Comprehensive Exam: Algorithms and Concrete Mathematics Autumn 2004

This is a one hour closed-book exam and the point total for all questions is 60.

In questions that ask you to provide an algorithm, please explain the algorithm in words and diagrams, no need to write code or pseudo code. Also, for any algorithm, state and prove its running time. No credit will be given for exponential-time algorithms. Polynomial but slow algorithms will get some partial credit. Amount of credit will depend on how much slower they are compared to what is achievable using the knowledge in the reading list.

For full credit, the answers should be short and precise. Long and convoluted answers will not get full credit even if they are correct.

1. **[14 pts]** Please answer "true" or "false" to each one of the following questions. Correct answers will give you **(2 pts)** each while wrong answers will reduce your score by **(2 pts)** each.

   - $2n = O(n^2)$
   - $n! = o(n^n)$
   - $(\log n)^{10} = \Omega(\sqrt[10]{n})$
   - Any graph with n nodes and m edges has a spanning tree with $n-1$ edges.
   - Assume that T is a minimum cost spanning tree for a weighted graph $G = (V, E)$ with positive weights $w : E \to R^+$. Will T be still a minimum cost spanning tree if instead of w we use $\log w$?
   - You are given a graph $G = (V, E)$ with positive lengths on edges and a shortest path P from $v \in V$ to $u \in V$. Next, lengths are transformed by uniformly adding 1 to the length of each edge. Will P be still a shortest path from v to u ?
   - Let A be an array of n distinct numbers. The goal is to rearrange the numbers so that $\forall i$ where $0 < i < n/2$ we have $A[i] < A[2i]$. It is true that this operation requires $\Omega(n \log n)$ time in the comparison model?.

2. **[15 pts]** Let u and v be two n-bit numbers, where for simplicity n is a power of 2. The traditional multiplication algorithm requires $\Theta(n^2)$ operations. A divide-and conquer based algorithm splits the numbers into two equal parts, computing the product as

   $$uv = (a2^{n/2} + b)(c2^{n/2} + d) = ac2^n + (ad + bc)2^{n/2} + bd$$

   Here, a and c are the higher order bits, and b and d are the lower order bits of u and v respectively. Multiplications are done recursively, except multiplication by 2^i for integer i, which is a shift and we assume takes $\Theta(i)$ time. We also assume that addition/subtraction of i-bit numbers takes $\Theta(i)$.

- **[6 pts]** Write a recurrence for the running time of this algorithm as stated. Solve the recurrence and determine the running time.
- **[9 pts]** Note that $ad + bc$ can be computed as $(a + b)(c + d) - ac - bd$. Why is this advantageous? Write and solve a recurrence for the running time of modified algorithm.

3. **[16 pts]** Given a set of items $x_1, x_2, \ldots, x_n$ where item x_i has a non-negative weight $w(x_i)$, the goal is to find a subset of items with maximum total weight under the constraint that if item x_i is chosen, then you are forbidden to chose either x_{i+1} or x_{i-1} (forbidden to choose the "neighbors" of x_i). For example, you can choose x_5, x_8, and x_{10}, but you are not allowed to choose x_5, x_6, x_9, x_{11}. Design an efficient algorithm to solve this problem, prove correctness and analyze running time. For simplicity, your algorithm should just produce the value of the maximum total weight and does not need to list the items that you are choosing to achieve this weight.

4. **[15 pts]** Consider a section of highway with numbered exits. You are given information about n cars that you suspect of speeding. Car i will enter the highway at point s_i and exits at $t_i > s_i$. The goal is to place minimum number of photo-radar units along the highway to ensure that you will catch (photograph) all of the n speeding cars. (Assume that once the photo-radar is placed, you cannot move it.)

   We say that paths of two cars i and j do not intersect (disjoint) if $t_i < s_j$ or $t_j < s_i$. Observe that if there is a set of k of cars that have disjoint paths, then you will need at least k photo-radars.

   Let k^* be the size of the maximum-size set of cars with disjoint paths. The observation above implies that you will need at least k^* photo-radars. Prove that in fact k^* photo-radars is sufficient, i.e. there exists a placement of k^* photo-radars that will photograph all of the cars.

Comprehensive Exam: Algorithms and Concrete Mathematics
Autumn 2004

This is a one hour closed-book exam and the point total for all questions is 60.

In questions that ask you to provide an algorithm, please explain the algorithm in words and diagrams, no need to write code or pseudo code. Also, for any algorithm, state and prove its running time. No credit will be given for exponential-time algorithms. Polynomial but slow algorithms will get some partial credit. Amount of credit will depend on how much slower they are compared to what is achievable using the knowledge in the reading list.

For full credit, the answers should be short and precise. Long and convoluted answers will not get full credit even if they are correct.

1. **[14 pts]** Please answer "true" or "false" to each one of the following questions. Correct answers will give you **(2 pts)** each while wrong answers will reduce your score by **(2 pts)** each.

a
- $2n = O(n^2)$

  Answer: TRUE

b
- $n! = o(n^n)$

  Answer: TRUE

c
- $(\log n)^{10} = \Omega(\sqrt[10]{n})$

  Answer: FALSE

d
- Any graph with n nodes and m edges has a spanning tree with $n-1$ edges.

  Answer: FALSE, graph can be disconnected

e
- Assume that T is a minimum cost spanning tree for a weighted graph $G=(V,E)$ with positive weights $w : E \to R^+$. Will T be still a minimum cost spanning tree if instead of w we use $\log w$?

  Answer: TRUE, relative order of weights does not change

f
- You are given a graph $G=(V,E)$ with positive lengths on edges and a shortest path P from $v \in V$ to $u \in V$. Next, lengths are transformed by uniformly adding 1 to the length of each edge. Will P be still a shortest path from v to u ?

  Answer: FALSE

g
- Let A be an array of n distinct numbers. The goal is to rearrange the numbers so that $\forall i$ where $0 < i < n/2$ we have $A[i] < A[2i]$. It is true that this operation requires $\Omega(n \log n)$ time in the comparison model?.

  Answer: FALSE

2. **[15 pts]** Let u and v be two n-bit numbers, where for simplicity n is a power of 2. The traditional multiplication algorithm requires $\Theta(n^2)$ operations. A divide-and conquer based algorithm splits the numbers into two equal parts, computing the product as

$$uv = (a2^{n/2} + b)(c2^{n/2} + d) = ac2^n + (ad + bc)2^{n/2} + bd$$

Here, a and c are the higher order bits, and b and d are the lower order bits of u and v respectively. Multiplications are done recursively, except multiplication by 2^i for integer i, which is a shift and we assume takes $\Theta(i)$ time. We also assume that addition/subtraction of i-bit numbers takes $\Theta(i)$.

- **[6 pts]** Write a recurrence for the running time of this algorithm as stated. Solve the recurrence and determine the running time.

  Answer: $T(n) = 4T(n/2) + \Theta(n), T(1) = 1$. This means that $T(n) = \Theta(n^2)$, no improvement over brute-force approach.

- **[9 pts]** Note that $ad + bc$ can be computed as $(a + b)(c + d) - ac - bd$. Why is this advantageous? Write and solve a recurrence for the running time of modified algorithm.

  Answer: There is one less multiplication per iteration, resulting in $T(n) = 3T(n/2) + \Theta(n), T(1) = 1$. This means that $T(n) = \Theta(n^{\log_2 3}) = o(n^2)$.

  Note that, strictly speaking, some of the multiplications are of $n+1$-bit numbers and not n-bit numbers. Our calculations are still correct since to multiply two $n+1$-bit numbers, one can execute a single multiplication of two n-bit numbers plus $\Theta(n)$ extra work.

3. **[16 pts]** Given a set of items $x_1, x_2, \ldots, x_n$ where item x_i has a non-negative weight $w(x_i)$, the goal is to find a subset of items with maximum total weight under the constraint that if item x_i is chosen, then you are forbidden to chose either x_{i+1} or x_{i-1} (forbidden to choose the "neighbors" of x_i). For example, you can choose x_5, x_8, and x_{10}, but you are not allowed to choose x_5, x_6, x_9, x_{11}. Design an efficient algorithm to solve this problem, prove correctness and analyze running time. For simplicity, your algorithm should just produce the value of the maximum total weight and does not need to list the items that you are choosing to achieve this weight.

Answer: Let W_i denote the maximum weight that we can collect by choosing only among items numbered 1 through i. Clearly, $W_1 = x_1$. To compute W_{i+1} we need to consider two cases: either x_{i+1} participates in the optimum solution when the problem is restricted to items 1 through $i+1$, or not. If it does not, then the optimum solution is the same as optimum for problem when input is restricted to items 1 through i, i.e. in this case $W_{i+1} = W_i$. Alternatively, if x_{i+1} participates, then x_i cannot participate. We claim that $W_{i+1} = W_{i-1} + w(x_{i+1})$ in this case. To prove this claim, assume that there is an optimum for the restricted problem (items 1 through $i+1$) that includes $i+1$ and that is that produces weight larger than $W_{i+1} = W_{i-1} + w(x_{i+1})$. Since x_{i+1} participates, x_i does not, and thus removing x_{i+1} from this solution we get a legal solution for the problem restricted to items 1 through $i-1$ with weight higher than W_{i-1}, which contradicts our assumption.

To summarize, $W_{i+1} = \max\{W_i, W_{i-1} + w(x_{i+1})\}$. This takes constant time per iteration. With n iterations, the total is $\Theta(n)$. Observe that one does not need an extra array of n elements to hold all the computed W_i values since at each iteration it is sufficient if we store only the latest two values, W_i and W_{i-1}.

4. **[15 pts]** Consider a section of highway with numbered exits. You are given information about n cars that you suspect of speeding. Car i will enter the highway at point s_i and exits at $t_i > s_i$. The goal is to place minimum number of photo-radar units along the highway to ensure that you will catch (photograph) all of the n speeding cars. (Assume that once the photo-radar is placed, you cannot move it.)

   We say that paths of two cars i and j do not intersect (disjoint) if $t_i < s_j$ or $t_j < s_i$. Observe that if there is a set of k of cars that have disjoint paths, then you will need at least k photo-radars.

   Let k^* be the size of the maximum-size set of cars with disjoint paths. The observation above implies that you will need at least k^* photo-radars. Prove that in fact k^* photo-radars is sufficient, i.e. there exists a placement of k^* photo-radars that will photograph all of the cars.

   Answer: Car i corresponds to a closed interval (s_i, t_i). We will prove the claim by giving an algorithm that finds k disjoint intervals and k places for photo-radars such that each interval has at least one photo-radar in it. By definition, $k^* \geq k$. Moreover, since k photo-radars are sufficient, $k^* \leq k$. Thus, $k^* = k$, proving the claim.

   The algorithm starts by ordering all intervals in increasing order by t_i. Pick the first interval in this order (s_{i_1}, t_{i_1}), mark it, and delete it from the set. Place photo-radar at t_{i_1} and delete all intervals that intersect with (s_{i_1}, t_{i_1}). Repeat until all intervals are deleted. Result is k marked intervals and k photo-radar placements.

   First we claim that, by construction, all k chosen intervals are disjoint. Next we claim that the placed photo-radars will photograph all cars, i.e. for every interval there is a photo-radar that is placed at one of the points in the interval. Observe that each iteration results in placement of a photo-radar and deletion of some intervals. We claim that all deleted intervals in an iteration are "touched" by the photo-radar placed in this iteration. This is trivially true for the interval marked in this iteration. Consider an interval (s_j, t_j) deleted during iteration q. By construction, $t_j > t_{i_q}$. The only way the photo-radar at t_i does not touch it is if $s_j > t_{i_q}$. But in this case we will not delete this interval.

   One can implement this algorithm to run efficiently, but this is irrelevant to the question, since all we needed was a proof of the claim.

2006 Comprehensive Examination
Artificial Intelligence

1. Propositional Constraint Satisfaction. (20 points) In this question, we consider representing propositional satisfiability (SAT) problems as CSPs.

(a) Consider the following SAT Problem.

$$(\neg X_1 \vee X_2) \wedge (\neg X_2 \vee X_3) \wedge \ldots \wedge (\neg X_{n-1} \vee X_n)$$

How many solutions are there for this SAT problem as a function of *n*?

(1) 0
(2) 1
(3) *n*
(4) *n*+1
(5) n^2
(6) 2^n

(b) Suppose we apply backtracking to find *all* solutions to a SAT problem of the type given in (a) using iterated constraint satisfaction. (To find all solutions to a CSP, we continue searching after each solution is found until all possibilities are tried.) Assume that the variables are ordered $X_1, \ldots, X_n$ and *false* is ordered before *true*. How much time will it take to terminate the search?

(1) constant time
(2) linear in *n*
(3) quadratic in *n*
(4) exponential in *n*

(c) True or False: Every Horn-form SAT problem can be solved in time linear in the number of variables.

(d) True or False: Every tree-structured binary CSP with discrete, finite domains can be solved in time linear in the number of variables.

2. Logic. (20 points) Let Γ and Δ be sets of closed sentences in first-order logic, and let φ and ψ be individual closed sentences in first-order logic. State whether each of the following statements is true or false. No explanation is necessary.

(a) If $\Gamma \cap \Delta \models \varphi$, then $\Gamma \models \varphi$ and $\Delta \models \varphi$.

(b) If $\Gamma \cup \Delta \models \varphi$, then $\Gamma \models \varphi$ or $\Delta \models \varphi$.

(c) $\Delta \models (\varphi \Rightarrow \psi)$ if and only if $\Delta \cup \{\varphi\} \models \psi$.

(d) $\Delta \models \varphi$ or $\Delta \models \psi$ if and only if $\Delta \models (\varphi \vee \psi)$.

(e) If $\Delta \models \varphi$ and $\Delta \models \neg\psi$, then $\Delta \not\models (\varphi \Rightarrow \psi)$.

(f) If $\Delta \models \varphi$ and $\Delta \models \psi$, then $\Delta \models (\varphi \Rightarrow \psi)$.

(g) If $\Delta \models p(\tau)$ for some ground term τ, then $\Delta \not\models \forall x.\neg p(x)$.

(h) If $\Delta \models p(\tau)$ for every ground term τ, then $\Delta \models \forall x.p(x)$.

(i) If $\Delta \models \forall x.(p(x) \Rightarrow q(x))$, then $\Delta \models \exists x.(p(x) \wedge q(x))$.

(j) If $\Gamma \models (\varphi \Rightarrow \psi)$ and $\Delta \models (\psi \Rightarrow \varphi)$,then $\Gamma \cap \Delta \models (\varphi \Rightarrow \psi) \vee (\psi \Rightarrow \varphi)$.

