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.

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	Total
Score					
Max	10	20	15	15	60

1. **[10 pts]** Prove a tight asymptotic bound on the behavior of $T(n) = 4T(\sqrt{n}) + \log n$, where $T(n) = \Theta(1)$ for $n < n_0$, where n_0 is a constants.

2. **[20 pts]** Your goal is to design a data structure for handling sets of positive integers (represented by arrays). It should support the following operations:

   - Given a set (represented by array of integers), add this set to the data structure. (e.g. ADD [1,7,2,13,55])
   - Given a set (represented by array of integers), detele this set from the data structure. For simplicity, assume that this set is legal, i.e. was added to the data structure at some point and was not yet deleted. (e.g. DELETE [7,550,78].)
   - Compute the size of the intersection of all of the sets in the data structure.

   Use n to denote the number of sets, k maximum number of elements in a set, and Q total number of distinct elements. For simplicity, assume that you know the values of Q, n, k in advance. Make sure that your memory usage is polynomial in the size of the input data.

   Make sure that dependence on Q is at most logarithmic. In other words, your data structure should be optimized for the case where Q and n are large, while k is relatively small.

   No need to write pseudo-code, just explain what techniques/elementary data structures you use and how you use them to implement each operation. You are allowed to use any data structures described in the reading list. In particular, it is ok to use arrays, lists, sorted lists, heaps, binary search trees, hashes, etc.

3. **[15 pts]** You are given a directed graph $G = (V, E)$ with n nodes and m edges, and two nodes s and t. Edges have non-negative weights. Given a path from s to t, redefine "length" of this path to be the sum of the lengths of the edges along the path *not counting the longest edge on this path.*
   Design an algorithm to compute shortest path from s to t using the modified definition of length. Faster algorithms will get more points. Make sure you state the asymptotic running time of your algorithm and provide proof that the algorithm is correct and that the running time is indeed as stated.

4. **[15 pts]** You are given an array of n integers from 0 to $n-1$, in sorted order starting from 0.

   - **[3 pts]** How much time will it take to convert this array into a heap ?
   - **[12 pts]** Once the array is converted into a heap, pick a random element and delete it from the heap, "fixing" the heap afterwards. What is the expected time of this operation ? Prove your answer. For simplicity, assume n is a power of 2.

Stanford University
Computer Science Department

Fall 2004 Comprehensive Exam in Artificial intelligence

1. **Closed book. Write only in the Blue Book provided.**
2. **The exam is timed for one hour.**
3. **Write your Magic Number on this sheet & on the Blue Book.**

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 un-permitted 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.

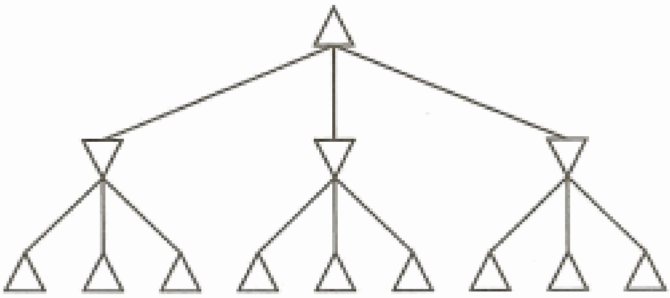
C. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

By writing my Magic Number below, I certify that I acknowledge and accept the Honor Code.

Magic Number---------------------------------

2004 Comprehensive Examination
Artificial Intelligence

1. **Search.** (20 points) Consider the game tree shown below. Upward-facing triangles are maximizing nodes, and downward-facing triangles are minimizing nodes.



(a) Assign the payoffs 1, 2, 3, 4, 5, 6, 7, 8, 9 to the terminal nodes of this tree in such a way that α-β pruning eliminates as *many* nodes as possible.
(b) Assign the payoffs 1, 2, 3, 4, 5, 6, 7, 8, 9 to the terminal nodes of this tree in such a way that α-β pruning eliminates as *few* nodes as possible.

2. Logic. (20 points) A *theory* is a set of sentences closed under logical entailment. A theory is *complete* for a language if and only if every sentence in the language *or* its negation is in the theory. Now, consider the language consisting of all sentences in First-Order Logic (1) that can be formed from just one unary relation constant p and two object constants a and b, (2) that includes variables x, y, z, ..., but (3) that does *not* include functions, explicit quantifiers, or equality. Which of the following sentences logically entails a theory that is complete for all sentences in this language?

(a) $p(a) \wedge p(b)$
(b) $p(a) \wedge \neg p(b)$
(c) $p(x)$ (equivalent to $\forall x.p(x)$)
(d) $p(a) \vee \neg p(a)$
(e) $p(a) \wedge \neg p(a)$

3. Automated Reasoning. (20 points) Use the resolution refutation method to prove $\exists x.\exists y.(p(x,y) \wedge q(x,y))$ from the following premises.

$$\exists x.\forall y. (p(x,y) \Leftrightarrow q(x,y))$$
$$\forall x.(\exists y.p(x,y) \vee \exists z.q(x,z))$$

Note that this is a question about the resolution refutation method. You will get zero points, nothing, nada, zip, no score for proving it in any other way.

4. Probability. (20 points) Adapted from Nilsson's *Artificial Intelligence: A New Synthesis*. Suppose that colored balls are distributed in three indistinguishable boxes, B1, B2, and B3, as shown in the following table.

	B1	B2	B3
Red	2	4	3
White	3	2	4
Blue	6	3	3

A box is selected at random from which a ball is selected at random. The ball is red. What is the probability of the box selected being B1? Unreduced fractions are okay.

5. Natural Language. (20 points) Consider the augmented phrase structure grammar shown below.

$S(r(x, z) \wedge r(y, z)) \rightarrow Q(r(both(x,y), z))$
$Q(w(u, v)) \rightarrow NP(u)\ Verb(w)\ NP(v)$
$NP(x) \rightarrow Noun(x)$
$NP(both(x, y)) \rightarrow NP(x)$ ***and*** $NP(y)$
$Noun(tom) \rightarrow$ ***Art***
$Noun(dick) \rightarrow$ ***Bob***
$Noun(harry) \rightarrow$ ***Cal***
$Noun(mary) \rightarrow$ ***Deb***
$Verb(hates) \rightarrow$ ***hate***
$Verb(hates) \rightarrow$ ***hates***

(a) Given that *s* is the top-level non-terminal, is there a semantic interpretation for the expression ***Deb hates Art and Bob***? If so, what is it?

(b) Given that *s* is the top-level non-terminal, is there a semantic interpretation for the expression ***Art and Cal hate Deb***? If so, what is it?

(c) Change the augmentations on the existing rules to eliminate ungrammatical sentences like *Art and Cal hates Deb* (without eliminating the corresponding grammatical sentences). If you are unable to do this, you can still get partial credit by changing the rules themselves.

2004 Comprehensive Examination
Artificial Intelligence

1. Search.

There are several ways this can be done. The following are examples.
(a) 9, 8, 7, 6, 5, 4, 3, 2, 1
(b) 1, 2, 3, 4, 5, 6, 7, 8, 9

2. Logic

(a) Incomplete
(b) Incomplete
(c) Complete
(d) Incomplete
(e) Complete

3. Automated Reasoning.

1. $\{-p(a,y), q(a,y)\}$
2. $\{p(a,y), -q(a,y)\}$
3. $\{p(x,f(x)), q(x,g(x))\}$
4. $\{-p(x,y), -q(x,y)\}$

5. $\{-q(a,y)\}$
6. $\{-p(a,y)\}$
7. $\{q(a,g(a))\}$
8. $\{\}$

4. Probability.

$p(\text{B1}|\text{Red}) = p(\text{Red} \mid \text{B1}) * p(\text{B1}) / p(\text{Red})$
$=2/11*1/3 \;/\; (2/11+4/9+3/10)*1/3$
$=0.1963$

5. Natural Language.

(a) There is no semantic interpretation in this case.

(b) *hates*(*tom*,*mary*) & *hates*(*harry*,*mary*)

(c) One way is to add a number parameter, as shown below.

