Introduction to Complexity Theory – CS-28100
Midterm Project – April 28, 2006
Instructor: Ketan Mulmuley Ry-165B

MIDTERM. **No collaboration is allowed on this assignment.**

Please **print your name on each sheet.** Please try to make your solutions readable.

This assignment is due on **Friday, May 5** at the **beginning of the class.**

In this project, we will prove Gödel's Incompleteness Theorem.

Definition 0.1 *A* first-order language *consists of a domain (for example, the non-negative integers), a set of operations (for example, +, *), a set of predicates (for example, =, <), a set of constants chosen from the domain, and a set of axioms defining the meaning of the operators and the predicates. For each theory we can define the language of true expressions over the constants, operators, predicates, variables, the logical connectives ($\wedge, \vee, \neg$), and the quantifiers ($\exists, \forall$).*

Definition 0.2 $(\mathbb{N}, +, *, =, <, 0, 1)$*, where* $\mathbb{N}$ *is the nonnegative integers, is known as number theory. In addition to the symbols given, we are allowed to refer to logical connectives (and,or,not) and quantifiers (exists, for all). Note that we are not allowed to use any symbols or operations other than those listed, such as indexing or modular arithmetic.*

Theorem 0.3 (Gödel's Incompleteness Theorem) *The language of true expressions in number theory is not decidable.*

1 **(Encoding IDs as sequences of integers)**

We encode our instantaneous description as a sequence of integers. An instantaneous description has letters from the tape alphabet, which we can assume to be $\{0, 1\}$, and the state. It is easiest to write the instantaneous description as $q, h, m, t_0, \ldots, t_m$, where q is an integer representing the state, h is an integer giving the tape position, and $t_0, \ldots, t_m$ are the contents of the non-blank tape squares.

Write a predicate that is true iff the ID describes a machine starting with a blank tape. Write a predicate that is true iff the ID describes a machine in an accepting state.

These predicates will NOT be in number theory - you can use operations not allowed by Definition 0.2. (In particular, you may use indexing.)

2 **(Encoding computations as sequences of sequences of integers)**

A computation is a sequence of instantaneous descriptions, $I_1, \ldots, I_n$, each of which follows the previous one according to the rules of the

Turing machine. We have a sequence of 0's and 1's representing each instantaenous description, a_{ij}. Write down a predicate that is true if and only if each step in the computation represented by a sequence of sequence of integers proceeds correctly. A correct computation is one in which the contents of the tape is unchanged except for the square under the head, and in which the change of symbol under the head and change of state proceed according to the rulebook of the Turing machine.

Again, this predicate will NOT be in number theory. (You may use indexing.) It should be of constant size (that is, not depending on the number of steps in the computation or on how long the tape gets during the computation). In particular, you cannot do things like say "the first ID moves to the second and the second moves to the third and the third mvoes to the fourth...". (You can, however, do something similar using quantifiers.)

Its size IS allowed to depend on the number of states used by the Turing machine, since we regard this as a constant.

3 **(Encoding sequences of integers as integers)**

In this part we will stick sequences of integers together into one large integer in such a way that we can recover the entire sequence using arithmetic.

Let $m = max\{n, x_0, x_1, \ldots, x_n\}$ Prove that the set of $u_i = 1 + (i + 1)m!, 0 \leq i \leq n$, are pairwise relatively prime and that $u_i > x_i$. This implies that there exists an integer $b < u_0 u_1 \cdots u_n$ such that $b \equiv x_i(\mod u_i), 0 \leq i \leq n$. We will use b as our encoding for the sequence $x_0, x_1, \ldots, x_n$, and we will denote it by $S_m(x_0, x_1, \ldots, x_n)$

4 **(Encoding sequences as predicates)**

Express Gödel's β function

$$\beta(b, c, i) = b \mod [1 + (i + 1)c]$$

as a predicate in number theory. That is, write a statement about integers b, c, i, k (using only the primitive symbols given in Definition 0.2) that is true if and only if $\beta(b, c, i) = k$. The only problem with part 3 is that it uses the mod function. Note that $\beta(S_m(x_0, \ldots, x_n), m!, i) = x_i$, if we have $m \geq \max\{n, x_0, x_1, \ldots, x_n\}$.

5 **(Encoding a computation as a predicate)** Argue that can you rewrite your predicate from part 2 so that it refers only to the integer representing a computation. Use the encoding given by part 3 and part 4. You should assume that no instantaneous description is longer than some constant m. (Which is true for some sufficiently large m.)

Now add your predicates from part 1, also rewritten in the language of part 3 and part 4, so that we know that the computation starts in the initial state with blank tape, and ends in the accepting state.

You don't have to write all this out, but you should argue that you will end up with a predicate in number theory, i.e., a predicate composed entirely of the symbols from Definition 0.2. This predicate should be of constant length, not depending on the number of steps in the computation or on m, the length of the longest ID.

6 **(Encoding computations as predicates)** If a TM M accepts when started on blank tape, it does so by a computation in which no ID is longer than some constant m. From step 5, we have a predicate $E_m(i)$, which is true if and only if i is the integer representing a computation leading to acceptance of ϵ with no ID longer than m.

The statement that M accepts ϵ, which is known to be undecidable, can be expressed as $\exists i \exists m(E_m(i))$, where E_m is a predicate that is true if and only if i is the binary encoding of a computation leading to acceptance of ϵ with no ID longer than m. Argue that number theory is undecidable.