3. Resolution. (20 points) Use the resolution method and the following premises to prove the conclusion shown below.

Premises:

a. $\forall x.\forall y.(\, p(x,y\,) \Rightarrow \exists z.q(x,y,z))$

b. $\exists x.\forall y.\forall z.(r(y,z) \Leftrightarrow q(x,y,z))$

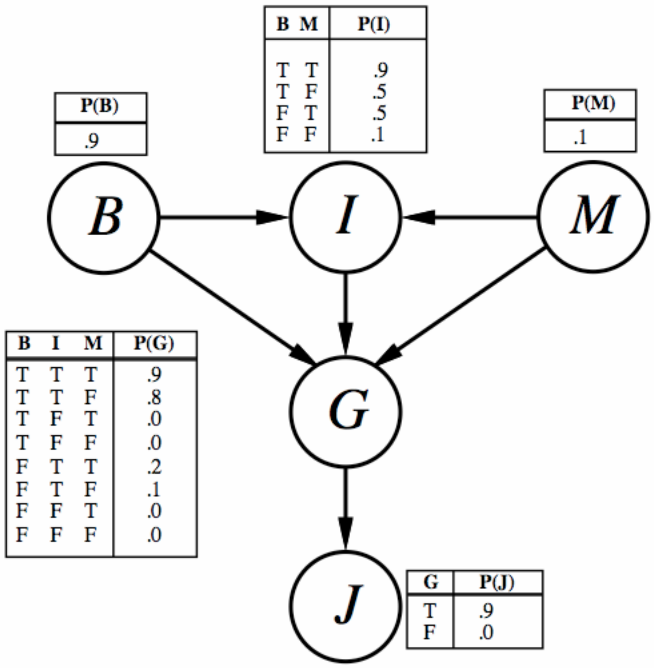
c. $\forall x.\exists y.(\neg p(x,y) \Rightarrow \forall z.q(x,y,z))$

Conclusion:

$\exists w.\exists x.\exists y.\exists z.(r(x,y) \wedge q(x,w,z))$

Note that this is a question about Resolution. You will get zero points (nil, nada, rien, zip, nothing) unless you prove it using the standard resolution procedure.

4. Bayes Nets. (20 points) Consider the Bayes net shown below. Here, B means that a person broke the election law, I means that the person was indicted, M means that the prosecutor in politically motivated, G means that the defendant is found guilty, and J means that the defendant is jailed.



(a) Which, if any, of the following are asserted by the network structure (ignoring the CPTs for now)?

(1) $p(\text{B, I, M}) = p(\text{B})\, p(\text{I})\, p(\text{M})$
(2) $p(\text{J} \mid \text{G}) = p(\text{J} \mid \text{G, I})$
(3) $p(\text{M} \mid \text{G, B, I}) = p(\text{M} \mid \text{G, B, I, J})$

(b) Calculate the value of $p(\text{B, I, -M, G, J})$

(c) Calculate the probability that someone goes to jail given that the person broke the law, has been indicted, and faces a politically motivated prosecutor.

5. Learning. (20 points) In this question, we consider decision trees with numerical input attributes A_1 and A_2 and a Boolean output attribute Y. In such trees, the test at each internal node is an inequality of the form $A_i > c$ where c may be any number (to be chosen by the learning algorithm). The value at each leaf node is *true* or *false*. In a *test-once* tree, each attribute may be tested at most once on any path in the tree. In a *test-many* tree, each attribute may be tested more than once.

Suppose we are given the following training set.

A_1	A_2	Y
3	3	*false*
6	13	*true*
15	14	*true*
14	22	*false*

(a) Draw a test-once tree that classifies the examples correctly.

(b) Write down the information gain of your root test. Your answer may contain unevaluated logs.

(c) True or False: Every non-noisy training set can be correctly classified by a test-once decision tree.

(d) True or False: Every non-noisy training set can be correctly classified by a test-many decision tree.

2006 Comprehensive Examination Solutions
Artificial Intelligence

1. Propositional Constraint Satisfaction. (20 points) Taken from a Final exam in CS188 at UCB.

(a) $n + 1$ solutions. Once any X_i is true, all subsequent X_js must be true. Hence, each solution consists of *I* falses followed by $n - i$ trues for $i = 0, \ldots, n$.

(b) Quadratic in *n*. Consider what part of the complete binary tree is explored during the search. The algorithm must follow all solutions sequences, which themselves cover a quadratic –sized portion of the tree. Failing branches are those trying a false after a preceding variable is assigned true. Such conflicts are detected immediately.

(c) True. Use the Forward Chaining Algorithm in Russell and Norvig.

(d) True. Directed arc consistency in Russell and Norvig.

2. Logic. (20 points) Adapted from a problem in CS157 at Stanford.

(a) True
(b) False
(c) True
(d) False
(e) False
(f) True
(g) False
(h) False
(i) False
(j) True

3. Resolution. (20 points) Adapted from a problem in CS157 at Stanford.

1. $\{\neg p(x, y), q(x, y, f(x,y))\}$	Premise a
2. $\{\neg r(y, z), q(a, y, z)\}$	Premise b
3. $\{r(y, z), \neg q(a, y, z)\}$	Premise b
4. $\{p(x, g(x)), q(x, g(x), z)\}$	Premise c
5. $\{\neg r(x,y), \neg q(x, w, z)\}$	Negated Goal
6. $\{\neg q(a, x, y), \neg q(x, w, z)\}$	3, 5
7. $\{q(x, g(x), f(x, g(x))), q(x, g(x), z)\}$	1, 4
8. $\{\neg q(g(a), w, z)\}$	6, 7 (factoring 7)
9. $\{\}$	7, 8 (factoring 7)

4. Bayes Nets. Taken from a Final exam in CS188 at UCB.

(a) Assertions (2) and (3) are implied by the structure of the net; assertion (1) is not.

(b) $p(b, i, \neg m, g, j) = p(b) * p(\neg m) * p(i \mid b, -m) * p(g \mid b, i, m) * p(j \mid g)$
$= 0.9 * 0.9 * 0.5 * 0.8 * 0.9 = 0.2916$

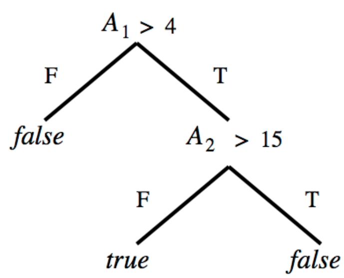
(c) Since B, I, M are true in the evidence, we can treat G as having a prior of 0.9 and look at the submodel with just G and J.

$$p(j \mid b, g, m) = p(j \mid g) * p(g) = 0.9 * 0.9 = 0.81$$

That is, the probability of going to jail is 0.81.

5. Learning. Taken from a Final exam in CS188 at UCB.

(a) One possibility follows.



(b) With 2 examples of each kind, the initial entropy is 1 bit. After the test, we have one subset with counts 0, 1 and one subset with counts 2, 1. Hence the information gain is as follows.

$$1- ((1/4 * 0) + (3/4) * (-1/3 \log(1/3) - 2/3 \log(2/3))$$
$$= 1+ (1/4) * \log(1/3) + (1/2) * \log(2/3)$$
$$\sim 0.3113 \text{ bits}$$

(c) False. A test-once tree can with one attribute creates exactly two regions on the real line, whereas the data may alternate along the line.

(d) True. A test-many tree can define arbitrarily small hyper-rectangles, each containing exactly one example.

Automata and Formal Languages Comprehensive Exam (65 Points)

Fall 2006

Note: this 1-hour exam is *closed book.*

Problem 1 (20 points)

For each of the following statements, provide a high-level proof or an explicit counterexample. You can assume that the various descriptions of regular and context-free languages are equivalent. If you provide a counterexample, you do not need to prove that it is not regular/context-free as long as it is correct.

(a) The union of two regular languages is regular.

(b) The intersection of two regular languages is regular.

(c) The union of two context-free languages is context-free.

(d) The intersection of two context-free languages is context-free.

Problem 2 (12 points)

Classify each of the following languages as being in one of the following classes of languages: *empty, finite, regular, context-free, recursive, recursively enumerable, all languages.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA must always be context-free, but the smallest class that contains all such languages is that of the *regular* languages. *Do not provide explanations. Correct answers receive 3 points, incorrect answers receive -2 points.*

(a) A PSPACE-complete problem.

(b) A subset of a recursive language.

(c) $\{0^n 1^n \mid n \geq 1\}$.

(d) $\{a^n \mid n \text{ is prime}\}$.

Problem 3 (18 points)

Let $L_{ne} = \{M \mid L(M) \neq \emptyset\}$ be the strings that encode a Turing machine that accepts a non-empty language.

(a) Give a high-level proof showing that L_{ne} is a recursively enumerable language.

(b) Give a high-level proof showing that L_{ne} is not recursive (i.e., is undecidable). Assume that the universal language $L_u = \{(M, w) \mid M \text{ accepts } w\}$ is not recursive.

Problem 4 (15 points)

The *subgraph-isomorphism problem* is the following: given graphs G_1 and G_2, does G_1 contain a copy of G_2 as a subgraph? That is, can we find a subset of the nodes of G_1 that, together with the edges among them in G_1, forms an exact copy of G_2? Prove that the subgraph-isomorphism problem is NP-hard.

[Hint: make use of the NP-complete problem CLIQUE.]

Automata and Formal Languages Comprehensive Exam Solutions

Fall 2006

Problem 1 (20 points)

(a) True. See Theorem 4.4 of HMU (page 132) for a proof.

(b) True. See Theorem 4.8 of HMU (page 135) for a proof.

(c) True. Let A and B be CFLs, say generated by grammars G_1 and G_2 with start symbols S_1 and S_2, respectively. Form a grammar generating $A \cup B$ by combining all productions of G_1 and G_2 along with a new start symbol S and new productions $S \to S_1$ and $S \to S_2$. (Also, before combining the grammars, make sure that they have disjoint sets of nonterminals, renaming the nonterminals of one grammar if needed.)

(d) False. For a counterexample, see Examples 7.19 and 7.26 of HMU.

Problem 2 (12 points)

Classify each of the following languages as being in one of the following classes of languages: *empty, finite, regular, context-free, recursive, recursively enumerable, all languages.*

(a) Recursive.

(b) All languages (as the set of all strings is recursive).

(c) Context-free.

(d) Recursive.

Problem 3 (18 points)

(a) See Theorem 9.8 in HMU (page 385) for a proof.

(b) See Theorem 9.9 in HMU (page 386) for a proof.

Problem 4 (20 points)

We give a polynomial-time reduction from CLIQUE to SUBGRAPH ISOMORPHISM. Since CLIQUE is NP-complete, this proves that SUBGRAPH ISOMORPHISM is NP-hard.

We begin with an instance of CLIQUE, specified by a graph G and a positive integer k. We construct an instance of SUBGRAPH ISOMORPHISM by defining G_1 to be the graph G and the graph G_2 to be a complete graph on k nodes. This reduction clearly runs in polynomial time.

Suppose G has a clique of size k. Then this same subset of nodes (viewed as a subgraph of G_1) is isomorphic to G_2. Conversely, if G_2 is isomorphic to a subgraph of G_1, then since G_2 is a complete graph on k nodes, this subset of nodes of G_1 corresponds to a clique of size k in G. These two facts establish the correctness of the reduction.

Compilers Comprehensive, November 2006

This is a 60 minute, closed book exam. Please mark your answers in the blue book.

1. (10 points)

   Suppose we wanted to build a lexical analyzer for a simplified version of C, which had literal strings beginning and ending with double-quote characters, `"like this"`, using an automated tool like Lex or Flex.

   (a) We could try using the regular expression `\".*\"` for the pattern describing the string. In Lex and Flex, "dot" represents any character except newline.

   Why won't this work very well?

   (b) On analyzing the difficulty in the previous problem, we could decide to build a superior variant of Flex that returns the *shortest* string matching the pattern.

   i. Without changing the pattern in the previous part, will string literals now be handled properly?

   ii. Would this change cause serious problems? Please explain.

2. (25 points)

   Consider this context-free grammar for "reverse Polish" arithmetic expressions:

$$
\begin{aligned}
S &\rightarrow E \\
E &\rightarrow E\ E\ op \\
E &\rightarrow id
\end{aligned}
$$

   (a) Write out the canonical collections of LR(0) items for this grammar (you do not need to add a special start production – it's already there).

   (b) Give an example to explain why an LR(1) parser would have more states than an LALR(1) parser for this grammar.

   (c) Is this grammar SLR(1)? Explain.

   (d) Explain why the grammar is not LL(1), and then convert it to a grammar for the same language that is suitable for LL(1) parsing.

3. (15 points)

   Local and *global common sub-expression elimination* are optimizations frequently performed in compilers.

   (a) What are they?

   (b) What kind of analysis is required to perform global common subexpression elimination?

   (c) Can common sub-expression elimination ever slow down a program? Explain your answer.

4. (10 points)

   It is easier to write a type-checker for function calls if the definition of every function (which specifies the argument types and return types) appears before the first call to the function (whose type correctness depends on its arguments and return types).

   However, if a program language allows mutually recursive functions (e.g., f calls g and g calls f), this is not possible. If f is defined first, a call to g will appear in the body of f – before g is defined – or *vice versa*.

   Give at least two examples of how programming languages and their compilers deal with this issue.

Computer Architecture Comprehensive Exam

Exam Instructions
Answer each of the questions included in the exam. Write all of your answers directly on the examination paper, including any work that you wish to be considered for partial credit. The examination is open-book, and you may make use of the text, handouts, your own course notes, and a calculator.

On equations: Wherever possible, make sure to include the equation, the equation rewritten with the numerical values, and the final solution. Partial credit will be weighted appropriately for each component of the problem, and providing more information improves the likelihood that partial credit can be awarded.

On writing code: Unless otherwise stated, you are free to use any of the assembly instructions listed in the Appendix at the back of the book, including pseudoinstructions. You do not need to optimize your MIPS code unless specifically instructed to do so.

On time: You will have one hour to complete this exam. Budget your time and try to leave some time at the end to go over your work. The point weightings correspond roughly to the time each problem is expected to take.

THE STANFORD UNIVERSITY HONOR CODE

The Honor Code is an undertaking of the students, individually and collectively:
(1) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
(2) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

I acknowledge and accept the Honor Code.

Magic Number ______________________________

		Score	Grader
1. Short Answer	(15)	______	______
2. Average Memory Access Time	(15)	______	______
3. Bits and Bytes	(15)	______	______
4. MIPS	(15)	______	______

Total (60) ______

Problem 1: Short Answer (15 points)
Please provide short, concise answers.

(a) [1 points] One benefit of RISC architectures is that they generally provide more general purpose registers. Why is this beneficial?

(b) [1 points] Ben Bitdiddle notices that many RISC architectures have 32 general purpose registers. Ben wonders if more registers are better, why not have even more? Explain to Ben the tradeoffs involved with increasing the number of registers to 64.

(c) [1 points] One of the differences between RISC architectures and CISC architecture is supposed to be the reduced types of instructions available. Ben Bitdiddle thinks it would be a good idea to simplify the instruction set even more to remove the special case instructions that take immediate operands such as “li”, “addi”, etc. Explain to Ben why this might not be such a good idea.

(d) [2 points] Some procedure-linkage conventions require the caller (rather than the callee) pop procedure arguments from the stack on return. Why is it awkward for the callee to perform this step?

(e) [2 points] A cache memory can speed up computation by eliminating some references to main memory. Does its use require clever programming techniques? Explain.

(f) [2 points] Must a user program written for use with a non-paged memory system be modified in order to be used with a paged memory system? Explain.