$S(r(x, z) \wedge r(y, z)) \rightarrow Q(r(both(x,y), z))$
$Q(w(u, v)) \rightarrow NP(u,n)\ Verb(w,n)\ NP(v)$
$NP(x,\mathrm{s}) \rightarrow Noun(x)$
$NP(both(x, y),p) \rightarrow NP(x,z)$ ***and*** $NP(y,z)$
$Noun(tom) \rightarrow$ ***Art***
$Noun(dick) \rightarrow$ ***Bob***
$Noun(harry) \rightarrow$ ***Cal***
$Noun(mary) \rightarrow$ ***Deb***
$Verb(hates,p) \rightarrow$ ***hate***
$Verb(hates,s) \rightarrow$ ***hates***

It is also possible to accomplish this by splitting the rules for *Q*, *NP*, and *Verb*; but this can be more cumbersome.

Automata and Formal Languages Comprehensive Exam

Fall 2004

Problem 1 (10 points)

Give context-free grammars generating the following languages over the alphabet $\{0, 1\}$ (you need not prove the correctness of your grammars):

(a) $\{a^i b^j a^{i+j+k} b^k : i, j, k \geq 0\}$;

(b) all strings with an equal number of a's and b's.

Problem 2 (15 points)

Decide whether the following statements are TRUE or FALSE. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) If L_1 and L_2 are both non-regular, then $L_1 \cap L_2$ must be non-regular.

(b) Suppose there is a polynomial-time reduction from the language L_1 to the language L_2. It is possible that L_1 is solvable in polynomial time but L_2 is not even in NP.

(c) Suppose there is a polynomial-time reduction from the language L_1 to the language L_2. If L_1 is recursive, then L_2 must be recursive.

(d) Every infinite regular set contains a subset that is not recursively enumerable.

(e) Every infinite recursively enumerable set contains an infinite subset that is recursive.

Problem 3 (15 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.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA could be empty or finite, and must always be context-free, but the smallest class that contains all such languages is that of the *regular* languages. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) The intersection of a context-free language and a regular language.

(b) The intersection of a recursive language and a regular language.

(c) The languages accepted by nondeterministic pushdown automata with a single state that accept by empty stack.

(d) The languages accepted by nondeterministic pushdown automata with two stacks.

(e) The complement of a language in NP.

Problem 4 (15 points)

Specify which of the following problems are *decidable* and which are *undecidable*. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) Given a Turing machine M, does M halt when started with an empty tape?

(b) Given a context-free language L and a regular language R, is $L \subseteq R$?

(c) Given a context-free language L and a regular language R, is $R \subseteq L$?

(d) Given a DFA, does it accept on only finitely many inputs?

(e) Given a PDA, does it accept on only finitely many inputs?

Problem 5 (15 points)

A *monotone* 2-SAT formula is a 2-CNF Boolean formula $F(x_1, \ldots, x_n)$ that does not contain negated variables. For example:

$$F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_3).$$

It is clear that there always exists a truth assignment for the variables $x_1, \ldots, x_n$ satisfying the formula F—simply set each variable to TRUE.

Consider the following problem called MONOTONE 2-SAT: given a monotone 2-SAT formula F and a positive integer k, determine whether there exists a truth assignment satisfying F such that the number of variables set to TRUE is *at most* k.

Prove that the MONOTONE 2-SAT problem is NP-complete. (**Hint:** Think about the NP-complete VERTEX COVER problem.)

Automata and Formal Languages Comprehensive Exam

Fall 2004

Problem 1 (10 points)

Give context-free grammars generating the following languages over the alphabet $\{0, 1\}$ (you need not prove the correctness of your grammars):

(a) $\{a^i b^j a^{i+j+k} b^k \ : \ i, j, k \geq 0\}$;

(b) all strings with an equal number of a's and b's.

Solution:

(a)

$$\begin{aligned} S &\rightarrow AC && (1) \\ A &\rightarrow aAa && (2) \\ A &\rightarrow B && (3) \\ B &\rightarrow bBa && (4) \\ B &\rightarrow \epsilon && (5) \\ C &\rightarrow aCb && (6) \\ C &\rightarrow \epsilon && (7) \end{aligned}$$

(b)

$$\begin{aligned} S &\rightarrow aSbS && (8) \\ S &\rightarrow bSaS && (9) \\ S &\rightarrow \epsilon && (10) \end{aligned}$$

Problem 2 (15 points)

Decide whether the following statements are TRUE or FALSE. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) If L_1 and L_2 are both non-regular, then $L_1 \cap L_2$ must be non-regular.

(b) Suppose there is a polynomial-time reduction from the language L_1 to the language L_2. It is possible that L_1 is solvable in polynomial time but L_2 is not even in NP.

(c) Suppose there is a polynomial-time reduction from the language L_1 to the language L_2. If L_1 is recursive, then L_2 must be recursive.

(d) Every infinite regular set contains a subset that is not recursively enumerable.

(e) Every infinite recursively enumerable set contains an infinite subset that is recursive.

Solution:

(a) FALSE

(b) TRUE

(c) FALSE

(d) TRUE

(e) TRUE

Problem 3 (15 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*. You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA could be empty or finite, and must always be context-free, but the smallest class that contains all such languages is that of the *regular* languages. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) The intersection of a context-free language and a regular language.

(b) The intersection of a recursive language and a regular language.

(c) The languages accepted by nondeterministic pushdown automata with a single state that accept by empty stack.

(d) The languages accepted by nondeterministic pushdown automata with two stacks.

(e) The complement of a language in NP.

Solution:

(a) Context-free

(b) Recursive

(c) Context-free

(d) Recursively enumerable

(e) Recursive

Problem 4 (15 points)

Specify which of the following problems are *decidable* and which are *undecidable*. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) Given a Turing machine M, does M halt when started with an empty tape?

(b) Given a context-free language L and a regular language R, is $L \subseteq R$?

(c) Given a context-free language L and a regular language R, is $R \subseteq L$?

(d) Given a DFA, does it accept on only finitely many inputs?

(e) Given a PDA, does it accept on only finitely many inputs?

Solution:

(a) Undecidable

(b) Decidable

(c) Undecidable

(d) Decidable

(e) Decidable

Problem 5 (15 points)

A *monotone* 2-SAT formula is a 2-CNF Boolean formula $F(x_1, \ldots, x_n)$ that does not contain negated variables. For example:

$$F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_3).$$

It is clear that there always exists a truth assignment for the variables $x_1, \ldots, x_n$ satisfying the formula F—simply set each variable to TRUE.

Consider the following problem called MONOTONE 2-SAT: given a monotone 2-SAT formula F and a positive integer k, determine whether there exists a truth assignment satisfying F such that the number of variables set to TRUE is *at most* k.

Prove that the MONOTONE 2-SAT problem is NP-complete. (**Hint:** Think about the NP-complete VERTEX COVER problem.)

Solution: Recall that in a graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$, a *vertex cover* is a set $C \subseteq V$ of vertices such that for each edge $(i, j) \in E$, at least one of its endpoints is in C: $\{i, j\} \cap C \neq \emptyset$. The VERTEX COVER problem is the following: given a graph $G = (V, E)$ and a positive integer k, does G contain a vertex cover of size at most k? We know that VC is NP-hard, and establish NP-hardness of MONOTONE 2-SAT via a polynomial-time reduction from VC.

The reduction starts with a VC instance $< G, k >$ and creates an instance $< F, k >$ of MONOTONE 2-SAT, where the monotone 2-CNF formula F is defined as follows: for each vertex $i \in V$, create a Boolean variable x_i; for each edge $(i, j) \in E$, create a clause $x_i \vee x_j$. The reduction runs in linear time, but it remains to verify its correctness.

Suppose G has a vertex cover C of size at most k. Consider the truth assignment for the variables in F in which $x_i =$ TRUE if and only if $i \in C$; clearly, the number of TRUE variables is at most k. We claim that this is a satisfying truth assignment for F. To establish the claim, consider an arbitrary clause $x_i \vee x_j$ of F. Since (i, j) must be an edge of G, and hence C must contain at least one of i and j, it follows that at least one of x_i and x_j is assigned TRUE and hence the clause is satisfied.

