Harvey Mudd College
CS 81 Final Exam
Fall semester, 2006

Part 1 Closed Book

1. This exam is in two parts.
2. For part 1, no reference materials are to be used. Part 1 counts for 25%.
3. You should probably not work on part 1 more than 45 minutes.
4. After submitting your solutions for part 1, you may begin part 2, wherein you may consult a single 2-sided crib sheet. Part 2 counts for 75%.
5. Once you start part 2, it is not permitted to return to part 1.
6. The exam has a 3-hour overall time limit for both parts combined.
7. It is best to spend your time working on problems that pay off in terms of the most points.
1. [5 points]

Consider the following families of languages that we have studied:

- regular
- context-free
- recursive
- recursively-enumerable

In what important way is the family “recursive” different from all of the others?
2. [10 points]

For each language row below, check the left-most column entry for which the language is in that family. Use the row numbers should you wish to attach a note clarifying your answer to any item. (For the rows involving formulas, assume that the variables and proposition symbols in the formula are represented using a finite alphabet, e.g. proposition symbols are $p, p', p'', p'''$, … so that a finite alphabet can be used overall.)

<table>
<thead>
<tr>
<th></th>
<th>Regular</th>
<th>Context-Free</th>
<th>Recursive</th>
<th>Recursively-Enumerable</th>
<th>None of the preceding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The set of all regular expressions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The set of all Turing machine encodings, using an encoding such as the one in our text.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The set of all Turing machine encodings of Turing machines that always halt when started with their own description as input.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>The set of all surnames in the most recent Los Angeles telephone directory.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>The set of all predicate logic formulas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>The set of all universally-valid predicate logic formulas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>The set of all satisfiable predicate logic formulas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>The set of all propositional logic formulas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>The set of all satisfiable propositional logic formulas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>The empty set.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. [5 points]

I.M. Brilliant has devised a new Turing-machine encoding that he thinks will “help solve a variant of the halting problem.” Instead of using $M_0, M_1, M_2, \ldots$ as the enumeration of all Turing machines using the encoding discussed in the text, I.M. suggests encoding the Turing machines as $N_0, N_1, N_2, \ldots$ in the following way:

a. As sets, the two sets of machines are equal \( \{M_0, M_1, M_2, \ldots\} = \{N_0, N_1, N_2, \ldots\} \), so both cover all of the machines.

b. The even-numbered $N_0, N_2, N_4, \ldots$ correspond to machines that halt on every input.

c. The odd-numbered machines in $N_1, N_3, N_5, \ldots$ correspond to machines that fail to halt on at least one input.

Comment on I.M.’s approach, specifically on its general utility and a possible algorithm for transforming the new numbering into the old numbering.
4. [5 points]

Goldbach’s conjecture is that every even integer > 2 is the sum of two primes (not necessarily distinct). Consider the language over \{0, 1\}:

\[ L = \{ x \mid x = 1 \text{ if Goldbach’s conjecture is true, 0 otherwise} \} \]

Show that L is recursive.
Harvey Mudd College
CS 81 Final Exam
Fall semester, 2006

Part 2

1. For this part, you may use a single 2-sided crib sheet, but no other reference material.
2. Please turn in your crib sheet along with the exam. It will be returned to you.
3. Part 2 counts for 75%.
4. You are not permitted to return to part 1 after starting part 2.
5. The exam has a 3-hour overall time limit for both parts combined.
6. It is best to spend your time working on problems that pay off in terms of the most points.
5. [10 points]

The sequent below states that the composition of two 1-1 functions is itself 1-1. Prove it using natural deduction.

\[ \forall x \forall y \ f(x) = f(y) \rightarrow x = y, \quad \text{(premise)} \]

\[ \forall x \forall y \ g(x) = g(y) \rightarrow x = y \quad \text{(premise)} \]

\[ \vdash \forall x \forall y \ f(g(x)) = f(g(y)) \rightarrow x = y \quad \text{(conclusion)} \]
6. [15 points]

Prove using natural deduction:

\[ \vdash \exists x (\varphi \land \psi) \rightarrow ((\exists x \varphi) \land (\exists x \psi)) \]
7. [15 points]

Using the tree method, establish whether or not the following formula is valid.

\[(\exists x \ (P(x) \lor Q(x))) \leftrightarrow ((\exists x \ P(x)) \lor (\exists x \ Q(x)))\]
8. [20 points]

“Polish” notation is a notation for terms that doesn’t require parentheses. It relies on each function symbol having a distinct arity. For simplicity, we consider here a term language with just two symbols, a 2-ary function symbol, \( f \), and a constant symbol, \( a \). In this case, the terms in Polish notation are given by:

a. The constant symbol \( a \) is a term.
b. If \( X \) and \( Y \) are terms, then so is \( fXY \).
c. The only terms are those derived by the above rules.

For example, here are five terms: \( a, faa, fafaa, ffaafaa, ffaafafaa \).

Let \( L \) be the language of all terms as above.

a. [2/20] Give a context-free grammar for \( L \).
b. [8/20] Diagram the abstract states of \( L \) and the transitions between them.
c. [4/20] Show that \( L \) is not regular.
d. [6/20] Give the transition rules for a pushdown automaton that accepts \( L \).
9. [15 points]

The sequent below states that the composition of two onto functions is itself onto.

Prove it by the resolution method.

\[ \forall y \exists x \quad f(x) = y \]  
\[ (\text{premise}) \]

\[ \forall y \exists x \quad g(x) = y \]  
\[ (\text{premise}) \]

\[ \vdash \forall y \exists x \quad f(g(x)) = y \]  
\[ (\text{conclusion}) \]

Here is an equivalent clause form, to help you get started, where clauses 4-6 represent facts about equality that will likely be needed.

1. \( E(f(i(y)), y) \).
2. \( E(g(j(y)), y) \).
3. \( \neg E(f(g(x)), c) \).
4. \( \neg E(x, y) \lor \neg E(y, z) \lor E(x, z) \).
5. \( \neg E(x, y) \lor E(f(x), f(y)) \).
6. \( \neg E(x, y) \lor E(g(x), g(y)) \).

Here \( E \) represents the equality predicate, \( i \) is a Skolem function representing the inverse of \( f \), \( j \) is a Skolem function representing the inverse of \( g \), and \( c \) is a Skolem constant that arises from negating the conclusion. (There is more that can be said about \( E \), but this is enough to enable the proof.)