunction \( C_1 \land \ldots \land C_m \) of clauses. We understand clauses and clausal forms as implicitly universally quantified over all their (free) variables. So clauses and clausal forms are special cases of sentences.

Let FIN-MC-CLAUSES be the restriction of FIN-MC to the case where \( F \) is a clausal form, i.e.:
   INPUT: a finite structure \( I \), in its standard representation; a clausal form \( S \).
   QUESTION: \( I \models S \)?

   Show that FIN-MC-CLAUSES is \( \mathcal{C} \)-complete, for some class \( \mathcal{C} \) that you should name.

II. Existential Second-Order Logic, and Fagin’s Theorem

Define the formulae of existential second-order logic as:

\[
\exists P_1 : n_1, \ldots, P_m : n_m \cdot F
\]

where \( F \) is a first-order formula, and \( P_1, \ldots, P_m \) are predicate symbols which we shall call predicate variables. The annotations : \( n_1, \ldots, : n_m \) describe the respective arities of these
predicate symbols; i.e., every atomic subformula of $F$ of the form $P_i(t_1, \ldots, t_n)$ must satisfy $n = n_i$. (We shall omit these arity annotations when clear from context.)

Extend the semantics by stating that $I, \rho \models \exists P : n_1, \ldots, P_m : n_m \cdot F$ iff there are subsets $D_1 \subseteq D^{n_1}, \ldots, D_m \subseteq D^{n_m}$ such that $I[P_1 \mapsto D_1, \ldots, P_m \mapsto D_m], \rho \models F$. Here $I[P_1 \mapsto D_1, \ldots, P_m \mapsto D_m]$ is the interpretation $I'$ defined as $I$, except that $I'_i = D_i$ for each $i$, $1 \leq i \leq m$. The list $D_1, \ldots, D_m$ is called a model of $F$ in $I$.

We shall be especially interested in equational structures, i.e., structures that have only one predicate symbol $=$ apart from the existentially quantified predicates, of arity 2, and such that $I_=$ is mere equality; and which have no function symbol.

1. A linear ordering on $D$ is a binary relation $<$ that is irreflexive, transitive, and total, i.e., for all $x, y \in D$, $x < y$ or $y < x$ or $y = x$. Let $P$ be a binary predicate variable (henceforth $=$). Write a first-order formula $\text{Lin}(P)$ such that, for any equational structure $I$, $I \models \text{Lin}(P)$ iff $I_P$ is a linear ordering on the domain $D$ of $I$.

2. Let $k \in \mathbb{N}$, and $I$ be a finite structure in its standard representation, say with domain $D = \{1, \ldots, n\}$. One can interpret $k$-tuples in $D^k$ as numbers in base $n$: the tuple $(a_1, a_2, \ldots, a_k)$ is interpreted as $\sum_{i=1}^{k} a_i n^{k-1}$. If $\vec{x}$ and $\vec{y}$ are $k$-tuples of variables, write a formula $<^k$ stating that the number denoted by $\vec{x}$ is strictly less than the number denoted by $\vec{y}$.

3. Similarly, write a formula stating that the number denoted by $\vec{y}$ is one plus that denoted by $\vec{x}$ (implying in particular that $\vec{x}$ is strictly less than $n^k$: this is no computation mod $n^k$).

4. We wish to encode a tableau for the computation of an $n^k$-bounded non-deterministic Turing machine $M$ (where $n$ is the size of the input). Times, between 0 and $n^k - 1$, will be encoded as $k$-tuples of values in $D = \{1, \ldots, n\}$. We create the following predicate variables:
   - $S_q$, for each control state $q$; $S_q(\vec{t})$ should hold exactly when $M$ is at control state $q$ at time (denoted by) $\vec{t}$;
   - $T_a$, for each tape letter $a \in \Sigma$; $T_a(\vec{t}, \vec{x})$ should hold exactly when the symbol at position (denoted by) $\vec{x}$ at time (denoted by) $\vec{t}$ is $a$;
   - $H$: $H(\vec{t}, \vec{x})$ should hold exactly when the head is at position $\vec{x}$ at time $\vec{t}$.

Using these predicate variables (plus the symbols $=$ and $P$, and unary predicate variables $X_a$ for each letter $a \in \Sigma$, and maybe some other predicate variables, but no function symbol), show that given a fixed $M$, one can define an existential second-order formula $F_2$, with no free first-order variable, and only $=$, $P$ and $X_a$, $a \in \Sigma$, as free predicate variables, such that for every input $x$ of size $n$, $M$ accepts $x$ if and only if $I_n[(X_a \mapsto D_a)_{a \in \Sigma}, P \mapsto D_\leq] \models F_2$, where $D_a$ is the set of positions in $x$ where one finds the letter $a$ (i.e., $X_a, a \in \Sigma$, encode the input $x$), and $I_n[(X_a \mapsto D_a)_{a \in \Sigma}, P \mapsto D_\leq]$ is the unique equational structure in standard representation with domain of cardinality $n$ that maps $X_a$ to $D_a$ for each $a \in \Sigma$ and $P$ to the strict ordering $1 < 2 < \ldots < n$. 

3
5. Fagin’s Theorem states that the languages in $\mathbf{NP}$ are exactly the languages definable over equational structures by existential second-order formulae. Using the previous question, say what the latter means precisely.