Suppose now that there is a satisfying truth assignment for F with no more than k variables set to TRUE. Consider the set of vertices $C = \{i : x_i = \text{TRUE}\}$; clearly, $|C| \leq k$. We claim that C is a vertex cover for G. To see this, focus on any one edge $(i, j) \in E$. Since F must have a clause $x_i \vee x_j$, and that clause is satisfied, at least one of x_i and x_j is assigned TRUE and so at least one end-point of the edge (i, j) belongs to C.

Finally, MONOTONE 2-SAT is in NP because the feasibility of a candidate solution (i.e., a truth assignment) can be checked in polynomial time.

Compilers Comprehensive Exam
Fall 2004

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

1. **Regular Expressions** (5 points)

   Consider a language where real constants are defined as follows: A real constant contains a decimal point or E notation, or both. For instance, 0.01, 2.71821828, ˜1.2E12, and 7E˜5 are real constants. The symbol "˜" denotes unary minus and may appear before the number or on the exponent. There is *no* unary "+" operation. There must be at least one digit to left of the decimal point, but there might be no digits to the right of the decimal point. The exponent following the "E" is a (possibly negative) integer.

   Write a regular expression for such real constants. Use the standard regular expression notation described by the following grammar:

   $$R \rightarrow \epsilon \,|\, \text{char} \,|\, R + R \,|\, R* \,|\, RR \,|\, (R)$$

   You may define names for regular expressions you want to use in more than one place (e.g., $foo = R$).

2. **Grammars** (20 points)

   Consider the following grammar. The nonterminals are E, T, and L. The terminals are +,**id**,(,), and ;. The start symbol is E.

$$
\begin{array}{rcl}
\mathrm{E} & \rightarrow & \mathrm{E} + \mathrm{T} \mid \mathrm{T} \\
\mathrm{T} & \rightarrow & \mathbf{id} \mid \mathbf{id}() \mid \mathbf{id}(\mathrm{L}) \\
\mathrm{L} & \rightarrow & \mathrm{E}; \mathrm{L} \mid \mathrm{E}
\end{array}
$$

   Give an LL(1) grammar that generates the same language as this grammar. For full credit, you must show (convincingly) that your grammar is LL(1).

3. **Parsing** (15 points)

Consider the following grammar. The nonterminals are S′ and S. The terminals are **op** and **x**. The start symbol is S′.

$$
\begin{aligned}
\mathrm{S}' &\rightarrow \mathrm{S} \\
\mathrm{S} &\rightarrow \mathrm{S}\ \mathbf{op}\ \mathrm{S} \mid \mathbf{x}
\end{aligned}
$$

(a) Draw the DFA built from sets of LR(0) items for this grammar. Show the contents of each state. (Note: Don't augment the grammar with a new start symbol.)

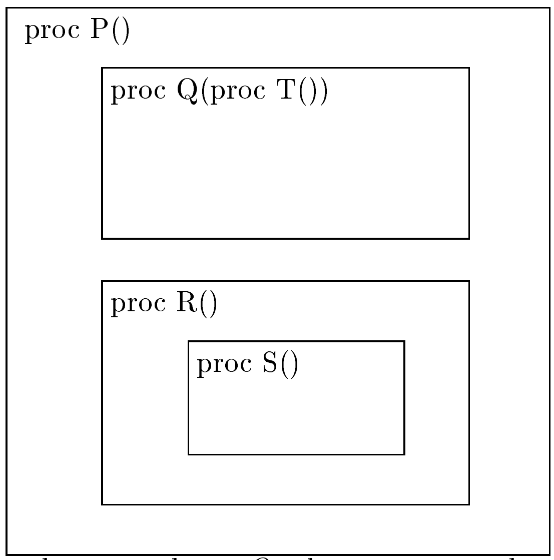
(b) Is this grammar LR(1)? Briefly explain why or why not.

4. **Scope** (10 points)

   Give a simple program that produces different results if executed using lexical scoping than if executed using dynamic scoping. Show what your program produces in both cases. Use any reasonable and clear programming notation.

5. **Activation Records** (10 points)

Consider a program with the following lexical structure. The program is written in a lexically scoped language with nested procedures (like Pascal):



$P, R,$ and S are parameterless procedures; Q takes a parameterless procedure T as a parameter. Suppose that at run-time the following sequence of calls is made:

P is called from some lexically-enclosing main program
P calls R
R calls S
S calls P
P calls R
R calls Q with S as a parameter
Q calls T
T calls S

Draw the stack of activation records present after this sequence of calls. You don't need to show the entire contents of the activation record—for each indicate only the name of the procedure being activated, the control (dynamic) link for that activation record, and the access (static) link for that activation record.

Compilers Comprehensive Exam
Fall 2004

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

1. **Regular Expressions** (5 points)

   Consider a language where real constants are defined as follows: A real constant contains a decimal point or E notation, or both. For instance, 0.01, 2.71821828, ˜1.2E12, and 7E˜5 are real constants. The symbol "˜" denotes unary minus and may appear before the number or on the exponent. There is *no* unary "+" operation. There must be at least one digit to left of the decimal point, but there might be no digits to the right of the decimal point. The exponent following the "E" is a (possibly negative) integer.

   Write a regular expression for such real constants. Use the standard regular expression notation described by the following grammar:

$$R \rightarrow \epsilon \mid \text{char} \mid R + R \mid R* \mid RR \mid (R)$$

   You may define names for regular expressions you want to use in more than one place (e.g., $foo = R$).

$$
\begin{aligned}
\texttt{digit} &= \texttt{0} + \texttt{1} + \texttt{2} + \texttt{3} + \texttt{4} + \texttt{5} + \texttt{6} + \texttt{7} + \texttt{8} + \texttt{9} \\
\texttt{posint} &= \texttt{digit}\ \texttt{digit}^* \\
\texttt{int} &= (\epsilon + \texttt{\~{}})\ \texttt{posint} \\
\texttt{exp} &= \texttt{E}\ \texttt{int} \\
\texttt{frac} &= \texttt{.}\ \texttt{digit}^* \\
\texttt{real} &= (\texttt{int}\ \texttt{frac}\ (\texttt{exp} + \epsilon)) + (\texttt{int}\ (\texttt{frac} + \epsilon)\ \texttt{exp})
\end{aligned}
$$

2. **Grammars** (20 points)

Consider the following grammar. The nonterminals are E, T, and L. The terminals are +,**id**,(,), and ;. The start symbol is E.

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow \textbf{id} \mid \textbf{id}() \mid \textbf{id}(L) \\ L &\rightarrow E; L \mid E \end{aligned}$$

Give an LL(1) grammar that generates the same language as this grammar. For full credit, you must show (convincingly) that your grammar is LL(1).

(a) Eliminate left recursion:

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \epsilon \\ T &\rightarrow \texttt{id} \mid \texttt{id}() \mid \texttt{id}(L) \\ L &\rightarrow E; L \mid E \end{aligned}$$

(b) Left factor:

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \epsilon \\ T &\rightarrow \texttt{id}\, T' \\ T' &\rightarrow \epsilon \mid (T'' \\ T'' &\rightarrow) \mid L) \\ L &\rightarrow EL' \\ L' &\rightarrow ;L \mid \epsilon \end{aligned}$$

(c) Check that it's LL(1). For this part you just needed to give enough information to show that there would be no conflicts in the parsing table. The following is sufficient:

$$
\begin{array}{rclrcl}
\mathtt{E}' & \rightarrow & +\mathtt{TE}' \,|\, \epsilon & \mathtt{First}(+\mathtt{TE}') & = & \{+\} \\
 & & & \mathtt{Follow}(\mathtt{E}') & = & \{\$,), ;\} \\
\mathtt{T}' & \rightarrow & \epsilon \,|\, (\mathtt{T}'' & \mathtt{First}((\mathtt{T}'') & = & \{(\} \\
 & & & \mathtt{Follow}(\mathtt{T}') & = & \{+, \$,), ;\} \\
\mathtt{T}'' & \rightarrow &) \,|\, \mathtt{L}) & \mathtt{First}()) & = & \{)\} \\
 & & & \mathtt{First}(\mathtt{L})) & = & \{\mathtt{id}\} \\
\mathtt{L}' & \rightarrow & ;\mathtt{L} \,|\, \epsilon & \mathtt{First}(;\mathtt{L}) & = & \{;\} \\
 & & & \mathtt{Follow}(\mathtt{L}') & = & \{)\}
\end{array}
$$

