Midterm Exam

This is a preview of the published version of the quiz

Started: Apr 19 at 11:20pm

Quiz Instructions

This is the Midterm exam and it is 100% on-line. As I indicated previously, you are not to discuss this exam with anyone until after the exam time is completed and, even then, only with students who also took the exam until grades are posted. You may use no resources that are not from this course. That means you can use your personal notes, all the material linked from the course website, and anything at this Webcourses site. You cannot go hunting on the Internet for answers. You cannot communicate with anyone while taking the exam. You cannot share the contents of this exam. You cannot make copies of the exam using screen capture or any other method. You cannot post questions or answers from the exam on any forum. Before taking the exam you must agree to all the conditions set in the Honor Code quiz associated with this course.

Note that Canvas loves to number all questions sequentially so my 11.1, 11.2, 12.1, 12.2, and 12.3 get numbered 11-15. That's annoying but not a big deal and I keep consistent numbering in the questions themselves.

All the best!!

Question 1 — 10 pts

1. This is a collection of 10 True/False Questions.

Membership in Phrase-Structured Languages is semi-decidable	TRUE
Membership in Context-Free Languages can be solved in polynomial time	TRUE
Every recursive (decidable) set is recognized by some primitive recursive function	FALSE
Every non-empty re set is enumerated by some primitive recursive function	TRUE
An algorithm exists to determine if a Context-Sensitive Language is finite	FALSE
The Context-Free Languages are closed under intersection	FALSE
We can decide if a Context-Free Grammar generates the empty set	TRUE
If S is a decidable set and R is a subset of S, then R is also decidable	FALSE

The quotient of a Context-Free Language and a Regular Language is Context-Free	TRUE
Deterministic PDAs have the same power as Non-Deterministic PDAs	FALSE

Question 2

8 pts

2. Upper Bounds on Complexity

We learned that quantification can be used to determine the upper bound of the complexity of some problem. The general form of such expressions is **Q [Algorithmic Predicate }**. Here **Q** is a sequence of alternating quantifiers. Usually, the predicate is Primitive Recursive involving the functions **STP** and **VALUE**. I will give you a set (membership in it is the problem) and an associated predicate. The Quantifier part won't be specified, and you will need to choose from a set of options, one of which specifies what that quantifier is.

S = { f | domain(f) is non-empty }.

f ∈ S iff ∃ <x,t> **[STP(f,x,t)]**

Let T = { <f,x> | 2x is in range(f) }.

<f,x> ∈ T iff ∃ <y,t> **[STP(f,y,t) & VALUE(f,y,t)=2x]**

Let U = { f | Range(f) contains no odd numbers }.

f ∈ U iff ∀ <x,t> **[STP(f,x,t) ⇒ EVEN(VALUE(f,x,t))]**

Let V= { f | Domain(f) is not the set of Natural Numbers }.

f ∈ V iff ∃ x ∀ t **[~STP(f,x,t)]**

Question 3

5 pts

3. Infinite Recursively Enumerable Sets

Let set **S** be a **recursive enumerable**, possibly **recursive (decidable)**, **non-empty** set. Further, let **S** be the range of some algorithm $\mathbf{f_S}$. Define $\mathbf{f_R}$ iteratively as follows:

$\mathbf{f_R(0) = f_S(0)}$
$\mathbf{f_R(y+1) = f_S(\ \mu z\ [\ f_S(z) > f_R(y)\ ]}$

Clearly, $\mathbf{f_R}$ enumerates a set **R** that is in some way related to **S**. That relation and various properties of **R** and **S** are the topics of the following question where you must select all true properties and not select the false ones.

- [x] If **S** is infinite, $\mathbf{f_R}$ is a monotonically increasing function
- [x] **S** is finite if and only if **R** is finite
- [ ] $\mathbf{f_R}$ halts for all input (is an algorithm)
- [x] **R** is a decidable (recursive) set
- [ ] $\mathbf{f_R}$ is a primitive recursive function

Question 4

5 pts

4. Rice's Theorem (Variations)

Match the problems about sets of function indices with the statements about the applicability of Rice's Theorems (including its three variants and circumstances under which no version is applicable). Note, your answer must always choose a Weak Form if one is applicable.

{ f | for every x, f(x) < x }	Strong Version of Rice's Thec
{ f | for every x, f(2x) converges and f(2x+1) diverges }	Domain (Weak) Version of Ric
{ f | for some x, f(x) converges in at most x steps }	Rice's Theorem does not app
{ f | for every x, there is some y such that f(y) = x }	Range (Weak) Version of Rice
{ f | there is some g not equal to f where, for some x, f(x) = g(x) or both diverge }	Rice's Theorem does not app

Question 5

5 pts

5. Inherent and Temporal Properties of Problems

Each of the following is either **solved**, **solvable** but unsolved, **RE** (semi-decidable) but not solvable, **Co-RE** but not solvable, or **non-RE no-Co-RE** (not semi-decidable or even complement of semi-decidable). Specify the correct, precise status of each of the following:

Is P = NP?	Solvable

Is f(x) = 2x, for some arbitrary procedure f and input x?	RE
Is f an algorithm, for arbitrary procedure f?	Non-RE/Non-Co-RE
What is the minimum number of states for some DFA?	Solved
Is domain(f) empty, for arbitrary procedure f?	Co-RE

Question 6

5 pts

6. Regular Expressions

The unique solution to the regular equation A = 0 + A1 is [01*].