(g) [2 points] Explain how a memory system that pages to secondary storage depends on locality of reference for efficient operation.

(h) [2 points] Program A consists of 1000 consecutive add instructions, while program B consists of a loop that executes a single add instruction 1000 times. You run both programs on a certain machine and find that program B consistently executes faster. Explain.

(i) [2 points] A simple strategy for aborting a CISC machine instruction when page faults occur would be simply to reset the program counter and stack pointer to the values they had when the current instruction began to execute. List a serious problem with this strategy.

Problem 2: Average Memory Access Time (15 points)

For this problem, assume that you have a processor with a cache connected to main memory via a bus. A cache access takes 1 cycle. A successful cache acces (a hit) finishes within that cycle. On an unsuccessful access (a miss) additional work must be performed to fetch a block from main memory over the bus. A bus transaction consists of one cycle to send the address to memory, four cycles of idle time for main-memory access, and then one cycle to transfer each word in the block to the cache. (Assume that the processor continues execution only after the last word of the block has arrived.)

Block size (B)	**Miss ratio (m), %**
1 word	3.40%
4 words	1.00%
16 words	0.40%
64 words	0.25%
256 words	0.19%

In other words, if the block size is B words (at 32 bits/word), a cache miss will cost 1 + 4 + B cycles. The adjacent table gives the average cache miss rates of a 1 megabyte cache for various block sizes.

(a) [3 points] Write an expression for the average memory access time for a 1-MByte cache and a B-word block size (in terms of m and B).

(b) [4 points] What block size yields the best average memory access time?

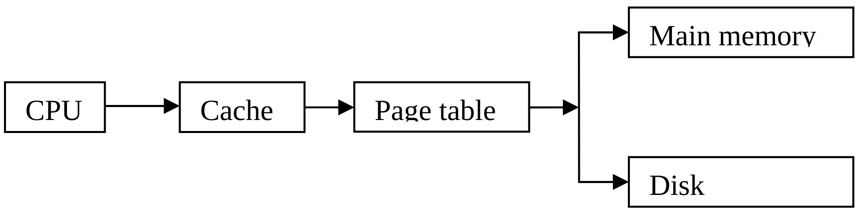
(c) [4 points] If bus contention adds three cycles to the main-memory access time, which block size yields the best average memory access time.

(d) [4 points] Ignoring bus contention, if the bus width is doubled to 64 bits, what is the optimal block size?

Problem 3: Bits and Bytes (15 points)

The figure below shows a CPU and its memory system. The computer features

- a single processor
- 32-bit virtual addresses
- a cache of 2^{10} sets that are four-way set-associative and have 8-byte blocks
- a main memory of 2^{26} bytes; and a page size of 2^{12} bytes.



(a) [1 points] Does this system cache virtual or physical addresses?

(b) [3 points] How many bytes of data from memory can the cache hold? (Don't count tags)

(c) [4 points] In the cache, each block of data must have a tag associated with it. How many bits long are these tags?

(d) [4 points] How many comparators are needed to build this cache while allowing single cycle access?

(e) [3 points] At any one time, what is the greatest number of page-table entries that can have their valid bit set to 1?

Problem 4: MIPS (15 points)

(a) [6 points] Ben Bitdiddle has been hired by Syl Valle to recover some code lost in a hard drive crash. He was able to recover some fragments of both the C code and its compiled MIPS assembly code from the drive. Help him out by filling in the parts he was unable to recover. Remember, in a MIPS function call, ra contains the return address, a0 contains the first argument, and upon return v0 contains the return value. The s# registers must be preserved across procedure calls. Also remember that standard MIPS has a branch delay slot.

```
int f(int n) {
    int f2;
    if (n<2)
        return n;
    f2 = f(n-2);
    return <A>;
}
```

```
00000000 <f>:
   0:   addiu   sp,sp,-40
   4:   sw      ra,36(sp)
   8:   sw      s8,32(sp)
   c:   move    s8,sp
  10:   sw      a0,40(s8)
  14:   lw      v0,40(s8)
  18:   nop
  1c:   slti    v0,v0,2
  20:   beqz    v0,3c <f+0x3c>
  24:   nop
  28:   lw      v0,40(s8)
  2c:   nop
  30:   sw      v0,<B>
  34:   j       84 <f+0x84>
  38:   nop
  3c:   lw      v0,40(s8)
  40:   nop
  44:   addiu   v0,v0,-2
  48:   move    a0,v0
  4c:   jal     0 <f>
  50:   nop
  54:   sw      v0,16(s8)
  58:   lw      v0,40(s8)
  5c:   nop
  60:   addiu   v0,v0,-1
  64:   move    a0,v0
  68:   jal     0 <f>
  6c:   nop
  70:   move    v1,v0
  74:   lw      <C>,16(s8)
  78:   nop
  7c:   addu    v1,v1,v0
  80:   sw      v1,24(s8)
  84:   lw      v0,24(s8)
  88:   move    sp,s8
  8c:   lw      ra,36(sp)
  90:   lw      s8,32(sp)
  94:   addiu   sp,sp,40
  98:   jr      ra
  9c:   nop
```

<A>:

<B>:

<C>:

(b) [3 points] Ben is pleased with your work and asks you to help him optimize the assembly.
Optimize instructions 0x44 – 0x50.

(c) [6 points] Ben wants to take out some of the nops in the code and is interested in seeing how the resulting code is executed on a pipelined processor with a 5-stage pipeline. Branches are resolved in the decode stage and have a delay slot. All memory accesses take 1 clock cycle.

The table below shows the instruction stream starting at instruction 0x14 with some nops taken out (the branch is taken). Fill in the remainder of the pipeline diagram. Show all cases of data forwarding (or "bypassing") using arrows to connect the source and destination stages.

Instruction	Cycle															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw v0,40(s8)	F	D	X	M	W											
slti v0,v0,2		F	D	-	X	M	W									
beqz v0,3c			F	-												
nop																
lw v0,40(s8)																
addiu v0,v0,-2																
move a0,v0																
jal 0																

Stanford University 16 November 2006

Computer Architecture Comprehensive Exam

Exam Instructions
Answer each of the questions included in the exam. Write all of your answers directly on the examination paper, including any work that you wish to be considered for partial credit. The examination is open-book, and you may make use of the text, handouts, your own course notes, and a calculator.

On equations: Wherever possible, make sure to include the equation, the equation rewritten with the numerical values, and the final solution. Partial credit will be weighted appropriately for each component of the problem, and providing more information improves the likelihood that partial credit can be awarded.

On writing code: Unless otherwise stated, you are free to use any of the assembly instructions listed in the Appendix at the back of the book, including pseudoinstructions. You do not need to optimize your MIPS code unless specifically instructed to do so.

On time: You will have one hour to complete this exam. Budget your time and try to leave some time at the end to go over your work. The point weightings correspond roughly to the time each problem is expected to take.

THE STANFORD UNIVERSITY HONOR CODE

The Honor Code is an undertaking of the students, individually and collectively:

(1) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

(2) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

I acknowledge and accept the Honor Code.

Magic Number ______________________________

		Score	Grader
1. Short Answer	(15)	______	______
2. Average Memory Access Time	(15)	______	______
3. Bits and Bytes	(15)	______	______
4. MIPS	(15)	______	______

Total (60) ______

Problem 1: Short Answer (15 points)
Please provide short, concise answers.

(a) [1 points] One benefit of RISC architectures is that they generally provide more general purpose registers. Why is this beneficial?

Reuse register values without spilling to memory

(b) [1 points] Ben Bitdiddle notices that many RISC architectures have 32 general purpose registers. Ben wonders if more registers are better, why not have even more? Explain to Ben the tradeoffs involved with increasing the number of registers to 64.

Smaller immediate values
OR
Slower register file access
OR
Opportunity cost for hardware for more registers than compiler can use
OR
More registers to context switch

(c) [1 points] One of the differences between RISC architectures and CISC architecture is supposed to be the reduced types of instructions available. Ben Bitdiddle thinks it would be a good idea to simplify the instruction set even more to remove the special case instructions that take immediate operands such as “li”, “addi”, etc. Explain to Ben why this might not be such a good idea.

Common operations will now require multiple instructions without any corresponding improvement in cycle time

(d) [2 points] Some procedure-linkage conventions require the caller (rather than the callee) pop procedure arguments from the stack on return. Why is it awkward for the callee to perform this step?

Callee does not always statically know the number of arguments passed, such as when there are variable number of arguments as in printf

(e) [2 points] A cache memory can speed up computation by eliminating some references to main memory. Does its use require clever programming techniques? Explain.

1 point – yes, optimal performance can require cache aware or cache oblivious code

1 point – no, program can benefit from cache without being modifed

2 points for yes and no explanation

(f) [2 points] Must a user program written for use with a non-paged memory system be modified in order to be used with a paged memory system? Explain.

1 point – yes, optimal performance can require page aware algorithm or working set management

1 point – no, program can use more than memory than physical memory with minimal impact if working set fits in physical memory

2 points for yes and no explanation (also points for page locality needed for high TLB hit rate)

(g) [2 points] Explain how a memory system that pages to secondary storage depends on locality of reference for efficient operation.

Without locality, the memory system would perform a disk speeds as every access could require an access to disk. A working set that fits within physical memory for efficient operations.

(h) [2 points] Program A consists of 1000 consecutive add instructions, while program B consists of a loop that executes a single add instruction 1000 times. You run both programs on a certain machine and find that program B consistently executes faster. Explain.

Program B fits easily in the instruction cache but program A takes more time to be fetched.

(i) [2 points] A simple strategy for aborting a CISC machine instruction when page faults occur would be simply to reset the program counter and stack pointer to the values they had when the current instruction began to execute. List a serious problem with this strategy.

Non-idempotent side effects need to be reversed.

Problem 2: Average Memory Access Time (15 points)

Block size (B)	Miss ratio (m), %
1 word	3.40%
4 words	1.00%
16 words	0.40%
64 words	0.25%
256 words	0.19%

For this problem, assume that you have a processor with a cache connected to main memory via a bus. A cache access takes 1 cycle. A successful cache acces (a hit) finishes within that cycle. On an unsuccessful access (a miss) additional work must be performed to fetch a block from main memory over the bus. A bus transaction consists of one cycle to send the address to memory, four cycles of idle time for main-memory access, and then one cycle to transfer each word in the block to the cache. (Assume that the processor continues execution only after the last word of the block has arrived.)
In other words, if the block size is B words (at 32 bits/word), a cache miss will cost 1 + 4 + B cycles. The adjacent table gives the average cache miss rates of a 1 megabyte cache for various block sizes.

(a) [3 points] Write an expression for the average memory access time for a 1-MByte cache and a B-word block size (in terms of m and B).

AMAT = 1 + m (1 + 4 + B)

(b) [4 points] What block size yields the best average memory access time?

For B == 1	1 + .0340(1+4+1)	= 1.204	
For B == 4	1 + .0100(1+4+4)	= 1.09	
For B == 16	1 + .0040(1+4+16)	= 1.084	block size of 16 is best
For B == 64	1 + .0025(1+4+64)	= 1.1725	
For B == 256	1 + .0019(1+4+256)	= 1.4959	

(c) [4 points] If bus contention adds three cycles to the main-memory access time, which block size yields the best average memory access time.

For B == 1	1 + .0340(3+1+4+1)	= 1.306	
For B == 4	1 + .0100(3+1+4+4)	= 1.12	
For B == 16	1 + .0040(3+1+4+16)	= 1.096	block size of 16 is best
For B == 64	1 + .0025(3+1+4+64)	= 1.18	
For B == 256	1 + .0019(3+1+4+256)	= 1.5016	

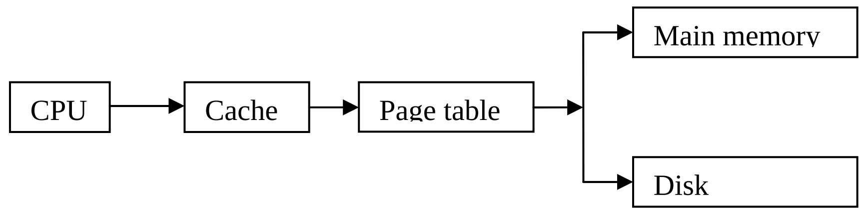
(d) [4 points] Ignoring bus contention, if the bus width is doubled to 64 bits, what is the optimal block size?

For B == 1	1 + .0340(1+4+1)	= 1.204	
For B == 4	1 + .0100(1+4+2)	= 1.07	
For B == 16	1 + .0040(1+4+8)	= 1.052	block size of 16 is best
For B == 64	1 + .0025(1+4+32)	= 1.0925	
For B == 256	1 + .0019(1+4+128)	= 1.2527	

Problem 3: Bits and Bytes (15 points)

The figure below shows a CPU and its memory system. The computer features

- a single processor
- 32-bit virtual addresses
- a cache of 2^{10} sets that are four-way set-associative and have 8-byte blocks
- a main memory of 2^{26} bytes; and a page size of 2^{12} bytes.



(a) [1 points] Does this system cache virtual or physical addresses?

virtual addresses

(b) [3 points] How many bytes of data from memory can the cache hold? (Don't count tags)

2^{10} sets * 4 lines/set * 8 bytes/line = 2^{15} bytes = 32kb

(c) [4 points] In the cache, each block of data must have a tag associated with it. How many bits long are these tags?

32 bits in virtual address – 10 bits in index – 3 bits in offset = 19 bits for tag

(d) [4 points] How many comparators are needed to build this cache while allowing single cycle access?

1 per each set way = 4

(e) [3 points] At any one time, what is the greatest number of page-table entries that can have their valid bit set to 1?

$2^{32}/2^{12}$ bytes/page = 2^{20} pages.
All of them can valid because then can all alias the same physical page.

Problem 4: MIPS (15 points)

(a) [6 points] Ben Bitdiddle has been hired by Syl Valle to recover some code lost in a hard drive crash. He was able to recover some fragments of both the C code and its compiled MIPS assembly code from the drive. Help him out by filling in the parts he was unable to recover. Remember, in a MIPS function call, ra contains the return address, a0 contains the first argument, and upon return v0 contains the return value. The s# registers must be preserved across procedure calls. Also remember that standard MIPS has a branch delay slot.

```
int f(int n) {
    int f2;
    if (n<2)
        return n;
    f2 = f(n-2);
    return <A>;
}
```

```
00000000 <f>:
   0:   addiu   sp,sp,-40
   4:   sw      ra,36(sp)
   8:   sw      s8,32(sp)
   c:   move    s8,sp
  10:   sw      a0,40(s8)
  14:   lw      v0,40(s8)
  18:   nop
  1c:   slti    v0,v0,2
  20:   beqz    v0,3c <f+0x3c>
  24:   nop
  28:   lw      v0,40(s8)
  2c:   nop
  30:   sw      v0,<B>
  34:   j       84 <f+0x84>
  38:   nop
  3c:   lw      v0,40(s8)
  40:   nop
  44:   addiu   v0,v0,-2
  48:   move    a0,v0
  4c:   jal     0 <f>
  50:   nop
  54:   sw      v0,16(s8)
  58:   lw      v0,40(s8)
  5c:   nop
  60:   addiu   v0,v0,-1
  64:   move    a0,v0
  68:   jal     0 <f>
  6c:   nop
  70:   move    v1,v0
  74:   lw      <C>,16(s8)
  78:   nop
  7c:   addu    v1,v1,v0
  80:   sw      v1,24(s8)
  84:   lw      v0,24(s8)
  88:   move    sp,s8
  8c:   lw      ra,36(sp)
  90:   lw      s8,32(sp)
  94:   addiu   sp,sp,40
  98:   jr      ra
  9c:   nop
```

<A>: f(n-1) + f2

<B>: 24(s8)

<C>: v0

(b) [3 points] Ben is pleased with your work and asks you to help him optimize the assembly.
Optimize instructions 0x44 – 0x50.