3. **Parsing** (15 points)

Consider the following grammar. The nonterminals are S′ and S. The terminals are **op** and **x**. The start symbol is S′.

$$
\begin{aligned}
S' &\rightarrow S \\
S &\rightarrow S\ \mathbf{op}\ S \mid \mathbf{x}
\end{aligned}
$$

(a) Draw the DFA built from sets of LR(0) items for this grammar. Show the contents of each state. (Note: Don't augment the grammar with a new start symbol.)

```
                  -----------------
                  | S' -> . S     |
                  | S -> . S op S |
                  | S -> . x      |
                  -----------------
          S       |            |     x
     ---------------           ------------
     V                                    V
-----------------                     ------------
| S' -> S .     | **                  | S -> x . |
| S -> S . op S |                     ------------
-----------------                           ^
     |           op           x             |
     |---------------      ---------------
                 V       |
                  -----------------
                  | S -> S op . S |
                  | S -> . S op S |
                  | S -> . x      |
                  -----------------
```

(b) Is this grammar LR(1)? Briefly explain why or why not.

`No.  The grammar is ambiguous.`

4. **Scope** (10 points)

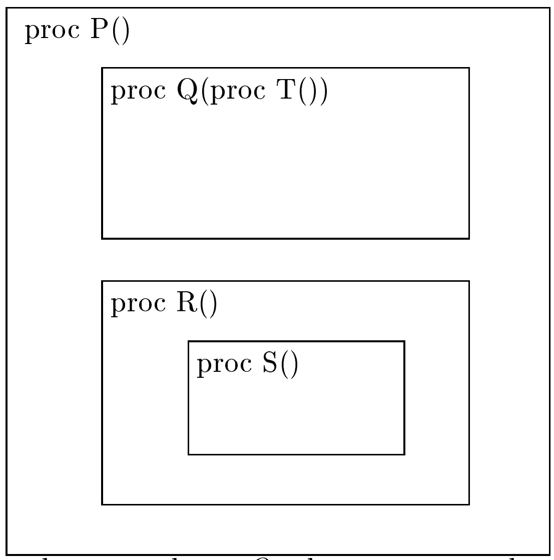
Give a simple program that produces different results if executed using lexical scoping than if executed using dynamic scoping. Show what your program produces in both cases. Use any reasonable and clear programming notation.

```
main()
    var x;

    proc p1
        var x;
        x := 1;
        p2();
    end;

    proc p2
        print(x);
    end;

x := 0;
p1();
end;
```

`This program prints 0 with lexical scope and 1 with dynamic scope.`

5. **Activation Records** (10 points)

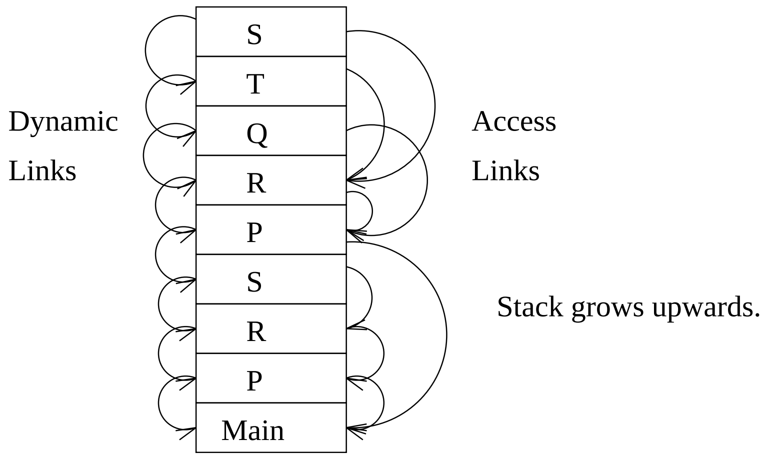
Consider a program with the following lexical structure. The program is written in a lexically scoped language with nested procedures (like Pascal):



$P, R,$ and S are parameterless procedures; Q takes a parameterless procedure T as a parameter. Suppose that at run-time the following sequence of calls is made:

P is called from some lexically-enclosing main program
P calls R
R calls S
S calls P
P calls R
R calls Q with S as a parameter
Q calls T
T calls S

Draw the stack of activation records present after this sequence of calls. You don't need to show the entire contents of the activation record—for each indicate only the name of the procedure being activated, the control (dynamic) link for that activation record, and the access (static) link for that activation record.


Dynamic
Links
S
T
Q
R
P
S
R
P
Main
Access
Links
Stack grows upwards.

Computer Science Comprehensive Examination
Computer Architecture
[60 points]

This examination is open book. Please do all of your work on these sheets. Do not do your work in a blue book.

Number: ____

Problem	*Max Score*	*Your Score*
1	20	
2	20	
3	20	
TOTAL	60	

Problem 1: Short Answer [20 points]

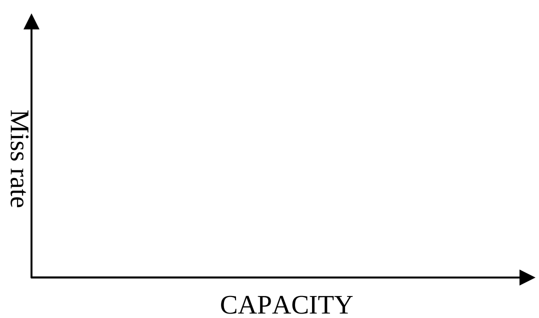
A. [7 points] Assume a pipelined processor with N pipeline stages. As we increase N, briefly explain that happens to the following (increase/decrease + 1 sentence reasoning)

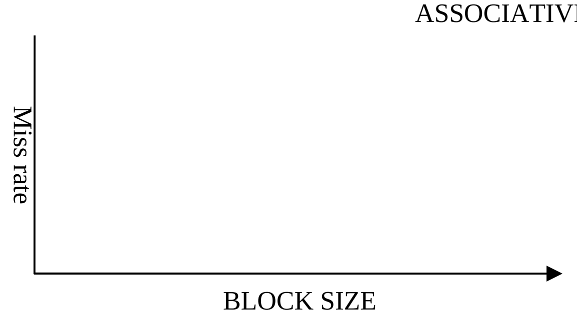
- Clock cycle time:

- Cycles per instruction:

Considering clock cycle time only, is there a limit or point of diminishing returns for the number of pipeline stages in a processor? Why?

B. [7 points] Assume a first level data cache with capacity (C), associativity (A), and block size (B). Draw the expected effect on hit rate as C, A, and B increase respectively. In each case, we vary one of the three parameters, while the other two are kept constant. Mark one or two interesting points on the graphs with 3-4 words to provide some reasoning.





C. [6 points] What does the following MIPS code do when it is invoked with the register $a0 containing the value 0x2000? What is the instruction cache miss rate? What is the data cache miss rate? Assume the caches are initially empty, allocate lines on both load and store misses, and have a 16 byte line size.

```
0x1000:                move $t0, $a0
0x1004:                addi $t1, $t0, 4000
0x1008:     loop:      sw   $t0, 0($t0)
0x100C:                addi $t0, $t0, 4
0x1010:                slt  $t2, $t0, $t1
0x1014:                bne  $t2, $zero, loop
0x1018:     exit:
```

Problem 2: Instruction-Set Architecture Design [20 Points]

Your task is to redesign the way branches work in the MIPS ISA to create a new architecture called MIPS-new. You decide to add branches with arbitrary compares, which are called complex branches. You are given the following information about MIPS and MIPS-new:

Instruction mix on MIPS:

Instruction type	*Frequency*
compares	23%
ALU ops (not compares)	20%
loads/stores	30%
conditional branches	25%
jumps	2%

Frequency of branch compares on MIPS-new:

Branch type	*Frequency*
NE/EQ compare to nonzero	10%
NE/EQ compare to zero	5%
LT/LE compare to nonzero	30%
GT/GE compare to zero	5%
LT/LE compare to nonzero	45%
GT/GE compare to zero	5%

A. [9 points] Using the information given above, what is the ratio of instruction count (IC) of MIPS to MIPS-new. You may assume that the compiler transforms a compare-branch instruction pair into a complex branch whenever possible.