The unique solution to the regular equation A = 0 + A1* is [Non-existent].

The k in $R_{i,j}^k$ is the [Max index of any intermediat].

The i in $R_{i,j}^k$ is the [Starting State].

The values i and j in $R_{i,j}^k$ [can] be equal

Question 7

4 pts

7. Pumping Lemmas

There is a Pumping Lemma for Regular Languages and a separate one for Context-Free Languages.

The Pumping Lemma for Context-Free Languages degenerates to the Pumping Lemma for Regular Languages when [The alphabet has just one lett].

The Pumping Lemma for Context-Free Languages assumes the grammar is in [Chomsky Normal] Form.

Myhill-Nerode provides an alternative to the Pumping Lemma for [Regular] Languages

Both Pumping Lemmas depend on the [Pigeon Hole] Principle

Question 8

5 pts

8. Meta-Theorem on Closures of Regular and Context-Free Languages
What language is gotten from each of these uses of the meta-theorem we developed from knowing the Regular and Context-Free Languages are both closed under Substitution, Homomorphism, and Intersection with Regular Languages? In each case, for **a** $\in \Sigma$, we define **g(a) = a'**, f**(a) = {a,a'**}, and **h(a) = a**, **h(a') =** λ . Here **a'** is a new symbol uniquely associated with **a** but not appearing in Σ. We then extend each of these in a natural way to apply to strings and do closure constructions for some operation using the form:

h(f(L) $\cap$ **E)**, where **E** is some **Regular Expression** and **L** is some **non-empty Context-Free Language**

What is the language we get when **E** is Σ* **g(**Σ***)?** [Prefix(L)]

What is the language we get when **E** is **g(**Σ***)**? [String of length 0]

What is the language we get when **E** is **g(**Σ***)** Σ* **g(**Σ***)**? [Substring(L)]

What is the language we get when **E** is **g(**Σ***)** Σ*? [Suffix(L)]

What is the language we get when **E** is Σ*? [L]

Question 9

5 pts

9.
PCP was used to show the undecidability of which of the following?

- [x] Is the language derived by some CSG non-empty?
- [ ] Does some CFG produce all strings over its alphabet?
- [x] Is the intersection of two CFLs non-empty?
- [x] Is an arbitrary CFG ambiguous?
- [ ] Is the intersection of two CFLs a non-CFL?

Question 10 3 pts

10.

The traces of terminal computations produced by running Turing Machines are [Context-Sensitive] but the complements of these traces are [Context-Free]. The reason is that the easier case is a [there exists] condition.

11. Closure Properties of Sets over the Natural Numbers.
Let set **A** be a **non-empty recursive** (**decidable**) set, let **B** be an **RE non-recursive** set, and let **C** be a **non-RE** set. Specify, for each set below, which can be **recursive**, **RE**, and/or **non-RE**. By can be, I mean there are some choices of the sets **A**, **B**, and **C**, used to construct the new set, that can make it **recursive**, **RE**, and/or **non-RE**. For each of the following two questions, specify what are the possible results of the given operation.

Question 11 3 pts

11.1
Let $\mathbf{D = A \oplus B = \{ x \mid x \in (A \cup B)}$ but $\mathbf{x \notin (A \cap B) \}}$. **D** can be

- [x] RE
- [ ] Recursive
- [x] Non-RE

Question 12 3 pts

11.2
Let $\mathbf{E = \{ z \mid z = y//x}$ where $\mathbf{x \in A, y \in B\}}$. Recall that **//** is just integer division with the odd exception that any value divided by **0** is **0**. **E** can be

- [x] RE

- [ ] Non-RE
- [x] Recursive

12. Closure Properties of Languages over some alphabet Σ .

Consider **non-empty** languages (sets) **A**, **B**, and **C**, where **A** is **Regular**, **B** is **Context-Free**, and **C** is **Context-Sensitive**. Specify, for each set below, which can be **Regular**, **Context-Free**, and/or **Context Sensitive**. By can be, I mean there are some choices of the sets **A**, **B**, and **C** used to construct the new set that can make it **Regular**, **Context-Free**, and/or **Context-Sensitive.** For each of the following four questions, specify what are the possible results (more than one might be true) of the given operation.

Question 13 — 3 pts

12.1

Let **D = A − B**, that is all elements in **A** but not in **B**, then **D** can be

- [x] Regular
- [x] Context-Sensitive
- [x] Context-Free

Question 14 — 3 pts

12.2

Let **E = A ∩ B**, that is all elements in both **A** and **B**, then **E** can be

- [ ] Context Sensitive
- [x] Regular
- [x] Context Free

Question 15 — 3 pts

12.3

Let **F =** Σ^* **−** **C**, that is all elements not in **C** can be

- [x] Context Free
- [ ] Regular
- [x] Context Sensitive

Quiz saved at 11:27pm

Submit Quiz