44: jal 0 <f>
48: addui a0, v0, -2
52: nop
56: nop

(c) [6 points] Ben wants to take out some of the nops in the code and is interested in seeing how the resulting code is executed on a pipelined processor with a 5-stage pipeline. Branches are resolved in the decode stage and have a delay slot. All memory accesses take 1 clock cycle.

The table below shows the instruction stream starting at instruction 0x14 with some nops taken out (the branch is taken). Fill in the remainder of the pipeline diagram. Show all cases of data forwarding (or "bypassing") using arrows to connect the source and destination stages.

Instruction	Cycle															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
`lw v0,40(s8)`	F	D	X	M↘	W											
`slti v0,v0,2`		F	D	-	X↘	M	W									
`beqz v0,3c`			F	-	-	D										
`nop`						F	D	X	M	W						
`lw v0,40(s8)`							F	D	X	M↘	W					
`addiu v0,v0,-2`								F	D	-	X↘	M	W			
`move a0,v0`									F	-	D	X	M	W		
`jal 0`											F	D				

Stanford University Computer Science Department
2006 Comprehensive Exam in Databases

- The exam is *open book and notes*, but not open computer.
- There are six problems on the exam, with a varying number of points for each problem and subproblem for a total of 60 points (i.e., one point per minute). It is suggested that you look through the entire exam before getting started, in order to plan your strategy.
- Please write your solutions in the spaces provided on the exam. Make sure your solutions are neat and clearly marked.
- *Simplicity and clarity of solutions will count.* You may get as few as 0 points for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.

MAGIC NUMBER: ____________

Problem	1	2	3	4	5	6	TOTAL
Max. points	4	14	12	12	12	6	60
Points							

1. *Data Models* **(4 points)**

   (a) Why does the E/R model require a notion of "weak entity set"? Be as specific as you can. (For example, "Because some entity sets don't have a key" is not an acceptable answer.)

   (b) Why do object-oriented models like ODL *not* have a notion equivalent to weak entity sets? Again, be as specific as you can.

2. *Relational Algebra* **(14 points)**

   Suppose there are relations $R(A, B)$ and $S(B, C)$. Write the following queries in *relational algebra* using standard operators only. You may write either single expressions or sequences of assignments to temporary variables, the last of which is the desired result. If you want your answers to be considered for partial credit you may provide some explanation of your reasoning, but it is sufficient just to provide the correct relational algebra queries.

   (a) (4 points) The tuples of R such that their B-component is neither a B-component of a tuple of S nor a C-component of a tuple of S.

(b) (4 points) The A-values that appear in at least two distinct tuples of R.

(c) (6 points) The A-values a such that:

$i.$ a appears in at least one tuple of R, and

$ii.$ Every B-value b, for which (a, b) is a tuple of R, appears as a B-value in at least one tuple of S.

3. *Relational Design* **(12 points)**

Suppose we are given relation $R(A, B, C, D)$ with functional dependencies $AB \rightarrow C$, $CD \rightarrow A$, and $B \rightarrow D$. Answer each of the following four questions. In each part, you may choose to justify your answer for partial credit, but it is sufficient just to provide an answer.

(a) (4 points) Find all the (minimal) keys of R.

(b) (2 points) How many superkeys does R have?

(c) (3 points) Of the given FD's, which violate Boyce-Codd normal form?

(d) (3 points) Of the given FD's, which violate third normal form?

4. *XML* **(12 points)**

Consider the following four DTDs for XML documents containing data about books and authors. You may assume that `isbn` and `ssn` are "keys" for books and authors, in the sense that no two actual books have the same `isbn`, and no two actual authors have the same `ssn`.

```
DTD1: <!DOCTYPE BA [
         <!ELEMENT BA (Book*)>
         <!ELEMENT Book (Title, Author+)>
         <!ELEMENT Title (#PCDATA)>
         <!ELEMENT Author>
         <!ATTLIST Author ssn ID name CDATA> ]>

DTD2: <!DOCTYPE BA [
         <!ELEMENT BA (Author*)>
         <!ELEMENT Author (Book)>
         <!ATTLIST Author ssn ID name CDATA>
         <!ELEMENT Book (Title)>
         <!ELEMENT Title (#PCDATA)> ]>

DTD3: <!DOCTYPE BA [
         <!ELEMENT BA (Book*, Author*)>
         <!ELEMENT Book (Title)>
         <!ATTLIST Book isbn ID author IDREF>
         <!ELEMENT Title (#PCDATA)>
         <!ELEMENT Author>
         <!ATTLIST Author ssn ID name CDATA> ]>

DTD4: <!DOCTYPE BA [
         <!ELEMENT BA (Author*, Book*)>
         <!ELEMENT Author>
         <!ATTLIST Author ssn ID book IDREFS>
         <!ELEMENT Book (Title)>
         <!ATTLIST Book isbn ID>
         <!ELEMENT Title (#PCDATA)> ]>
```

(a) Which of the DTDs can be used for XML documents that encode *many-many* relationships between books and authors? (circle those that can) `DTD1 DTD2 DTD3 DTD4`

(b) Which of the DTDs can be used for XML documents that encode *many-one* relationships from books to authors? (circle those that can) `DTD1 DTD2 DTD3 DTD4`

(c) Which of the DTDs can be used for XML documents that encode *one-many* relationships from books to authors? (circle those that can) `DTD1 DTD2 DTD3 DTD4`

(d) Which of the DTDs can be used for XML documents that encode *one-one* relationships between books and authors? (circle those that can) `DTD1 DTD2 DTD3 DTD4`

5. *SQL* **(12 points)**

   Consider the following SQL schema for a database about schools in a university and departments in those schools. Data types are omitted for brevity.

```
create table School(schoolName primary key, dean)
create table Dept(deptName primary key,
                  schoolName references School on delete cascade)
```

   Write a *single SQL statement* that deletes all schools with more than 10 departments and also deletes all departments in those schools.

6. *Transactions* **(6 points)**

   Consider a relation `Temperatures(time,temp)` and the following transaction `T` with SQL isolation level `REPEATABLE READ`:

```
T: (Q1) Select Avg(temp) From Temperatures;
   (Q2) Select Avg(temp) From Temperatures;
```

   Is it possible for a concurrently-running transaction to cause queries `Q1` and `Q2` to return different values? If so, show the simplest such transaction. If not, explain why not.

Stanford University Computer Science Department
2006 Comprehensive Exam in Databases
SAMPLE SOLUTION

1. *Data Models*

   (a) Why does the E/R model require a notion of "weak entity set"? Be as specific as you can. (For example, "Because some entity sets don't have a key" is not an acceptable answer.)

   ```
   There are entity sets whose members cannot be distinguished
   by their own attributes, but require the help of entities
   to which they are linked by relationships. These are weak
   entity sets.
   ```

   (b) Why do object-oriented models like ODL *not* have a notion equivalent to weak entity sets? Again, be as specific as you can.

   ```
   Since object-oriented models have a notion of object-identity,
   all members of a class are distinguishable from one another,
   even if they have the same attribute-values.
   ```

2. *Relational Algebra*

   Suppose there are relations $R(A, B)$ and $S(B, C)$. Write the following queries in *relational algebra* using standard operators only. You may write either single expressions or sequences of assignments to temporary variables, the last of which is the desired result. If you want your answers to be considered for partial credit you may provide some explanation of your reasoning, but it is sufficient just to provide the correct relational algebra queries.

   (a) The tuples of R such that their B-component is neither a B-component of a tuple of S nor a C-component of a tuple of S.

   $RA := \pi_A(R)$
   $Svals := \pi_B(S) \cup \pi_C(S)$
   $Ans := R - (RA \times Svals)$

   (b) The A-values that appear in at least two distinct tuples of R.

   $\pi_A(\sigma_{A=D \wedge B \neq E}(R \times \rho_{T(D,E)}(R)))$

(c) The A-values a such that:

$i.$ a appears in at least one tuple of R, and

$ii.$ Every B-value b, for which (a, b) is a tuple of R, appears as a B-value in at least one tuple of S.

RA := $\pi_A(R)$
$BadB$:= $\pi_B(R) - \pi_B(S)$
$BadA$:= $\pi_A(R \cap (RA \times BadB)$
Ans := $RA - BadA$

3. *Relational Design*

Suppose we are given relation $R(A, B, C, D)$ with functional dependencies $AB \rightarrow C$, $CD \rightarrow A$, and $B \rightarrow D$. Answer each of the following four questions. In each part, you may choose to justify your answer for partial credit, but it is sufficient just to provide an answer.

(a) Find all the (minimal) keys of R.

AB and BC

(b) How many superkeys does R have?

6 (all sets that contain, not necessarily properly, either AB or BC or both)

(c) Of the given FD's, which violate Boyce-Codd normal form?

$CD \rightarrow A$ and $B \rightarrow D$

(d) Of the given FD's, which violate third normal form?

Only $B \rightarrow D$

4. *XML*

Consider the following four DTDs for XML documents containing data about books and authors. You may assume that `isbn` and `ssn` are "keys" for books and authors, in the sense that no two actual books have the same `isbn`, and no two actual authors have the same `ssn`.

```
DTD1: <!DOCTYPE BA [
         <!ELEMENT BA (Book*)>
         <!ELEMENT Book (Title, Author+)>
         <!ELEMENT Title (#PCDATA)>
         <!ELEMENT Author>
         <!ATTLIST Author ssn ID name CDATA> ]>

DTD2: <!DOCTYPE BA [
         <!ELEMENT BA (Author*)>
         <!ELEMENT Author (Book)>
         <!ATTLIST Author ssn ID name CDATA>
         <!ELEMENT Book (Title)>
         <!ELEMENT Title (#PCDATA)> ]>

DTD3: <!DOCTYPE BA [
         <!ELEMENT BA (Book*, Author*)>
         <!ELEMENT Book (Title)>
         <!ATTLIST Book isbn ID author IDREF>
         <!ELEMENT Title (#PCDATA)>
         <!ELEMENT Author>
         <!ATTLIST Author ssn ID name CDATA> ]>

DTD4: <!DOCTYPE BA [
         <!ELEMENT BA (Author*, Book*)>
         <!ELEMENT Author>
         <!ATTLIST Author ssn ID book IDREFS>
         <!ELEMENT Book (Title)>
         <!ATTLIST Book isbn ID>
         <!ELEMENT Title (#PCDATA)> ]>
```

(a) Which of the DTDs can be used for XML documents that encode *many-many* relationships between books and authors? (circle those that can) `DTD4`

(b) Which of the DTDs can be used for XML documents that encode *many-one* relationships from books to authors? (circle those that can) `DTD3 DTD4`

(c) Which of the DTDs can be used for XML documents that encode *one-many* relationships from books to authors? (circle those that can) `DTD1 DTD4`

(d) Which of the DTDs can be used for XML documents that encode *one-one* relationships between books and authors? (circle those that can) `DTD1 DTD2 DTD3 DTD4`

5. *SQL*

Consider the following SQL schema for a database about schools in a university and departments in those schools. Data types are omitted for brevity.

```
create table School(schoolName primary key, dean)
create table Dept(deptName primary key,
                  schoolName references School on delete cascade)
```

Write a *single SQL statement* that deletes all schools with more than 10 departments and also deletes all departments in those schools.

```
delete from School
where 10 < (select count(*) from Dept
            where Dept.schoolName = School.schoolName)
```

6. *Transactions*

Consider a relation `Temperatures(time,temp)` and the following transaction `T` with SQL isolation level `REPEATABLE READ`:

```
T: (Q1) Select Avg(temp) From Temperatures;
   (Q2) Select Avg(temp) From Temperatures;
```

Is it possible for a concurrently-running transaction to cause queries `Q1` and `Q2` to return different values? If so, show the simplest such transaction. If not, explain why not.

Yes. This transaction could run between `Q1` *and* `Q2`:

```
insert into Temperatures values (12:00:00, 1000)
```

Computer Graphics Comprehensive Exam

Computer Science Department
Stanford University
Fall 2006

NAME:

Note: This is exam is *closed-book.*

The exam consists of 5 questions. Each question is worth 20 points. Please answer all the questions in the space provided, overflowing on to the back of the page if necessary.

You have 60 minutes to complete the exam.

1. [20 points] Rotations

A 2D rotation about the origin by θ is represented by the following matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

1A [10 points] Show that $R(\theta) \bullet R(\varphi) = R(\theta + \varphi)$. That is, show that multiplying two rotation matrices yields another rotation matrix; the angle of rotation of the resulting matrix is the sum of the angle of rotations of the two product matrices.

1B [10 points] Show that $R^{-1}(\theta) = R(-\theta)$. That is, show that the inverse of the rotation matrix is equal to a rotation in the opposite direction. In order to solve this problem, you need to find the inverse of the rotation matrix.

2. [20 points] Color coordinates.

2A [5 points]. Why do displays such as cathode-ray tubes and liquid crystal displays use three color primaries, red, green and blue? That is, why don't they need more colors?

2B [5 points]. Why do color-laser and ink-jet printers use 4 or more colors? For example, many printers use cyan, magenta, yellow and black inks, and high-end printers often use 6 or more colors.

2C [10 points]. What is an (1) *absolute* color coordinate system, a (2) *relative* color coordinate system, and an (3) *intuitive* or artistic color coordinate system. Give a one sentence definition of each type of coordinate system.

3. [20 points] Texture mapping.

Texture mapping is one of the most important operations in a modern 3D graphics system. Textures are images that are applied to a surface. The basic texture map drawing loop is:

```
for (x,y) inside triangle
    (u,v) = TransformFromPixelSpaceToTextureSpace(x,y)
    color = TextureLookup(texture, u, v)
```

Here, (x,y) is the location of a pixel inside the triangle being drawn, and (u,v) is the location of the pixel in texture coordinates. Assume pixel and texture coordinates are floating point numbers, and that integral values fall on pixel and texel centers. For example, (0.0,0.0) is the first pixel/texel, (1.0,0.0) is the second pixel/texel, etc.

3A [5 points] Images and textures are represented as images, and images are represented as 2D arrays of values. Yet, texture coordinates have higher precision. That is, texture coordinates are not necessarily integers. Why should texture coordinates have additional precision?