B. [9 points] Complex branches in MIPS-new force you to move the branch decision point from the ID stage to the EX stage. To maintain compatibility with MIPS, you decide to use delayed branching for the first cycle of branch delay and to predict taken for any subsequent cycles of branch delay. Assume that the target PC is calculated in the ID stage. If the delay slot is usefully filled 50% of the time and branches are taken 70% of the time, what is the ratio of CPIs of MIPS to MIPS-new? Assume that noops do not count as instructions and all instructions except branches and jumps have a CPI of 1.0.

C. [2 points] Which architecture is better? Why?

Problem 3: Memory Hierarchy Design [20 Points]

You have a computer with two levels of cache memory and the following specifications:

- Processor: 2GHz, 64-bit RISC CPU
- On-chip L1 caches
  - split instruction & data cache, blocking, single-ported
  - write-through & non-write allocate
  - 1 CPU cycle access time (i.e. hits do not stall the processor)
  - Block size = 32 bytes
- Off-chip L2 cache
  - unified single-ported cache, blocking
  - write-back
  - 10 CPU cycles access time (L1 miss penalty) for both reads and writes
  - Block size = 32 bytes
- Main memory:
  - Bus: 64-bit data transfers at 400 MHz
  - Latency: 15+5+5+5 CPU cycles access time for 32 bytes (L2 miss penalty – includes latency of both DRAM and memory bus)

A. What is the:

- Peak L1 data cache bandwidth available to CPU (assuming 0% L1 misses)?
- Peak L2 cache bandwidth available to L1 cache (assuming 0% L2 misses)?
- Main memory bandwidth available to L2 cache?

Report the bandwidths in Gbytes/sec, i.e. 2^{30} bytes/sec.

B. You are given the following L1 cache statistics for a program executing on this system

Metrics	Access Type: Total	Instrn	Data	Loads	Stores
Accesses	10000000	7362210	2637790	1870945	766845
Misses	52206	8466	43740	36764	6976

```
Words Read From Lower-levels          180920 (i.e. 45230 cache lines)
Words Written-back to Lower-levels    766845
Total Traffic                         947765
```

How long does an average instruction take to execute (in ns), assuming 1 clock cycle per instruction in the absence of memory hierarchy stalls, no write buffering at the L1 cache level, and 0% L2 miss rate? Ignore register dependencies between instructions.

Computer Science Comprehensive Examination
Computer Architecture
[60 points]

This examination is open book. Please do all of your work on these sheets. Do not do your work in a blue book.

Number: ____

Problem	Max Score	Your Score
1	20	
2	20	
3	20	
TOTAL	60	

Problem 1: Short Answer [20 points]

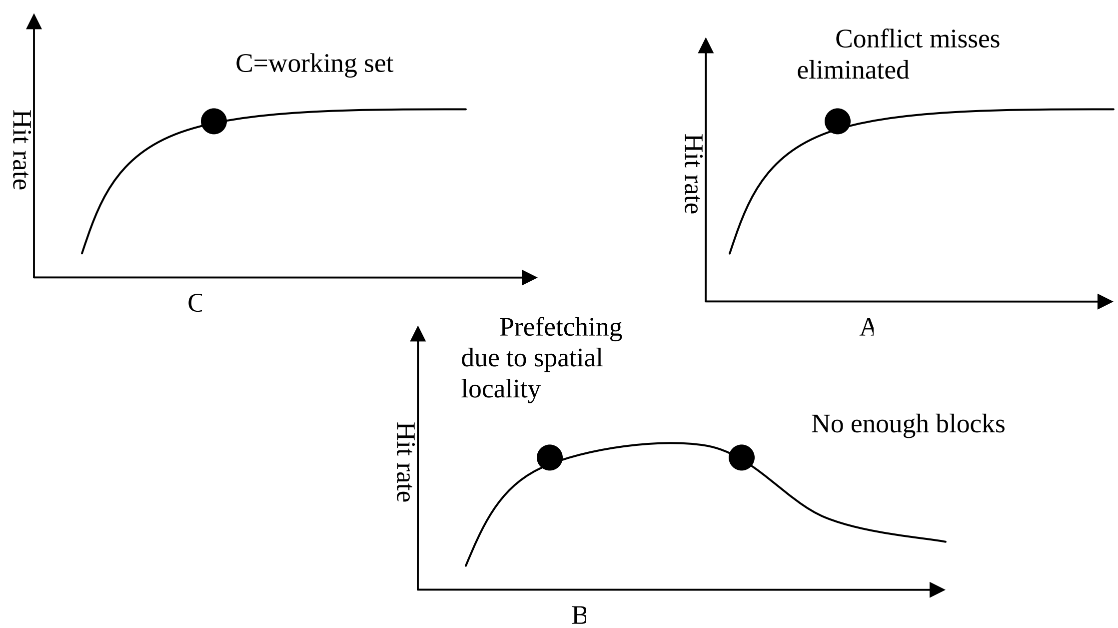
A. [7 points] Assume a pipelined processor with N pipeline stages. As we increase N, briefly explain that happens to the following (increase/decrease + 1 sentence reasoning)

- Clock cycle time: Decreases – fewer gates per pipeline stage
- Cycles per instructions: Increases – more data and control hazards

Considering clock cycle time only, is there a limit or point of diminishing returns for the number of pipeline stages in a processor? Why?

Yes there is. Clock cycle is limited by clock skew and pipeline register clocking overhead. As we subdivide the work for each instruction in more pipeline stages, eventually the clock cycle will become equal to the skew+overhead. Deeper pipelining, will not lead to additional clock cycle benefits.

[7 points] Assume a first level data cache with capacity (C), associativity (A), and block size (B). Draw the expected effect on hit rate as C, A, and B increase respectively. In each case, we vary one of the three parameters, while the other two are kept constant. Mark any interesting points on the graphs with 3-4 words to provide some reasoning. .



B. [6 points] What does the following MIPS code do? What is the instruction cache miss rate? What is the data cache miss rate? Assume the caches are initially empty and have a 16 byte line size.

```
        move  $t0, $a0
        addi  $t1, $t0, 4000
loop:   Sw    $t0, 0($t0)
        addi  $t0, $t0, 4
        slt   $t2, $t0, $t1
        bne   $t2, $zero, loop
exit:
```

The code intitializes a 1000 words to the memory address of the word

Icache miss rate = 2/(2 + 1000*4) = 2/4002 = 0.05%
Dcache miss rate = 1/4 = 25%

Problem 2: Instruction-Set Architecture Design [20 Points]

Your task is to redesign the way branches work in the MIPS ISA to create a new architecture called MIPS-new. You decide to add branches with arbitrary compares which are called complex branches. You are given the following information about MIPS and MIPS-new

Instruction type	Frequency
compares	23%
ALU ops (not compares)	20%
loads/stores	30%
conditional branches	25%
jumps	2%

Instruction mix on MIPS

Branch type	Frequency
NE/EQ compare to nonzero	30%
NE/EQ compare to zero	25%
LT/LE compare	25%
GT/GE compare	20%

Frequency of branch compares on MIPS-new

A. [10 points] Using the information given above, what is the ratio of instruction count (IC) of MIPS to MIPSnew.

```
IC_MIPS-new = IC_MIPS - compares eliminated
compares eliminated = branch compares except NE/EQ compares to zero
IC_MIPS-new = IC_MIPS - %branches(%branch compares eliminated)
IC_MIPS-new = IC_MIPS (1 - 25%(75%)) = IC_MIPS81.25%
IC_MIPS/IC_MIPS-new = 1.23
```

B. [10 points] Complex branches in MIPS-new force you to move the branch decision point from the ID stage to the EX stage. To maintain compatibility with MIPS, you decide to use delayed branching for the first cycle of branch delay and to predict taken for any subsequent cycles of branch delay. Assume that the target PC is calculated in the ID stage. If the delay slot is usefully filled 50% of the time and branches are taken 70% of the time, what is the ratio of CPIs of MIPS to MIPS-new? Assume that noops do not count as instructions and all instructions except branches and jumps have a CPI of 1.0.