Note moreover that we can restrict to existential second-order formulae without function symbol, and whose sole non-variable predicate symbols are $=,$ interpreted as equality, $P,$ interpreted as some fixed linear order, and $X_a,$ $a \in \Sigma,$ used to encode the input $x.$

6. A special structure $I$ (in its standard representation, with domain $D = \{1, 2, \ldots, n\}$) is one that has no function symbol, and only six predicate symbols apart from existentially quantified predicates: $=,$ of arity 2, where $I_=$ is equality; $\neq,$ of arity 2, where $I_{\neq}$ is non-equality; $\text{succ},$ of arity 2, where $I_{\text{succ}}(i, j)$ holds iff $j = i + 1$ (implying $i < n$), $\overline{\text{succ}},$ of arity 2, where $I_{\overline{\text{succ}}}(i, j)$ holds iff $j \neq i + 1;$ $Z,$ of arity 1, where $I_Z(i)$ holds iff $i = 0;$ and $L,$ of arity 1, where $I_L(i)$ holds iff $i = n$ ($L$ stands for “last”).

An existential second-order Horn formula is any existential second-order formula $\exists P_1 : P_k : n_k \cdot F$ with no function symbol, and where the only predicates in $F$ are $P_1, \ldots, P_m, =, \neq, \overline{\text{succ}}, Z$ and $L$, and where $F$ is a conjunction of Horn clauses (implicitly, universally quantified over all first-order variables).

Show that the languages in $\mathbf{P}$ are exactly the languages definable over special structures by existential second-order Horn formulae.

You will need the following piece of theory about Horn formulae, which you will admit. First, if $I \models \exists P_1 : n_1, \ldots, P_m : n_m \cdot F,$ i.e., if $F$ has a model in $I$ (i.e., $D_1, \ldots, D_m$ such that $I[P_1 \leftrightarrow D_1, \ldots, P_m \leftrightarrow D_m], \rho \models F$), then $F$ has a least model, i.e., a model $D^0_1, \ldots, D^0_m$ in $I$ such that for every other model $D_1, \ldots, D_m$ in $I,$ $D^0_1 \subseteq D_1, \ldots, D^0_m \subseteq D_m.$ Let us write Horn clauses as $H \leftrightarrow A_1, \ldots, A_p,$ where $A_1, \ldots, A_p$ are atomic formulae and $H$ is either an atomic formula or the special symbol $\bot,$ denoting false. (So $A \equiv A_1, \ldots, A_p$ is the clause $A \lor \neg A_1 \lor \ldots \lor \neg A_p,$ and $\bot \equiv A_1, \ldots, A_p$ is the clause $\neg A_1 \lor \ldots \lor \neg A_p.$) Call a fact any statement of the form $P_i(v_1, \ldots, v_n),$ where $1 \leq i \leq m,$ and $v_1, \ldots, v_n \in D,$ or $\bot.$ Any conjunction $F$ of Horn clauses defines a deduction system, which deduces some facts, and whose rules can be read as:

$$\begin{array}{cccc}
A_1 & A_2 & \ldots & A_p \\
\hline
H
\end{array}$$

where $H \equiv A_1, A_2, \ldots, A_p$ is a Horn clause in $F.$ Formally, the above rules mean that if one can find a substitution $\rho$ from variables to values such that $A_1\rho, \ldots, A_p\rho$ are facts that occur as conclusions to a derivation, then one can extend the derivation to one of $H\rho.$ ($A\rho$ is defined as the fact obtained by replacing each variable $x$ in $A$ by the corresponding value $\rho(x).$ Recall that there is no function symbol in $A.$ $\bot\rho$ is $\bot.$) A proof in this deduction system (called an $F$-proof) is a finite tree, defined in the usual way.

The main result you will need is that $I \models \exists P_1 : n_1, \ldots, P_m : n_m \cdot F$ iff there is no $F$-proof of $\bot;$ in this case, $F$ has a least model $D^0_1, \ldots, D^0_m,$ and this is characterized by the fact that $(v_1, \ldots, v_n) \in D^0_i$ iff $P_i(v_1, \ldots, v_n)$ has an $F$-proof.
For example, $I \models \exists P : 2 \cdot F$, where $F$ is the conjunction of the (implicitly, universally quantified) Horn clauses $P(x, y) \iff \text{succ}(x, y)$ and $P(x, y) \iff P(x, z), P(z, y)$. The facts that have an $F$-proof are then exactly those of the form $P(v, v')$ with $v < v'$ in $\{1, \ldots, n\}$. (End of example.)

III. Finite Models of Existential Second-Order Logic.

Here we consider again existential second-order formulae with function symbols, and arbitrary sets of predicates.

1. Given a fixed existential second-order formula $F_2$, the problem FIN-MC-$\exists_2(F_2)$ is the following problem:
   INPUT: a finite structure $I$, in its standard representation.
   QUESTION: $I \models F_2$?
   Why is FIN-MC-$\exists_2(F_2)$ in NP?

2. Show that the problem FIN-MC-$\exists_2$ is $\text{NEXPTIME}$-complete, where $\text{NEXPTIME}$ is the class of all languages that are decidable in so-called exponential time, i.e., bounded by $2^{\text{poly}(n)}$, on a non-deterministic Turing machine. FIN-MC-$\exists_2$ is the following problem:
   INPUT: a finite structure $I$, in its standard representation; an existential second-order sentence $F_2$.
   QUESTION: $I \models F_2$?
   For the sake of simplicity, we only ask for a polynomial-time reduction, not a logspace reduction. However, you must show that the result already holds if we restrict the domain $D$ to be exactly the domain of Booleans $\{0, 1\}$.