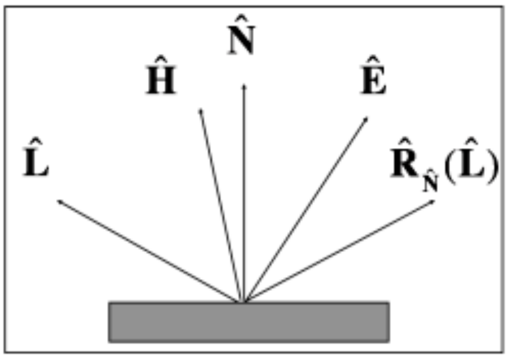
3B [5 points] One way to implement `TextureLookup` is by using *nearest-neighbor interpolation*. Write pseudo-code to implement nearest neighbor interpolation.

3C [5 points] A better way to implement `TextureLookup` is by using *bilinear interpolation*. Write pseudo-code to implement bilinear interpolation.

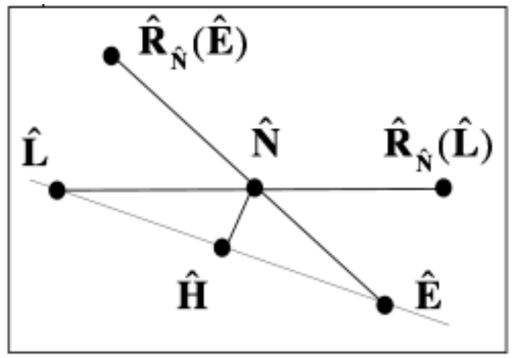
3D [5 points] Bilinear interpolation is much more expensive than nearest-neighbor interpolation, yet all high quality implementation of texture mapping include this type of interpolation. Why, according to theory, is it preferable to use bilinear interpolation?

4. [20 points] Highlights.

The following figure shows the geometry of reflection. $\hat{\mathbf{N}}$ is a unit vector normal to the surface at the point of reflection. $\hat{\mathbf{L}}$ is a unit vector from the surface to the light. $\hat{\mathbf{E}}$ is a unit vector from the surface to the eye. And, $\hat{\mathbf{H}}$ is the unit halfway vector $\hat{\mathbf{H}} = \dfrac{\hat{\mathbf{L}} + \hat{\mathbf{E}}}{\left|\hat{\mathbf{L}} + \hat{\mathbf{E}}\right|}$.



side view



top view

There are two other vectors of interest. These are the reflections of $\hat{\mathbf{L}}$ and $\hat{\mathbf{E}}$ about the normal. These vectors may be computed with the reflection transformation, $\hat{\mathbf{R}}_{\hat{\mathbf{N}}}(\mathbf{V}) = \mathbf{V} - 2\left(\hat{\mathbf{N}} \bullet \mathbf{V}\right)\mathbf{V}$, which reflects the vector $\mathbf{V}$ about the normal $\hat{\mathbf{N}}$. With these definitions, the Phong model is $\left(\hat{\mathbf{R}}_{\hat{\mathbf{N}}}(\hat{\mathbf{L}}) \bullet \hat{\mathbf{E}}\right)^s$ and the Blinn model is $\left(\hat{\mathbf{H}} \bullet \hat{\mathbf{N}}\right)^s$.

4A [5 points]. The parameter *s* is controlled by the application. What is the significance of this parameter. What happens when you change s?

4B [15 points]. Suppose $\hat{\mathbf{N}}$, $\hat{\mathbf{L}}$, and $\hat{\mathbf{E}}$ are in the same plane. What is the relationship between $\left(\hat{\mathbf{R}}_{\hat{\mathbf{N}}}(\hat{\mathbf{E}}) \bullet \hat{\mathbf{L}}\right)^s$ and $\left(\hat{\mathbf{H}} \bullet \hat{\mathbf{N}}\right)^s$. Hint, is there a relationship between the angles between the various vectors?

5 [20 points] Visualization.

Your friend Sally the brilliant mathematics graduate student asks for your help because of your expertise in computer graphics. She wants to visualize the function

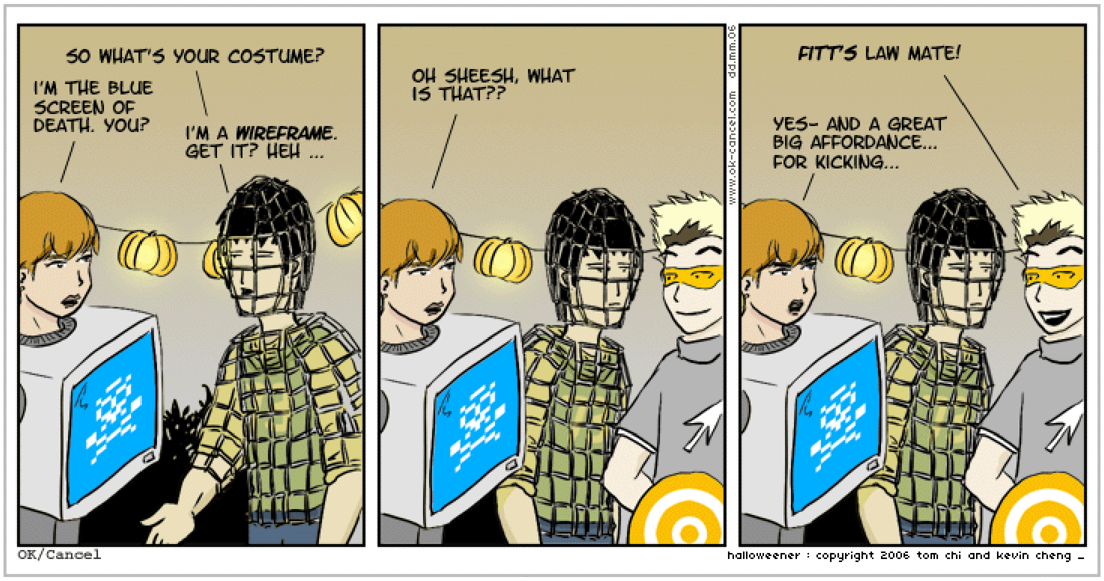
$$f(x,y,z) = 1 - x^2 + y^2 + z^2 + \sin 4x + \sin 4y + \sin 4z$$

5A [15 points]. How would you make a picture of her function? Assume you have access to a basic graphics library, but not an advanced visualization system such as Matlab, Mathematica or a graphing calculator. That is, break down the visualization process into standard algorithms and basic graphic primitives. Outline the method you would use. Make sure to include enough detail about each method to ensure that they work properly.

5B [5 points]. Does your picture show the important properties of her function? What features might be of interest, and how would you guarantee that those features were clear in your picture?

Comprehensive Exam in Human-Computer Interaction

Scott Klemmer, 14 November 2006



1	**/ 20**
2	**/ 16**
3	**/ 15**
4	**/ 16**
5	**/ 15**
6	**/ 18**
Total	**/ 100**

This is an open book, open note exam. You may use a laptop computer and the internet to reference course readings and lecture notes only; all other use is prohibited.

MAGIC NUMBER:

1. Pointing and Fitts's Law [20 Points]

(a) [12 Points]

The window shown to the left is the movable tool palette for selecting drawing tools in Adobe Photoshop. The user selects a tool by clicking on one of the icons in the palette. Assume the user is running Photoshop on a standard desktop machine using a mouse.

The Fitts's formula for Movement Time (MT), given in Hinckley's paper, is

$$MT = a + b \log_2 (A/W + 1)$$

For each of the variables (a, b, A, and W), describe what it measures, and suggest how to redesign tool selection in Photoshop to decrease MT based on a change to that variable.

(b) [8 Points]

The MacOS has a single menu bar at the top of the screen, which changes contents to suit whatever application is currently active. Windows puts a menu bar near the top of each active window, below the windows' title bars.

i. From a cognitive perspective, what principles argue for doing it the Windows way?

ii. From the perspective of Fitts's Law, what is the advantage of the Macintosh way?

2. User Testing [16 Points]

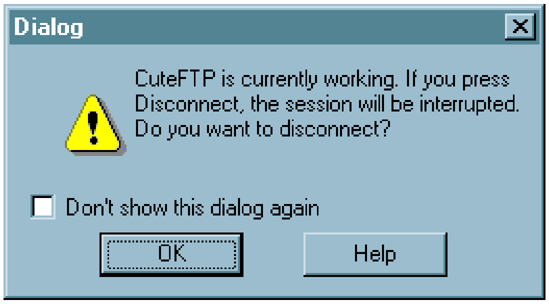
To make use of your newly acquired expertise in HCI, you volunteered as a peer reviewer for an academic conference, and you were invited to review a paper submission entitled "Improving Data Entry with the GOOBER System." Read the following excerpt from the paper's "Evaluation" section.

> To evaluate my system, I invited undergraduate CS students enrolled in my course to participate in an experiment for extra credit. I explained to them that they would be comparing an old system for data entry (Microsoft Excel) to a new data entry system that I had been building for three years. Five participants were asked to enter one page of data, first using Microsoft Excel, and then using my GOOBER system. I then asked participants to rate both systems on an integer scale from 1 (bad) to 4 (phenomenally awesome). Excel received an average rating of 3, while GOOBER received an average rating of 3.5. Several subjects commented that GOOBER was "easy to use" and "intuitive."

Point out four problems with the experimental methodology, and suggest a way to fix each problem. **[4 points each]**

3. Heuristic Evaluation [15 Points]

(a) While performing a **heuristic evaluation** of a file transfer program you see the following dialog. Describe one problem with the dialog by identifying a **heuristic,** describing the **problem** (in one sentence) and describing a **potential fix** in another sentence. **[6 points]**



Heuristic:

Problem:

Fix:

(b) One of Nielson's Heuristics is “Recognition Rather than Recall.” Why is this heuristic sometimes more difficult to satisfy in speech user interfaces than in graphical user interfaces? What is an example of a technique you can use to address this issue? What is a major trade-offs your solution may introduce? **[9 points]**

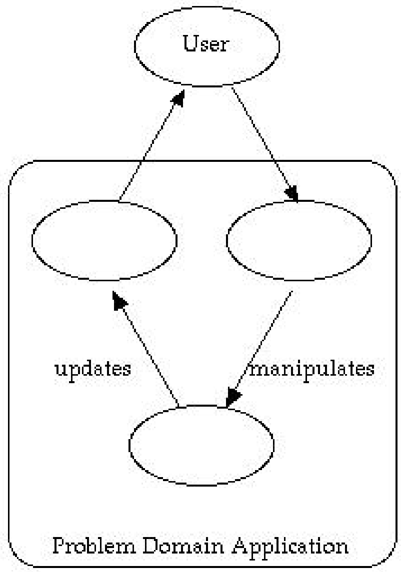
4. Design Patterns [16 Points]

(a) Draw an example sketch of the use of **Breadcrumbs**, and describe briefly (in <3 sentences) what this design pattern is used for. **[5 Points]**

(b) Describe the basic principle of the **Process Funnel** design pattern and how it is used in websites. **[5 Points]**

(c) The **Model-View-Controller** pattern is often used in building web applications.

i. Complete the **Model-View-Controller** diagram by writing text in the empty circles. **[2 Points]**



ii. Describe the primary advantages of the MVC pattern, in three sentences or fewer. **[4 Points]**

5. Task Analysis [15 points]

Microsoft has just hired you as an interface designer. Your first assignment is to design a newspaper layout tool.

This software is targeted towards journalism classes in several junior high schools in low-income neighborhoods. Students in these courses are less familiar with computers, as most do not have them at home. They are more familiar with video arcade games and Xbox-style home entertainment systems. Thus, they currently do not use many software tools. Instead, they write stories with typewriters and layout articles using scissors and glue.

The teaching staff is happy with the typewriter/scissors approach. However, they are willing to give the new software a chance, given that it will support the current tasks and provide more flexibility in story editing and newspaper layout.

Ask and answer five standard task analysis questions about the newspaper layout tool. For example: who are the users? Your answers to these questions should come directly from the above assignment and *logical inferences* about the assignment. Answers should be specific and detailed.

6. Experimental Design [18 Points]

(a) Circle the **one** answer that **best** completes the sentence **[4 points]**

A **between-groups** experimental design:

A. means that each participant uses all of the systems being compared.
B. specifies what occurs between the time you test two different participants.
C. means that each participant uses only one of the systems being compared.
D. reduces the variability in the results.
E. is best for testing low-level interaction techniques.

(b) What type of users, performance, and tasks are GOMS analyses generally limited to? **[6 points]**

(c) What are **four** common dependent variables in an HCI experiment? **[8 points]**

Comprehensive Exam in Logic
November 2006

37 questions in 6 parts

Time: 1 hour

Instructions

- **Do not open the test booklet until instructed to do so.**
- The exam is open book and open notes, but no laptops or electronic accessories are allowed.
- Answer each question in the exam itself. The answers will fit in the given space. Writing outside of the given space will **not** be considered.
- It is strongly recommended that you work out the answer on scratch paper before answering.
- Unanswered questions are not penalized. Incorrect answers are penalized. Read the instructions before each section carefully.
- **THE HONOR CODE:**
  1. The honor code is an undertaking of the students individually and collectively:
     (a) that they will not give or receive aid in examinations; they will not give or receive unauthorized aid in class work, in the preparation of reports, or in any other work that is to be used by the instructors as the basis of grading;
     (b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the honor code.
  2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will avoid, as far as possible, academic procedures that create temptation to violate the Honor code.
  3. While the faculty alone have the right and the obligation to set academic requirements, the students and the faculty will work together to establish optimal conditions for honorable academic work.

  By writing my “magic–number” below, I acknowledge and accept the honor code.