CPI_MIPS = 1 + CPI_branch-delay + CPI_jump-delay

In MIPS the single delay slot is usefully filled 50% of the time

CPI_branch-delay + CPI_jump-delay = 25% (0.5) + 2%(0.5)

CPI_MIPS = 1.14

CPI_MIPS-new = 1 + CPI_branch-delay + CPI_jump-delay

In MIPS-new the first cycle of branch delay behaves like MIPS and the second cycle of branch delay is hidden when the branch is taken (70% of the time) and all the time for jumps.

You must renormalize the instruction mix to account for the compares that have been eliminated

MIPS-new branch frequency = 25%/81.25% = 31%

MIPS-new jump frequency = 2%/81.25% = 2.5%

CPI_branch-delay + CPI_jump-delay = 31%(0.5 + (1-70%)) + 2.5%(0.5)

CPI_MIPS-new = 1.26

CPI_MIPS/CPI_MIPS-new = 1.14/1.26 = 0.9

Problem 3: Memory Hierarchy Design [20 Points]

You have a computer with two levels of cache memory and the following specifications:

- Processor: 2GHz, 64-bit RISC CPU
- On-chip L1 caches
  - split instruction & data cache, blocking, single-ported
  - write-through & non-write allocate
  - 1 CPU cycle access time
  - Block size = 32 bytes
- Off-chip L2 cache off-chip
  - unified single-ported cache, blocking
  - write-back
  - 10 CPU cycles access time (L1 miss penalty) for both reads and writes
  - Block size = 32 bytes
- Main memory:
  - Bus: 64-bit data transfers at 400 MHz
  - Latency: 15+5+5+5 CPU cycles access time for 32 bytes (L2 miss penalty – includes latency of both DRAM and memory bus)

A. What is the:

- Peak L1 data cache bandwidth available to CPU (assuming 0% L1 misses)?
- Peak L2 cache bandwidth available to L1 cache (assuming 0% L2 misses)?
- Main memory bandwidth available to L2 cache?

Report the bandwidths in Gbytes/sec, i.e. 2^{30} bytes/sec.

Peak L1 cache bandwidth:
8 bytes / 1 CPU cycle access time = 8 bytes/0.5ns = 16 Gbytes/sec

Peak L2 cache bandwidth:
32 bytes / 10 CPU cycle access time = 32 bytes/5ns = 6.4 Gbytes/sec

Peak memory bandwidth:
32 bytes / 30 CPU cycle access time = 32 bytes/15ns = 2.13 Gbytes/sec

B. You are given the following L1 cache statistics for a program executing on this system

Metrics	Access Type: Total	Instrn	Data	Loads	Stores
Accesses	10000000	7362210	2637790	1870945	766845
Misses	52206	8466	43740	36764	6976

```
Words Read From Lower-levels          180920 (i.e. 45230 cache lines)
Words Written-back to Lower-levels    766845
Total Traffic                         947765
```

How long does an average instruction take to execute (in ns), assuming 1 clock cycle per instruction in the absence of memory hierarchy stalls, no write buffering at the L1 cache level, and 0% L2 miss rate? Ignore register dependencies between instructions.

Any instruction that hits in the cache will not be penalized by the misses. (Unless of course there are data dependencies, but since the problem tells us to ignore this). Thus, we just need to find average miss penalty of an instruction since that will incur extra latency. Note, that since we have a 0% L2 miss rate, we never incur any main memory accesses.

Average instruction latency = $latency_{ideal}$ + extra $latency_{stalls}$
= 1 + instruction stall cycles + data stall cycles

Instruction stall cycles = *I$ miss rate * L1 miss penalty*
Data stall cycles = (*load rate * load miss rate * L1 miss penalty*) + (*store rate * write penalty*)

The L1 miss penalty is 10 cycles. Since L1 is write-through, we assume the write penalty is the same as the L1 miss penalty which is 10 cycles. Also, since the L2 is single-ported, there may be the case where a instruction miss and a data miss occurs at the same time and there will be a structural hazard. In that case, one of the misses will incur an extra 10 cycle penalty. However the chance of this happening is almost zero: *I$ miss rate * D$ miss rate* = 0.115%*1.66% ~= 0%. Therefore we've neglected this in our equation.

The *I$ miss rate* is given in the table as: 8466/7362210 = 0.115%
The *load rate* is given in the table as: 1870945/7362210 = 25.41%
The *load miss rate* is given in the table as: 36764/1870945 = 1.965%
The *store rate* is given in the table as: 766845/7362210 = 10.42%

Thus,
Instruction stall cycles = 0.115% * 10 = 0.0115 cycles
Data stall cycles = 25.41% * 1.965% * 10 + 10.42% * 10 = 1.09 cycles
Average instruction latency = 1 + 0.0115 + 1.09 = 2.1015 cycles
So, average instruction time = Average latency * cycle time = 2.1015 * 0.5ns = 1.05ns

C. You are considering replacing the L2 cache with a victim cache. Given the information provided to you, compute a measure of "speed" for each alternative and indicate which is the faster solution. Justify the metric you choose to compare the two alternatives and state your assumptions. Assume the performance statistics are:

- L2 cache local miss ratio = 0.18
- Victim cache miss ratio = 0.23
- Victim cache transport time from L1 miss = 2 CPU clock

Hint: Use a metric that's simple and is representative of the common case.

Given the information provided, it's clear that the common case is a cache read. The reads, i.e. instruction fetch and loads, account for (7362210+2637790)/10000000 = 92.3% of total L1 cache accesses. Also, the write miss rate is much lower than the load miss rate. Thus, one metric that we can use for comparison then is **AMAT of a read.**

The L1 read miss rate = (instruction & load misses) / (instruction & load accesses)

= (8466 + 36764) / (7362210 + 1870945) = 0.4899%

1. L2 Cache,

$AMAT_{L2}$ = 10 + L2 miss rate * 30 = 10 + 0.18*30 = 15.4 cycles

$AMAT_{L1}$ = 1 + L1 read miss rate * $AMAT_{L2}$ = 1 + 0.004899*15.4 = 1.0754 cycles

2. Victim Cache,

$AMAT_{VC}$ = 2 + VC miss rate * 30 = 1 + 0.23*30 = 8.9 cycles

$AMAT_{L1}$ = 1 + L1 read miss rate * $AMAT_{L2}$ = 1 + 0.004899*8.9 = 1.0436 cycles

So, it would seem that the victim cache would be a better choice in this case.

Stanford University
Computer Science Department

Fall 2004 Comprehensive Exam in Databases

1. **Open Book & Notes / No Laptops.**
2. **The exam is timed for 60 minutes.**
3. **Write your Magic Number on this Honor Code statement sheet and the cover sheet of the exam.**

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 un-permitted 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.

Magic Number----------------------------------

Stanford University Computer Science Department
2004 Comprehensive Exam in Databases

- The exam is *open book and notes.*
- There are 8 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	7	8	TOTAL
Max. points	15	10	5	4	5	6	5	10	60
Points									

1. (15 points) The following facts must be converted to an E/R diagram:

- All motors have a manufacturer, a model number that is unique for that manufacturer, and a number of horsepower.
- Some motors are gasoline engines, and have a number of miles-per-gallon.
- Some motors are electric motors, and have a voltage.
- All automobiles have a manufacturer, a model name that is unique among all automobiles, and a price. All automobiles have one gasoline engine.
- Some automobiles are hybrid and also have an electric motor. For all hybrid cars, there is a waiting time for delivery.