WRITE MAGIC NUMBER: ____________________

I. Choose one answer to each of the following questions. F is a sentence (closed formula).
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **2**

1. If F is valid, then $\neg F$ is
   (*a*) satisfiable
   (*b*) unsatisfiable
   (*c*) valid
   (*d*) none of the above

2. If F is T-valid for some theory T, then F is necessarily __________ in FOL.
   (*a*) satisfiable
   (*b*) unsatisfiable
   (*c*) valid
   (*d*) none of the above

3. If F is T-satisfiable for some theory T, then F is necessarily __________ in FOL.
   (*a*) satisfiable
   (*b*) unsatisfiable
   (*c*) valid
   (*d*) none of the above

4. If F is T_1-valid and T_2-valid, then F is necessarily $(T_1 \cup T_2)$-valid.
   (*a*) True
   (*b*) False

5. If F is T-valid for complete theory T, then $\neg F$ is necessarily T-__________.
   (*a*) satisfiable
   (*b*) unsatisfiable
   (*c*) valid
   (*d*) none of the above

6. $\forall x.\ (3x = 2 \ \rightarrow\ x \leq 0)$ is __________ in the theory of integers, $T_{\mathbb{Z}}$, and __________ in the theory of rationals, $T_{\mathbb{Q}}$.
   (*a*) valid, valid
   (*b*) invalid, valid
   (*c*) valid, invalid
   (*d*) invalid, invalid

7. If $\neg F$ has a T-interpretation, then F is necessarily T-__________.
   (*a*) satisfiable
   (*b*) unsatisfiable
   (*c*) valid
   (*d*) none of the above

8. An inconsistent theory can be made consistent by adding more axioms.
   (*a*) True
   (*b*) False

II. Write a check next to the valid propositional logic formulae. Write a line through the invalid formulae.
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **1**

1. $(P \land Q) \rightarrow (P \rightarrow Q)$
2. $(P \rightarrow Q) \lor (P \land \neg Q)$
3. $(P \rightarrow (Q \rightarrow R)) \rightarrow (P \rightarrow R)$
4. $(P \rightarrow (Q \lor R)) \rightarrow (P \rightarrow R)$
5. $\neg(P \land Q) \rightarrow (R \rightarrow (\neg R \rightarrow Q))$
6. $(P \land Q) \lor \neg P \lor (\neg Q \rightarrow \neg P)$
7. $(P \rightarrow (Q \rightarrow R)) \rightarrow (\neg R \rightarrow (\neg Q \rightarrow \neg P))$
8. $(\neg R \rightarrow (\neg Q \rightarrow \neg P)) \rightarrow (P \rightarrow (Q \rightarrow R))$

III. Write a check next to the valid first-order logic formulae. Write a line through the invalid formulae.
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **1**

1. $(\forall x, y.\ (p(x, y) \rightarrow p(y, x))) \rightarrow \forall z.\ p(z, z)$
2. $\forall x, y.\ (p(x, y) \rightarrow (p(y, x) \rightarrow \forall z.\ p(z, z)))$
3. $p(a) \rightarrow \exists x.\ p(x)$
4. $(\exists x.\ p(x)) \rightarrow \forall y.\ p(y)$
5. $(\forall x.\ p(x)) \rightarrow \exists y.\ p(y)$

IV. Write a check next to the decidable first-order theories and fragments. Write a line through the undecidable ones. *QFF* abbreviates *quantifier-free fragment*.
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **1**

1. Propositional logic
2. Empty theory (all of first-order logic)
3. Theory of equality
4. QFF of theory of equality
5. First-order Peano arithmetic
6. QFF of first-order Peano arithmetic
7. Presburger arithmetic
8. QFF of Presburger arithmetic
9. Theory of reals with addition and multiplication (elementary algebra)
10. QFF of theory of elementary algebra
11. Theory of (possibly cyclic) LISP-like lists
12. QFF of theory of LISP-like lists
13. Theory of arrays
14. QFF of theory of arrays

V. Find a most general unifier for the following pair of terms, if one exists; otherwise, show where unification fails.
Incorrect or no answer: 0 *Correct answer*: 10 *Partial credit may be given.*

$$\langle p(g(y), x, f(g(y))),\ p(z, h(z, w), f(w)) \rangle$$

VI. Annotate the following function so that its annotations are inductive. The variable rv in the function postcondition holds the *return value.*
Incorrect or no answer: 0 *Correct answer*: 10 *Partial credit may be given.*

```
@pre ⊤
@post ∀i. 0 ≤ i < |rv| → rv[i] ≥ 0
int[] abs(int[] a₀) {
  int[] a := a₀;
  for
    @[                                              ]
    (int i := 0; i < |a|; i := i + 1) {
    if (a[i] < 0) {
      a[i] := − a[i];
    }
  }
  return a;
}
```

Comprehensive Exam in Logic
November 2006

37 questions in 6 parts

Time: 1 hour

Instructions

- **Do not open the test booklet until instructed to do so.**
- The exam is open book and open notes, but no laptops or electronic accessories are allowed.
- Answer each question in the exam itself. The answers will fit in the given space. Writing outside of the given space will **not** be considered.
- It is strongly recommended that you work out the answer on scratch paper before answering.
- Unanswered questions are not penalized. Incorrect answers are penalized. Read the instructions before each section carefully.
- **THE HONOR CODE:**
  1. The honor code is an undertaking of the students individually and collectively:
     (a) that they will not give or receive aid in examinations; they will not give or receive unauthorized aid in class work, in the preparation of reports, or in any other work that is to be used by the instructors as the basis of grading;
     (b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the honor code.
  2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will avoid, as far as possible, academic procedures that create temptation to violate the Honor code.
  3. While the faculty alone have the right and the obligation to set academic requirements, the students and the faculty will work together to establish optimal conditions for honorable academic work.

  By writing my “magic–number” below, I acknowledge and accept the honor code.

WRITE MAGIC NUMBER: ____________________

I. Choose one answer to each of the following questions. F is a sentence (closed formula).
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **2**

1. If F is valid, then $\neg F$ is
   (a) satisfiable
   (b) unsatisfiable (ANSWER)
   (c) valid
   (d) none of the above

2. If F is T-valid for some theory T, then F is necessarily ________ in FOL.
   (a) satisfiable
   (b) unsatisfiable
   (c) valid
   (d) none of the above (ANSWER)

3. If F is T-satisfiable for some theory T, then F is necessarily ________ in FOL.
   (a) satisfiable (ANSWER)
   (b) unsatisfiable
   (c) valid
   (d) none of the above

4. If F is T_1-valid and T_2-valid, then F is necessarily $(T_1 \cup T_2)$-valid.
   (a) True (ANSWER)
   (b) False

5. If F is T-valid for complete theory T, then $\neg F$ is necessarily T-________.
   (a) satisfiable
   (b) unsatisfiable (ANSWER)
   (c) valid
   (d) none of the above

6. $\forall x.\ (3x = 2 \ \rightarrow\ x \leq 0)$ is ________ in the theory of integers, $T_{\mathbb{Z}}$, and ________ in the theory of rationals, $T_{\mathbb{Q}}$.
   (a) valid, valid
   (b) invalid, valid
   (c) valid, invalid (ANSWER)
   (d) invalid, invalid

7. If $\neg F$ has a T-interpretation, then F is necessarily T-________.
   (a) satisfiable
   (b) unsatisfiable
   (c) valid
   (d) none of the above (ANSWER)

8. An inconsistent theory can be made consistent by adding more axioms.
   (a) True
   (b) False (ANSWER)

II. Write a check next to the valid propositional logic formulae. Write a line through the invalid formulae.
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **1**

1. $(P \wedge Q) \rightarrow (P \rightarrow Q)$ (CHECK)
2. $(P \rightarrow Q) \vee (P \wedge \neg Q)$ (CHECK)
3. $(P \rightarrow (Q \rightarrow R)) \rightarrow (P \rightarrow R)$
4. $(P \rightarrow (Q \vee R)) \rightarrow (P \rightarrow R)$
5. $\neg(P \wedge Q) \rightarrow (R \rightarrow (\neg R \rightarrow Q))$ (CHECK)
6. $(P \wedge Q) \vee \neg P \vee (\neg Q \rightarrow \neg P)$
7. $(P \rightarrow (Q \rightarrow R)) \rightarrow (\neg R \rightarrow (\neg Q \rightarrow \neg P))$
8. $(\neg R \rightarrow (\neg Q \rightarrow \neg P)) \rightarrow (P \rightarrow (Q \rightarrow R))$

III. Write a check next to the valid first-order logic formulae. Write a line through the invalid formulae.
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **1**

1. $(\forall x, y.\ (p(x,y) \rightarrow p(y,x))) \rightarrow \forall z.\ p(z,z)$
2. $\forall x, y.\ (p(x,y) \rightarrow (p(y,x) \rightarrow \forall z.\ p(z,z)))$
3. $p(a) \rightarrow \exists x.\ p(x)$ (CHECK)
4. $(\exists x.\ p(x)) \rightarrow \forall y.\ p(y)$
5. $(\forall x.\ p(x)) \rightarrow \exists y.\ p(y)$

IV. Write a check next to the decidable first-order theories and fragments. Write a line through the undecidable ones. *QFF* abbreviates *quantifier-free fragment*.
No answer: **0** *Incorrect answer*: **-1** *Correct answer*: **1**

1. Propositional logic (CHECK)
2. Empty theory (all of first-order logic)
3. Theory of equality
4. QFF of theory of equality (CHECK)
5. First-order Peano arithmetic
6. QFF of first-order Peano arithmetic
7. Presburger arithmetic (CHECK)
8. QFF of Presburger arithmetic (CHECK)
9. Theory of reals with addition and multiplication (elementary algebra) (CHECK)
10. QFF of theory of elementary algebra (CHECK)
11. Theory of (possibly cyclic) LISP-like lists
12. QFF of theory of LISP-like lists (CHECK)
13. Theory of arrays
14. QFF of theory of arrays (CHECK)

V. Find a most general unifier for the following pair of terms, if one exists; otherwise, show where unification fails.
Incorrect or no answer: 0 *Correct answer*: 10 *Partial credit may be given.*

$$\langle p(g(y), x, f(g(y))),\ p(z, h(z, w), f(w))\rangle$$

Solution:

$$\{z \mapsto g(y),\ x \mapsto h(g(y), w),\ w \mapsto g(y)\}$$

VI. Annotate the following function so that its annotations are inductive. The variable rv in the function postcondition holds the *return value.*
Incorrect or no answer: 0 *Correct answer*: 10 *Partial credit may be given.*

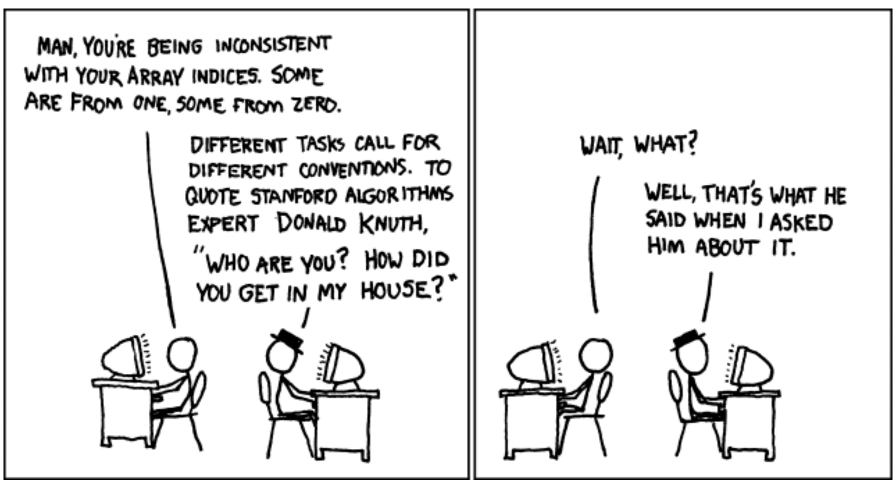
```
@pre ⊤
@post ∀i. 0 ≤ i < |rv| → rv[i] ≥ 0
int[] abs(int[] a₀) {
  int[] a := a₀;
  for
    @[                                        ]
    (int i := 0; i < |a|; i := i + 1) {
    if (a[i] < 0) {
      a[i] := − a[i];
    }
  }
  return a;
}
```

Solution: $\forall j.\ 0 \leq j < i \ \rightarrow\ a[j] \geq 0$

Stanford University
Computer Science Department

Fall 2006 Comprehensive Exam in Networking

You are allowed 1 hour to complete this exam.
(i) This exam is closed book, closed lap-top and closed notes.
(ii) Write your solution in your blue-book. Be sure to write your name and student ID clearly on the front of the book.
(iii) Don't panic! Be sure to start by reading the exam all the way through. Then answer the questions in whatever order you choose.
(iv) Show your reasoning clearly. If your reasoning is correct, but your final answer is wrong, you will receive most of the credit. If you just show the answer without reasoning, and your answer is wrong, you may receive no points at all.



The Stanford Honor Code
In accordance with both the letter and spirit of the Honor Code, I didn't cheat on this exam.

Signature:____

Enter your magic number here:____

Multiple Choice Questions.

Instructions: *in the following questions, check all listed assertions that appear to be correct. There is at least one correct assertion per question, but there may be more. Each correct assertion checked will earn you one point.* ***For each incorrect assertion you check, you will lose one point.*** *If you don't know an answer, checking no assertion will neither earn you nor lose you any points. Yep, you could end up with negative points on this section.*

1. **Layering.** "Layering" is commonly used in computer networks because:
   (a.) It forces all network software to be written in ANSI 'C'.
   (b.) Encapsulation is the lowest overhead method to transmit data.
   (c.) It allows widespread code and implementation re-use.
   (d.) It keeps networks warm enabling them to run faster.

2. **Elasticity Buffer.** An elasticity buffer is used to store bits arriving at a network interface. If the receiving station uses a 200-bit elasticity buffer and the clocks of the transmitter and receiver have a minimum frequency of 99.999MHz and a maximum frequency of 100.001MHz, which of the following statements are true:
   (a.) All packets have to be less than or equal to 12,500 bytes long.
   (b.) All packets have to be less than or equal to 4500 bytes long.
   (c.) The two clocks have a tolerance of +/- 100ppm.
   (d.) The two clocks have a tolerance of +/- 10ppm.
   (e.) The transmitter's clock is always faster than the receiver's clock.

3. **IP.** During normal IP packet forwarding by a router, which of the following packet fields are updated?
   (a.) IP header Source address
   (b.) IP header Destination address
   (c.) IP header TTL
   (d.) IP header checksum
   (e.) Destination UDP address
   (f.) Destination UDP port number

4. **TCP**. In the TCP protocol, the receiver advertises the current size of the receive window to the sender. The sender uses this information to control congestion on the network.
   (a.) True.
   (b.) False.

5. **IP addresses.** You are given the following network prefixes:

   171.64.179.0/24 171.64.180.0/24
   171.64.181.0/24 171.64.182.0/24
   171.64.183.0/24 171.64.184.0/24

   You are to aggregate the prefixes together into the smallest possible number of shorter prefixes. What is the smallest number of prefixes, and what are they?
   (a.) 1 prefix; 171.64.179.0/21
   (b.) 2 prefixes; 171.64.179.0/23, 171.64.182.0/23
   (c.) 2 prefixes; 171.64.179.0/24, 171.64.181.0/22
   (d.) 2 prefixes; 171.64.179.0/22, 171.64.183.0/24
   (e.) 3 prefixes 171.64.179.0/24, 171.64.180.0/22, 171.64.184.0/24

6. **TCP.** The TCP protocol uses a sliding window protocol. The window size varies because:
    (a.) Routers along the route advertise a varying window size to prevent congestion.
    (b.) The destination advertises a reduced window size when its buffers are congested.
    (c.) The destination advertises a reduced window size when packets take a long time to reach it.
    (d.) The source reduces its window size when it detects that congestion is occurring.

7. **Fair Queueing.** Which of the following are true:
    (a.) A fair queueing scheduler used in a router transmits one packet at a time.
    (b.) A fair queueing scheduler used in a router transmits just one bit from each packet at a time before moving onto the next packet.
    (c.) If traffic arriving at each router in a network is leaky-bucket constrained, and if each router uses weighted fair queueing schedulers, then bounds can be placed on the end-to-end delay of each packet.
    (d.) If a fair queueing scheduler calculates the finishing time of two packets, *A* and *B*, such that *A* is scheduled to depart before *B*, then at a later time as new packets arrive, the scheduler may change its mind and schedule *B* ahead of *A*.
    (e.) Weighted fair queueing allows a router to provide each flow with a weighted share of the link capacity.

8. **TCP**. TCP guarantees a reliable, in-order stream of data. Which of the following are true?
    (a.) TCP guarantees that if a byte did not reach the receiver, the sender will find out and be able to retransmit the data.
    (b.) TCP guarantees that any error that corrupts the TCP header is detected by the receiver
    (c.) TCP guarantees that any error that corrupts the TCP payload is detected by the receiver
    (d.) When a sender receives an acknowledgement with ACK sequence number A, it knows that all the data up to and including byte A-1 has been correctly received by the destination
    (e.) When a sender receives an acknowledgement with ACK sequence number A, it knows that all the data with a sequence number greater than A has not been received by the destination

9. **Link layer.** Which of the following are true?
    (a.) An Ethernet switch can interconnect a 10Mb/s Ethernet network and a 1Gb/s Ethernet.
    (b.) An Ethernet hub can interconnect a 10Mb/s Ethernet network and a 1Gb/s Ethernet.
    (c.) An Ethernet network cannot detect collisions until it has computed a checksum over the frame.
    (d.) 4B/5B is considered more efficient than Manchester encoding because more user data is transmitted in same amount of time.
    (e.) The 802.11b wireless protocol incorporates a link-layer ACK not present in regular Ethernet.

Longer questions

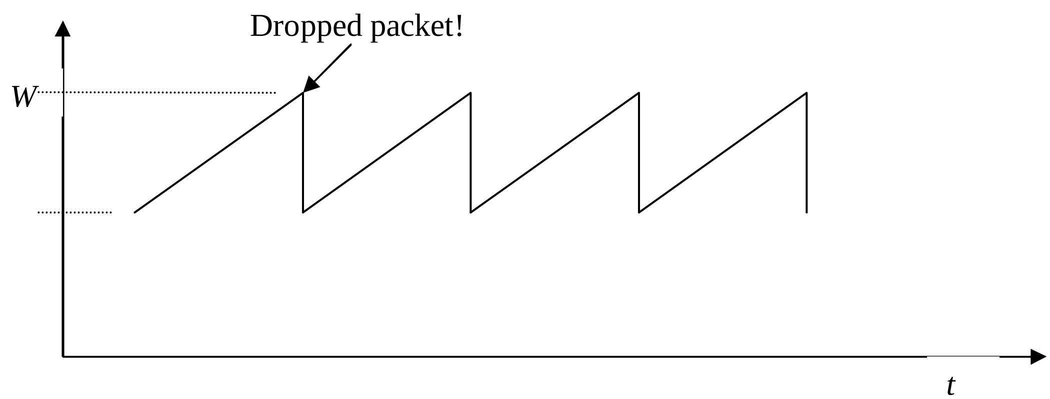
10. **(3 points) IP**. Explain why IPv4 fragments are reassembled at the end-point, rather than at an intermediate router along the path.

11. **(5 points) TCP sequence numbers**.
    (a.) Why does TCP use a 32-bit sequence number, instead of, say, 16 bits?

    (b.) Compute the maximum data rate at which a sender needs to worry about TCP sequence number wrap-around, assuming a maximum segment lifetime of two minutes.

12. **(20 points) TCP Congestion Control.** In this question we'll find approximate equations for the throughput of the TCP AIMD mechanism.
    (a.) Why does TCP use *additive* increase rather than *multiplicative* increase?

    The graph below shows the "sawtooth" evolution of the TCP sender's window size as a function of time. *W* is the maximum window size (measured in packets). In this question, assume that all packets are *P* bits long, and that exactly one packet is dropped every time the window size reaches *W*.



(b.) If we ignore the "slow-start" phase at the beginning of the flow, show that the average rate at which the transmitter sends packets is given by:

$$\overline{R} = \frac{3}{4}\frac{W}{RTT} \text{ packets per second.}$$

RTT is the round-trip time, which we will assume is constant.

(c.) Show that the fraction of packets dropped, *L*, is given by the following expression:

$$L = \frac{1}{\frac{3}{4}W\left(1+\frac{W}{2}\right)}$$

Hint: Remember that we're assuming that exactly one packet is dropped every time the window size reaches W.

(d.) Using your answers to (b) and (c), and assuming that is very large, show that the average rate at which the transmitter sends is given by:

$$\overline{R} \approx \frac{\frac{3}{4}\sqrt{\frac{8}{3}}}{\sqrt{L \cdot RTT}} P \approx \frac{1.22P}{\sqrt{L \cdot RTT}} \text{ bits per second.}$$

(e.) The *goodput* of a TCP connection is defined to be the rate at which data is sent once. i.e. the rate does not include data that is retransmitted. Is the goodput rate smaller or larger than *R*?

Throughout this question we assumed that *RTT* remains constant as the window size changes. We'll now see how this assumption is not only incorrect, but actually leads to a very different conclusion about rate *R*. Look again at the sawtooth figure. We say that the TCP flow is in equilibrium – the average rate available to the flow is constant; the only reason it follows a sawtooth is that TCP is varying the window size to see if the available rate has changed (which it hasn't). Each time the window size reaches *W*, the router at the bottleneck link drops a packet, which means the router's buffer is full. Assume that when the window size reaches the "trough" (i.e. bottom of the sawtooth) the router's buffer is empty (this is indeed the case if the buffer is sized accordingly, but you **don't** need to prove it here). Each time the source increases the window size from the trough to the peak the buffer will increase in size (and hence so will *RTT*) until the buffer overflows again.

(f.) How much does *RTT* increase each time the window size is increased? In this case, what is the actual average rate, *R*, at which the source sends?

13. **(20 points) Internet Design.** The Internet was originally designed to: (1) Quickly recover from link and router failure; (2) Efficiently use expensive long-haul links.

(a.) Explain what design choices were made to accomplish these two characteristics
(b.) Comment on whether you think the Internet accomplishes both goals, how it compares to alternative design choices, and whether you think these two design choices are relevant any more.

It was observed that a design choice was made – the so-called "End to End Principle".

(a.) Explain the end-to-end principle.
(b.) Give three examples that suggest the end-to-end principle still characterizes the Internet.
(c.) Give three examples that suggest it doesn't any more.

COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY
COMPREHENSIVE EXAMINATION IN NUMERICAL ANALYSIS
FALL 2006

1. **Newton's Method.** Suppose we want to compute the reciprocal of a number on a computer which does not have division.
   (a) Consider the equation
   $$f(x) = \frac{1}{x} - a, \qquad a \geq 0.$$
   Show how Newton's method can be used for computing $1/a$. What is your initial choice of x_0?
   (b) Discuss the convergence properties of the algorithm. Show the behavior of
   $$x_k - \frac{1}{a}.$$
   (c) Suppose A is an $n \times n$ matrix of rank n.
      (i) Show how Newton's method can be used for computing A^{-1}. What is the initial choice of X_0?
      (ii) How many operations are needed at each iteration?

2. **Interpolation.**
   (a) Let $p_n(x)$ and $q_n(x)$ be polynomials of degree n. Assume that
   $$p_n(x_i) = y_i, \qquad i = 0, 1, \ldots, n,$$
   $$q_n(x_i) = y_i, \qquad i = 0, 1, \ldots, n,$$
   and $x_1, \ldots, x_n$ are distinct. Show that $p_n(x) \equiv q_n(x)$.
   (b) Give a definition of a cubic spline. Explain the benefits of using a spline rather than using classical interpolation. What are the computational costs associated with a cubic spline?

3. **Differential equations.** Consider the differential equation
   $$y' = \lambda y$$
   with
   $$y(0) = 1.$$
   Assume that $\lambda < 0$. Construct the following numerical methods for solving this equation:

   Euler: $y_{k+1} = y_k + hy'_k$.
   Backward Euler: $y_{k+1} = y_k + hy'_{k+1}$.

   (a) Discuss the stability of each method as $k \to \infty$.
   (b) Give the advantages and disadvantages of using a Runge-Kutta method as opposed to a multistep method.

Comprehensive Exam: Programming Languages Autumn 2006

This is a 60-minute closed-book exam and the point total for all questions is 60.

All of the intended answers may be written within the space provided. (*Do not use a separate blue book.*) Succinct answers that do not include irrelevant observations are preferred. You may use the back of the preceding page for scratch work. If you to use the back side of a page to write part of your answer, be sure to mark your answer clearly.

The following is a statement of the Stanford University Honor Code:

A. *The Honor Code is an undertaking of the students, individually and collectively:*

   (1) *that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;*

   (2) *that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.*

B. *The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.*

C. *While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.*

By writing my "magic number" below, I certify that I acknowledge and accept the Honor Code.

____

(Number)

Prob	# 1	# 2	# 3	# 4	# 5	Total
Score						
Max	21	9	10	12	8	60

1. (*21 points*) .... Short Answer

Answer each question in a few words or phrases.

(a) (*3 points*) Why is memory usually separated into a stack and a heap?

(b) (*3 points*) Why is it possible to implement C without creating closures?

(c) (*3 points*) What is the difference between overloading and polymorphism?

(d) (*3 points*) Explain the difference between subtyping and inheritance.

(e) (*3 points*) If `A` <: `B` and `B` <: `C` is `A`→`B` <: `A`→`C` ? Why or why not?

(f) (*3 points*) Explain how the Java compiler treats this statement and, if the statement compiles, what happens at run time?

```
Character c = (Character) new Object();
```

(g) (*3 points*) What property of the Java architecture makes it necessary to do bytcode rewriting to get efficient method lookup?

2. (*9 points*) .... Pointer arithmetic

Unlike most programming languages, C allows pointer arithmetic. This question asks you about advantages and disadvantages of this language feature, with some parts of the question referring to the following excerpt from a JPEG decoder that performs a discrete cosine transformation using a loop that iterates through an array of elements.

```
void jpeg_fdct_ifast (DCTELEM *data)
{
    ...
    DCTELEM *dataptr;
    int ctr;
    ...
    /* Pass 1: process rows. */
    dataptr = data;
    for (ctr = DCTSIZE-1; ctr>= 0; ctr--) {
        tmp0 = dataptr[0] + ... + dataptr[DCTSIZE-1];
        ...
        dataptr += DCTSIZE;  /* advance pointer to next row */
    }
    ...
```

(a) (*2 points*) Name two disadvantages of pointer arithmetic. In answering the question, consider compiler features, desirable language properties, or language run-time features that are difficult or impossible in languages with pointer arithmetic.

(b) (*3 points*) One argument sometimes given in favor of pointer arithmetic is efficiency. In the code example above, each loop iteration processes one row of a DCTSIZE×DCTSIZE matrix whose entries are stored sequentially in memory. Iteration number `ctr` accesses the elements $\texttt{data}[\texttt{DCTSIZE} * ((\texttt{DCTSIZE} - 1) - \texttt{ctr})]$ to $\texttt{data}[\texttt{DCTSIZE} * ((\texttt{DCTSIZE} - 1) - \texttt{ctr}) + (\texttt{DCTSIZE} - 1)]$. Explain why this loop above might be more efficient than a loop that uses expressions of the form $\texttt{data}[\texttt{DCTSIZE} * ((\texttt{DCTSIZE} - 1) - \texttt{ctr}) + \texttt{i}]$ to index into the data array.

(c) (*2 points*) Suppose that the omitted code inside the loop accesses memory using $*$`dataptr` $+$ `j`, with `j` set by a `for` loop that only gives `j` values between 0 and `DCTSIZE` $-$ `1`. Explain why all memory accessed by the loop is memory that is allocated to the program, assuming that the function parameter `data` points to a heap-allocated region of `DCTSIZE` $*$ `DCTSIZE` locations.

(d) (*2 points*) Consider the more general setting of a pair of nested loops of the following form:

```
for (i = 0; i < imax; i++) {
   for (j = 0; j < jmax; j++) {
      ...
      ... data[i*A+j*B] ...
      ...
   }
}
```

i. (*1 points*) Assuming a simple-minded compiler with no loop-related optimizations, what arithmetic operation is used to compute the location `data[i*A+j*B]` in each execution of the inner loop ?

ii. (*1 points*) Suppose `A` and `B` are constants and `i` and `j` are only used inside array indices. What could an optimizing compiler do to simplify the arithmetic that is needed on each execution of the inner loop?

3. (*10 points*) .... **Scope and parameter passing**

(a) (*4 points*) Consider this simple program.

```
1 { int x;
2   int y;
3   int z;
4   x := 3;
5   y := 7;
6   { int f(int y) { return x*y };
7     int y;
8     y := 11;
9     { int g(int x) { return f(y) };
10       { int y;
11         y := 13;
12         z := g(2);
13       };
14     };
15   };
16 }
```

What value is assigned to z in line 12 under static scoping?

What value is assigned to z in line 12 under dynamic scoping?

(b) (*6 points*) What are the values of y and z at the end of the following block under the assumption that parameters are passed:

i. call by value
ii. call by reference
iii. call by value-result

```
{ int y;
  int z;
  y := 7;
  { int f(int x) {
       x := x+1;
       y := x;
       x := x+1;
       return y
     };
     int g(int x) {
        y := f(x)+1;
        x := f(y)+3;
        return x
     }
     z := g(y);
   };
}
```

4. (*12 points*) .... Dynamic Lookup

Answer each of the following questions about dynamic lookup in Smalltalk, C++, and Java in a few concise, carefully worded and legible sentences. Focus on the main points that distinguish these languages.

(a) (*3 points*) Briefly describe the C++ implementation of dynamic lookup, mentioning the key data structure(s).

(b) (*3 points*) Briefly describe the Smalltalk implementation dynamic lookup, mentioning the key data structure(s).

(c) (*2 points*) Which implementation would you expect to run faster? Why?

(d) (*2 points*) Could Smalltalk use the C++ implementation of dynamic lookup? Why or why or not?

(e) (*2 points*) How is Java similar or different from each of these implementations?

5. (*8 points*) .... Concurrency and Parallelism

Graphics processors (GPUs) are one example of a parallel processor that nearly everyone has on their desktop. The frame buffer is a region of memory on the GPU that stores pixel colors. In a single rendering pass, the frame buffer can only be written to and can not be read from. Conversely, the rest of the memory on the graphics card can be read from at any time but can not be written to. After a rendering pass, the frame buffer can be copied into another part of memory where the values can be read in a subsequent rendering pass. These restrictions are enforced by the device driver for the GPU.

(a) (*3 points*) What general problem with concurrent memory access do we solve by having a write-only frame buffer with all other memory being read-only?

(b) (*3 points*) Java concurrency primitives allow us to write a multi-threaded software simulation of the parallel processors in a GPU. Suppose we wanted to use our GPU simulator to see what happens when we change the frame buffer from having write-only access to having both read and write access.

Fill in the blanks in the following Java FrameBuffer class. You may need to write more than word per blank. Explain in one sentence.

```
class FrameBuffer {
  private FBdata[] pixels;
  ...

  public ____________________ read( int addr ) {
    return pixels[addr];
  }

  public ____________________ write( int addr, FBdata data ) {
    pixels[addr] = data;
  }

  ...
}
```

(c) (*2 points*) After running several experiments with your Java GPU simulation, you decide that allowing frame buffer reads would be a very useful feature. (This is actually true for some graphics algorithms.) There is a condition, restricting which threads get to write to each pixel, that would allow writes without introducing concurrency problems. Since a hardware implementation of pixel locking is too expensive to be feasible, what condition on access to pixels or groups of pixels could a device driver enforce to prevent concurrency problems? Explain.