(a) (10 pts.) Draw an E/R diagram that represents these facts. Make sure there is no redundancy in your design, and entity sets are not used where an attribute will do. Indicate keys for entity sets. Indicate when a relationship is many-one, and also whether it is "onto"; i.e., there is guaranteed to be an entity on the "one" side associated with each entity on the "many" side.

(b) (5 pts.) Convert your E/R diagram from part (a) into a relational database schema. For motors, use the object-oriented style, where each motor entity is represented in exactly one relation. For automobiles, use the E/R style, where an automobile entity is represented in as many relations as it belongs to entity sets in the automobile hierarchy. Your schema can be informal, i.e., just the names of the relations and their attributes; it does not have to be SQL `CREATE TABLE` statements. However, indicate keys for your relations.

2. (10 points) Consider the following relational schema:

```
Student(ID, SSN, Name)   \\ ID is a key; SSN is a key
StudVoter(SSN, State)    \\ (SSN,State) is a key
```

You may assume the `StudVoter` relation only contains students—that is, any `SSN` in `StudVoter` also appears in `Student`.

Write a SQL query that lists the ID's (no duplicates please) of all students registered to vote in three or more states. *Your query will be graded on simplicity as well as correctness.*

3. (5 points) Start with the schema from the previous problem. Add to the schema one or more SQL99 tuple-based `CHECK` constraints that ensure no student is registered to vote in more than one state.

4. (4 points) Is there an easier way to modify the original schema to express the "no student is registered to vote in more than one state" requirement from the previous problem? If so, show it. If not, briefly explain why not.

5. (5 points) Continuing from the previous problem, assume the "no student is registered to vote in more than one state" requirement is being enforced, one way or another. Now suppose in addition you want to ensure that every student is indeed registered to vote. Can you express this requirement using SQL99 `CHECK` constraints? If so, add the `CHECK` constraints to the schema. If not, briefly explain why not, and suggest other mechanisms that could be used to enforce the desired requirement.

6. (6 points) Continuing with the same schema, consider the following two transactions:

```
T1: select count(*) from StudVoter;
    select count(*) from StudVoter;

T2: insert into StudVoter values ('123-45-6789', 'California');
```

(a) Do the possible behaviors of these two transactions differ if we choose isolation level `repeatable-read` versus isolation level `serializable`? Answer "Yes" or "No" with a brief explanation.

(b) Now answer the same question, except replace transaction `T2` with:

```
T2: delete from StudVoter
    where SSN='123-45-6789' and state='California'
```

Do the possible behaviors of these two transactions differ if we use isolation level `repeatable-read` versus isolation level `serializable`? Answer "Yes" or "No" with a brief explanation.

7. (5 points) Continuing with the same schema, using standard SQL capabilities is it possible to authorize a database user to query only those students registered in California or Nevada? If so, briefly explain how. If not, briefly explain why not.

8. (10 points) Consider a relation $R(A, B, C, D)$ with the functional dependencies $ABC \to D$, $D \to C$, and $C \to B$.

(a) (3 pts.) Find all the keys for R.

(b) (1 pt.) Which of the given functional dependencies violate Boyce-Codd Normal Form?

(c) (1 pt.) Which, if any, of the given functional dependencies violate Third Normal Form. Explain why, if so.

(d) (2 pts.) How many superkeys does R have?

(e) (3 pts.) Explain why the multivalued dependency $D \twoheadrightarrow A$ holds in R.

Computer Graphics Comprehensive Exam

Computer Science Department
Stanford University
Fall 2004

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] Computer graphics definitions.

Define in a few sentences each of the following computer graphics terms. Some of these terms may be used in other fields, so be sure to give the computer graphics meaning.

1A [5 points] Law of diffuse reflection.

1B [5 points] Gamma.

1C [5 points] Image matte.

1D [5 points] Bump-mapping.

2. [20 points] Homogeneous coordinates.

Computer graphics relies heavily on geometry. One powerful approach represents points using homogeneous coordinates. In 3D, the position of a point in normal coordinates is denoted by a 3-vector (x, y, z). Using homogeneous coordinates, the same point is represented as a 4-vector (xw, yw, zw, w). The 3D position of the point represented using homogeneous coordinates is defined to be equal to (xw/w, yw/w, zw/w).

2A [4 points]. What is the geometric interpretation of a point whose 4^{th} coordinate w equals 0?

2B [8 points]. It is common in computer graphics to transform points using linear transformations. A linear transformation may be represented by a matrix. To transform a 3D point represented in homogeneous coordinates, a 4x4 matrix is used. Write out the 4x4 matrix that translates a point by the amount (tx, ty, tz).

2C [8 points]. Suppose the viewer is placed at the origin, and the image plane is placed at z=1. Write out the 4x4 matrix that performs a perspective projection of the point (x,y,z) from the point of view of the viewer onto the image plane.

3. [20 points] Color.

The human visual system is trichromatic. That means that the perceived color from any visible spectrum may be described by three coordinates.

3A [6 points]. Describe the physiological basis of trichromatic color vision in humans.

3B [7 points]. Cathode-Ray Tubes (monitors) are made using three light emitting phosphors because human vision is trichromatic. Because phosphors emit light, monitors use an additive color space. Suppose you were to use 4 phosphors. Describe the general form of the equation that would relate the brightnesses and colors of the 4 phosphors to the 3 colors perceived by the observer.

3C [7 points]. Color prints use colored inks on white paper. The color inks act as color filters. For this reason, the color space used in color printing is a subtractive color space. Suppose you used 4 inks for color printing. Describe the general form of the equation that would relate the quantities and the colors of the inks to the perceived colors.

4. [20 points] Clipping.

Consider a graphics package that draws 2D line segments. Normally the image is rectangular and one of the first steps is to clip the 2D line segment - returning the portion of the line segment that is inside the image rectangle. This clipped line segment is then drawn into the framebuffer.

Suppose the image is circular, not rectangular. Describe an algorithm for clipping a 2D line segment to a circle. The clipping routine should return the portion of the line segment inside the circle. Assume the line segment is given by the two points (x1, y1) and (x2, y2) and that the circle is the unit circle (radius equal to 1) centered at the origin. Make sure to consider all the cases that can occur. Use pseudocode to describe your algorithm, but work out the mathematics in detail.

5 [20 points] Key-frame Interpolation.

The classic method used to produce animation is key-frame interpolation. In this technique, images are created at certain frames, the key frames, and inbetween frames are interpolated from the previous and next key frames. In the first computer animation systems, the objects in the key frames were represented by polygons and lines.

5A [10 points]. Suppose that two successive key frames have the same number of polygons. Suppose further that each polygon in the first key frame is in correspondence with a polygon in the next key frame (that is, each vertex in one polygon matches a vertex in the corresponding polygon in the other key frame). Suppose further that the first key frame is at frame 0 and the next key frame is at frame 10. How would you create a polygon at frame i using linear interpolation (write out a formula for the positions of the interpolated vertices)? Use the notation Vn for vertex n in key frame 0 and Wn for vertex n in key frame 10.

5B [5 points] Linear interpolation often creates a very mechanical, non-life-like motion. How would you change the interpolation so that it was smoother?

5C [5 points] Suppose that each polygon in one key frame has a matching polygon in the other key frame, but that these polygons have a different numbers of vertices so that vertices are no longer in direct correspondence. Briefly describe an approach to interpolating between two polygons in this case.

Comprehensive Examination in Logic

Stanford University

Department of Computer Science

November 2004

THE HONOR CODE:

(A) The honor code is an undertaking of the students individually and collectively :

(I) 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;

(II) 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 avoid, as far as possible, academic procedures that create temptation to violate the Honour code.

(C) 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.

	CORRECT	INCORRECT	UNANSWERED	SCORE
SECTION A				
SECTION B				
TOTAL				

WRITE MAGIC NUMBER: ______________

Instructions

Please read these instructions and the *Notations* section carefully. **Do not read beyond this page until instructed to do so.**

- The exam is open book and open notes. But no laptops or electronic accessories are allowed.
- **All questions have penalties for wrong answers.** Read the instructions carefully before you start.
- Be sure to write your **magic number** on the previous page.