Comprehensive Exam: Programming Languages Autumn 2006

1. (*21 points*) .... Short Answer

   Answer each question in a few words or phrases.

   (a) (*3 points*) Why is memory usually separated into a stack and a heap?

   Answer: Local data structures used only inside one block can be allocated/deallocated more efficiently using the stack discipline. When the lifetime of an object exceeds the lifetime of the block where it is created, some other mechanism (user deallocation or garbage collection) must be used to deallocated that space.

   (b) (*3 points*) Why is it possible to implement C without creating closures?

   Answer: No C functions are declared in nested scopes. If a function is passed to another function, then the lexical environment of the function that is passed is the global environment of the function it is passed to. Similarly, if a function is returned from another function, its lexical environment is the global environment, not a local environment of the function from which it is returned.

   (c) (*3 points*) What is the difference between overloading and polymorphism?

   Answer: Overloading allows one symbol to have more than one meaning. Overloading is resolved at compile time. Polymorphism allows a single function to be applied to many different types of arguments. Although a polymorphism function may conceptually have many different types, the name of a polymorphic function is associated with one algorithm, not many.

   (d) (*3 points*) Explain the difference between subtyping and inheritance.

   Answer: Subtyping is a relation between interfaces (or types), while inheritance is a relation between implementations.

   (e) (*3 points*) If `A` <: `B` and `B` <: `C` is `A`→`B` <: `A`→`C` ? Why or why not?

   Answer: Yes, by covariance of function types, `B` <: `C` implies `A`→`B` <: `A`→`C`. If f is a function of type `A`→`B`, then $f(x)$ has type `B`. Since `B` <: `C`, the return value $f(x)$ can appear where a value of type `C` is required. Therefore `f` can be used when a function of type `A`→`C` is needed.

   (f) (*3 points*) Explain how the Java compiler treats this statement and, if the statement compiles, what happens at run time?

   ```
   Character c = (Character) new Object();
   ```

   Answer: This statement compiles, because the cast makes it type-correct at compile time. However, a run-time error occurs when the statement is executed.

   (g) (*3 points*) What property of the Java architecture makes it necessary to do bytecode rewriting to get efficient method lookup?

   Answer: Run-time loading and linking of separately compiled class files. More specifically, a class file that calls a method on objects of class `C` may be compiled without knowing what class `C` will look like at run time. Therefore, the class file cannot be compiled to use fixed offsets of methods in the method table of class `C`. However, after `C` is loaded, the bytecode can be rewritten to use a fixed offset.

2. (*9 points*) .... Pointer arithmetic

   Unlike most programming languages, C allows pointer arithmetic. This question asks you about advantages and disadvantages of this language feature, with some parts of the question referring to the following excerpt from a JPEG decoder that performs a discrete cosine transformation using a loop that iterates through an array of elements.

```
void jpeg_fdct_ifast (DCTELEM *data)
{
   ...
   DCTELEM *dataptr;
   int ctr;
   ...
   /* Pass 1: process rows. */
   dataptr = data;
   for (ctr = DCTSIZE-1; ctr>= 0; ctr--) {
      tmp0 = dataptr[0] + ... + dataptr[DCTSIZE-1];
      ...
      dataptr += DCTSIZE;  /* advance pointer to next row */
   }
   ...
```

(a) (*2 points*) Name two disadvantages of pointer arithmetic. In answering the question, consider compiler features, desirable language properties, or language run-time features that are difficult or impossible in languages with pointer arithmetic.

Answer: Type safety, array bounds checking, garbage collection.

(b) (*3 points*) One argument sometimes given in favor of pointer arithmetic is efficiency. In the code example above, each loop iteration processes one row of a DCTSIZE×DCTSIZE matrix whose entries are stored sequentially in memory. Iteration number ctr accesses the elements $\texttt{data}[\texttt{DCTSIZE} * ((\texttt{DCTSIZE} - 1) - \texttt{ctr})]$ to $\texttt{data}[\texttt{DCTSIZE} * ((\texttt{DCTSIZE} - 1) - \texttt{ctr}) + (\texttt{DCTSIZE} - 1)]$. Explain why this loop above might be more efficient than a loop that uses expressions of the form $\texttt{data}[\texttt{DCTSIZE} * ((\texttt{DCTSIZE} - 1) - \texttt{ctr}) + \texttt{i}]$ to index into the data array.

Answer: No need for repeated addition and multiplication to compute the array index.

(c) (*2 points*) Suppose that the omitted code inside the loop accesses memory using $*(\texttt{dataptr} + \texttt{j})$, with j set by a **for** loop that only gives j values between 0 and $\texttt{DCTSIZE} - 1$. Explain why all memory accessed by the loop is memory that is allocated to the program, assuming that the function parameter data points to a heap-allocated region of $\texttt{DCTSIZE} * \texttt{DCTSIZE}$ locations.

Answer: A simple calculation shows that for $0 \leq \texttt{ctr} \leq \texttt{DCTSIZE} - 1$ and $0 \leq \texttt{j} \leq \texttt{DCTSIZE} - 1$, we have $\texttt{dataptr} = \texttt{data} + (\texttt{DCTSIZE} - 1 - \texttt{ctr}) * \texttt{DCTSIZE}$ and therefore

$$\begin{aligned} \texttt{data} &\leq \texttt{data} + (\texttt{DCTSIZE} - 1 - \texttt{ctr}) * \texttt{DCTSIZE} + \texttt{j} \\ &\leq \texttt{data} + (\texttt{DCTSIZE} - 1) * \texttt{DCTSIZE} + \texttt{DCTSIZE} - 1 \\ &= \texttt{data} + \texttt{DCTSIZE} * \texttt{DCTSIZE} - 1 \end{aligned}$$

(d) (*2 points*) Consider the more general setting of a pair of nested loops of the following form:

```
for (i = 0; i < imax; i++) {
   for (j = 0; j < jmax; j++) {
      ...
      ... data[i*A+j*B] ...
      ...
   }
}
```

i. (*1 points*) Assuming a simple-minded compiler with no loop-related optimizations, what arithmetic operation is used to compute the location `data[i*A+j*B]` in each execution of the inner loop ?
Answer: Two multiplications and an addition.

ii. (*1 points*) Suppose `A` and `B` are constants and `i` and `j` are only used inside array indices. What could an optimizing compiler do to simplify the arithmetic that is needed on each execution of the inner loop?
Answer: The product $\texttt{i} * \texttt{A}$ can be done outside the inner loop. In addition, the expression $\texttt{i} * \texttt{A} + \texttt{j} * \texttt{B}$ can be recognized as a linear function of the loop variable `j` and compiled using repeated addition of `j` on each loop iteration. This eliminates the multiplication $\texttt{j} * \texttt{B}$.

3. (*10 points*) .... Scope and parameter passing

(a) (*4 points*) Consider this simple program.

```
1 { int x;
2   int y;
3   int z;
4   x := 3;
5   y := 7;
6   { int f(int y) { return x*y };
7     int y;
8     y := 11;
9     { int g(int x) { return f(y) };
10       { int y;
11         y := 13;
12         z := g(2);
13       };
14     };
15   };
16 }
```

What value is assigned to `z` in line 12 under static scoping? **Answer:** 33

What value is assigned to `z` in line 12 under dynamic scoping? **Answer:** 26

(b) (*6 points*) What are the values of `y` and `z` at the end of the following block under the assumption that parameters are passed:

i. call by value **Answer:** y = 10, z = 13

ii. call by reference **Answer:** y = 15, z = 15

iii. call by value-result **Answer:** y = 13, z = 13

```
{ int y;
  int z;
  y := 7;
  { int f(int x) {
         x := x+1;
         y := x;
         x := x+1;
         return y
      };
      int g(int x) {
         y := f(x)+1;
         x := f(y)+3;
         return x
      }
      z := g(y);
    };
}
```

Answer: Here are traces for each case.

```
Value:        Reference:    Value-result (assignments in parens from copying-out):
g(7)          g(y)          g(y=7)
  f(7)          f(y)         f(x=7)
    x=8           y=8          x=8
    y=8           y=8          y=8
    x=9           y=9          x=9
    ret 8         ret 9        ret 8
  y=9           y=10         (x=9)
  f(9)          f(y)         y=9
    x=10          y=11       f(y=9)
    y=10          y=11         x=10
    x=11          y=12         y=10
    ret 10        ret 12       x=11
  x=13          y=15           ret 10
  ret 13        ret 15       (y=11)
z=13          z=15           x=13
                             ret 13
                            (y=13)
                            z=13
```

4. (*12 points*) .... Dynamic Lookup

Answer each of the following questions about dynamic lookup in Smalltalk, C++, and Java in a few concise, carefully worded and legible sentences. Focus on the main points that distinguish these languages.

(a) (*3 points*) Briefly describe the C++ implementation of dynamic lookup, mentioning the key data structure(s).

Answer: Uses indirection through the vtable of the class with an offset known at compile-time

(b) (*3 points*) Briefly describe the Smalltalk implementation dynamic lookup, mentioning the key data structure(s).

Answer: Method dictionary with a run-time search. If a method is not found in the method diction of a class, the superclass dictionary is searched next.

(c) (*2 points*) Which implementation would you expect to run faster? Why?

Answer: C++; the offset of the method in the table is known at compile-time, so there's no need to search the vtable at run-time

(d) (*2 points*) Could Smalltalk use the C++ implementation of dynamic lookup? Why or why or not?

Answer: No, smalltalk is not a statically typed language, so different classes might have the same method in different positions in the method dictionary. Therefore, a run-time search is needed.

(e) (*2 points*) How is Java similar or different from each of these implementations?

Answer: Java uses the Smalltalk-style search algorithm the first time a method is called from a specific line of the source program. If the class is known at the call site, then subsequent calls use the C++-style lookup.

5. (*8 points*) .... Concurrency and Parallelism

Graphics processors (GPUs) are one example of a parallel processor that nearly everyone has on their desktop. The frame buffer is a region of memory on the GPU that stores pixel colors. In a single rendering pass, the frame buffer can only be written to and can not be read from. Conversely, the rest of the memory on the graphics card can be read from at any time but can not be written to. After a rendering pass, the frame buffer can be copied into another part of memory where the values can be read in a subsequent rendering pass. These restrictions are enforced by the device driver for the GPU.

(a) (*3 points*) What general problem with concurrent memory access do we solve by having a write-only frame buffer with all other memory being read-only?

Answer: Some approach is needed to handle race conditions. These restrictions avoid race conditions without using locks. It also avoids reading pixels that have not been calculated and written yet.

(b) (*3 points*) Java concurrency primitives allow us to write a multi-threaded software simulation of the parallel processors in a GPU. Suppose we wanted to use our GPU simulator to see what happens when we change the frame buffer from having write-only access to having both read and write access.

Fill in the blanks in the following Java FrameBuffer class. You may need to write more than word per blank. Explain in one sentence.

```
class FrameBuffer {
  private FBdata[] pixels;
  ...

  public ____________________ read( int addr ) {
    return pixels[addr];
  }

  public ____________________ write( int addr, FBdata data ) {
    pixels[addr] = data;
  }

  ...
}
```

Answer: We'll need to synchronize both reads and writes because assignment is not atomic

(c) (*2 points*) After running several experiments with your Java GPU simulation, you decide that allowing frame buffer reads would be a very useful feature. (This is actually true for some graphics algorithms.) There is a condition, restricting which threads get to write to each pixel, that would allow writes without introducing concurrency problems. Since a hardware implementation of pixel locking is too expensive to be feasible, what condition on access to pixels or groups of pixels could a device driver enforce to prevent concurrency problems? Explain.

Answer: If a program can guarantee that the value read from the frame buffer is only being read by the same process/thread that writes the new value and nobody else (ie the computation that writes pixel[i,j] is the only one that can have read access to pixel[i,j]). This means the modification will not effect any other processing, and no locks are needed.

Comprehensive Exam — Systems software

Fall 2006

November 15, 2006

Answer each of the following questions, and give a sentence or two justification for your answer **(5 points each)**.

1. Initial versions of BSD disabled and re-enabled interrupts as the sole means of providing mutual exclusion. What must BSD do (and why) if a kernel thread running with interrupts disabled sleeps (i.e., puts itself on a block queue and allows another thread to run)? Give an example of a problem this can cause. (Note: you only need to reason about interrupts to answer this question, the details of BSD are irrelevant.)

2. You have three processes P1, P2, P3, where P1 has the highest priority, P2 the middle, and P3 the lowest. P3 acquires a lock `l`: give an example where this can interact arbitrarily badly with priorities and suggest a realistic fix.

3. Why can we trivially implement perfect LRU for files accessed using `read` and `write` system calls, but not for memory mapped files?

Answer each of the following questions, and give a sentence or two justification for your answer **(10 points each)**.

1. Consider very-broken code that uses the "double-check" lock idiom:

```
0:   int *p = 0;
....
1:   if(!p)
2:      lock(l);
3:      if(!p) {
4:         int *t = malloc(sizeof *t);
5:         *t = 3;
6:         p = t;
7:      }
8:      unlock(l);
9:   }
10:  x = x / *p;
```

   How can an optimizing compiler interact badly with the use of the `t` temporary? Assume you don't use an optimizing compiler: if another thread has done a `free` previously, what can happen if the initializing thread gets context switched immediately after line 6 and another thread executes line 10 on another processor?

2. Assuming you have a program that never frees memory running on an OS that provides a routine "`void *getpage(void)`" that allocates a page of virtual memory and returns a pointer to it. Explain how to write a "`malloc`" implementation guaranteed to never fragment more than a page of memory. Can you generally make the same guarantee if the program uses `free`? Give the intuition why or why not.

3. On a system with a TLB, what does the OS have to do after revoking a page from a process? Assuming a fixed page size, what feature of the TLB will place a hard upper bound on the amount of physical memory your system can use? You have a 64-entry, direct-mapped TLB, and 4K pages. Describe two repeating memory accessses that will produce horrible performance as compared to a 64-entry, 2-way set associative TLB.

4. Conservative OSes such as various BSD operating systems require that whenever a physical page is allocated for virtual memory that they can also reserve a page size chunk of swap space. Other OSes (such as Linux) use a page allocation policy called "over commit" where they find space in swap on demand, as processes need it. What is the advantage of overcommit? The price Linux pays for these benefits is that under high memory load it kills processes. Why does it have to do this?

5. A "perfectly consistent" file system would synchronously write all persistent data modified by a system call to disk before returning to the user. Most file systems do no such thing but instead defer writes as much as possible. Assume: you can detect when the user can externally observe the result of a program. Explain how you can modify the synchronous file system to be much more efficient while still behaving "as if" it was synchrnous.

6. Assume the common Unix file system interface, in particular, that you have a way to *non-atomically* write file data, that you have `sync()`, and that rename is atomic. Explain how to overwrite a file `A` with new contents such that any crash will result in A having either the old or new contents. Give the sequence of calls you would do for this.