Notations

The notation is the one used by Enderton in *A Mathematical Introduction to Logic*, with the difference that the equality symbol is denoted by $==$ instead of $\approx$ and arguments to predicate and function symbols are enclosed in parentheses and separated by commas. Thus, for example, instead of Enderton's $fxyz$, $f(x, y, z)$ is used.
In some problems, the following symbols are used, whose definition is repeated here for completeness:

- $\text{Cn}(\Gamma)$ is the set of logical consequences of an axiom set Γ;
- $\text{Th}(\mathfrak{M})$ is the first-order theory of the structure $\mathfrak{M}$, i.e. the set of first-order sentences, of a given language, that are true in $\mathfrak{M}$.
- The composition of variable substitutions σ and τ is denoted by $\sigma \circ \tau$.

Do not turn this page until instructed to do so.

Section A

For each question in this section you need to choose one out of five choices provided. If you answer correctly you get **4 points.** Answering incorrectly will result in **3 points** being **deducted** from your score. Indicate your answer by writing your choice clearly in the box provided. **No points are deducted for leaving questions unanswered.**

1. Let $\mathcal{F}$ be a non-valid sentence. Which of the following statements are **possibly** true?

I. $(\neg\ \mathcal{F})$ is valid.

II. $(\neg\ \mathcal{F})$ is satisfiable.

III. $\mathcal{F}$ is unsatisfiable.

IV. $(\neg\ \mathcal{F})$ is unsatisfiable.

(A) **II** only.

(B) **I** and **II** only.

(C) **I, II** and **III** only.

(D) **II, III** and **IV** only.

(E) all of them.

2. Consider deductive tableaux for propositional logic with resolution and the **"polarity strategy"**. Let ψ be a valid proposition. Which of the following statements are **necessarily** true?

I. There may exist a proof of ψ which uses resolution but does not necessarily follow the polarity strategy.

II. A tableau construction using resolution with the polarity strategy always terminates with a proof of ψ.

III. There always exists a proof of ψ using resolution with the polarity strategy.

IV. A tableau construction using resolution only, always terminates with a proof of ψ.

(A) **I, II** and **III** only.

(B) **II** and **IV** only.

(C) **II** and **III** only.

(D) **I** and **III** only.

(E) all of them.

3. Which of the following sentences in first-order logic do **not** have a one-element model?

I. $(\forall x)(\exists y)R(x,y) \wedge (\forall x)[\neg R(x,x)]$.

II. $(\forall x)[R(x) \to (R(a) \vee R(b))]$.

III. $(\forall x)(\exists y)[R(x,x) \equiv \neg R(x,y)]$.

IV. $(\forall x)(\forall y)[R(y,x) \equiv \neg R(x,y)]$.

(A) **II** and **III** only.

(B) **II** and **IV** only.

(C) **III** and **IV** only.

(D) **I**, **III** and **IV** only.

(E) all of them.

4. Assume duality is added as a rule to the tableau. Which of the following sets of rules are complete for propositional logic tableau?

I. Duality, AA resolution without the polarity strategy.

II. Duality, GG resolution with the polarity strategy.

III. Duality, AA and GG resolutions without the polarity strategy.

IV. Duality with splitting rules.

(A) **II** only.

(B) **II** and **III** only.

(C) **II**, **III** and **IV** only.

(D) **I**, **III** and **IV** only.

(E) **I**, **II** and **III** only.

5. Given a theory $\mathcal{T}$ and a first-order formula ψ, if ψ is valid in $\mathcal{T}$, which of the following statements is **necessarily** true about $\mathcal{T}$ and ψ?

I. ψ is valid in first-order logic.

II. ψ is satisfiable in first-order logic.

III. $\mathcal{T}$ is consistent.

IV. For every model $\mathcal{M}$, if $\mathcal{M}$ satisfies $\mathcal{T}$ then $\mathcal{M}$ also satisfies ψ.

(A) **II** and **IV** only.

(B) **II** only.

(C) **II** and **III** only.

(D) **II**, **III** and **IV** only.

(E) **IV** only.

6. Let Σ be a **set** of first-order sentences. Which of the following statements about Σ are **necessarily** true?

I. For any ψ, if $\Sigma \models \psi$, then $\Sigma \not\models \neg\psi$.

II. If Σ is unsatisfiable then every subset of Σ is unsatisfiable.

III. If $\Sigma \models \psi$ and $\Sigma \models \neg\psi$ for some ψ then Σ does not have a model.

IV. If Σ is inconsistent then there is no ψ such that $\Sigma \models \psi$.

(A) **I** only.

(B) **III** only.

(C) **I** and **III** only.

(D) **II** and **IV** only.

(E) none of them.

7. Let $\mathcal{T}, \mathcal{T}_1, \mathcal{T}_2$ be first-order theories with $\mathcal{T}_1 \subseteq \mathcal{T}$ and $\mathcal{T}_2 \supseteq \mathcal{T}$. Which of the following statements are **necessarily** true?

I. If $\mathcal{T}$ is undecidable, then $\mathcal{T}_1$ is undecidable.

II. If $\mathcal{T}$ is undecidable, then $\mathcal{T}_2$ is undecidable.

III. If $\mathcal{T}$ is decidable, then $\mathcal{T}_1$ is decidable.

IV. If $\mathcal{T}$ is decidable, then $\mathcal{T}_2$ is decidable.

(A) **II** only.

(B) **III** only.

(C) **II** and **III** only.

(D) **I** and **IV** only.

(E) none of them.

8. Let $\mathcal{F}$ be a valid sentence in predicate logic. Which of the following statements are **necessarily** true?

I. $\mathcal{F}$ is valid in the theory of natural numbers with 0,1, addition and multiplication.

II. $\mathcal{F}$ is valid in the theory of natural numbers **with** 0,1, addition and **without** multiplication.

III. $\mathcal{F}$ is true in every model of the theory of natural numbers **with** 0, 1, addition and **without** multiplication.

IV. There are **first–order** theories $\mathcal{T}$ such that $\mathcal{T} \not\models \mathcal{F}$.

(A) **III** only.

(B) **II** and **III** only.

(C) **I, II** and **III** only.

(D) **II, III** and **IV** only.

(E) all of them.

9. Which of the following variable substitutions are the most general unifiers (mgu's) of x and y?

I. $\{x \leftarrow y\}$.

II. $\{x \leftarrow y, y \leftarrow x\}$.

III. $\{x \leftarrow z, y \leftarrow z\}$.

IV. $\{x \leftarrow z, y \leftarrow z, z \leftarrow y\}$.

(A) **I** only.

(B) **I** and **IV** only.

(C) **I**, **III** and **IV** only.

(D) **I**, **II** and **III** only.

(E) all of them.

10. Let $\mathcal{F}$ and $\mathcal{G}$ be two expressions. Which of the following statements about unifiers of $\mathcal{F}$ and $\mathcal{G}$ are **necessarily** true? (Recall that $\circ$ denotes the composition operation of variable substitutions.)

I. If θ is a mgu, and θ' is a permutation, then $\theta \circ \theta'$ is a mgu.

II. If both θ and θ' are mgu's, then there exists a permutation θ'' such that $\theta \circ \theta'' = \theta'$.

III. If both θ and θ' are mgu's, then $\theta \circ \theta'$ is also a mgu.

IV. If neither θ nor θ' is a mgu, then $\theta \circ \theta'$ can not be a mgu.

(A) **I**, **II** and **III** only.

(B) **I** only.

(C) **II** only.

(D) **I** and **II** only.

(E) **I** and **IV** only.

11. Let $\mathcal{F}$ and $\mathcal{G}$ be sentences in first-order logic. Given that $\mathcal{F}$ is valid if and only if $\mathcal{G}$ is **not** valid, which of the following statements are **necessarily** true?

I. $(\mathcal{F} \not\equiv \mathcal{G})$ is valid.

II. $(\mathcal{F} \not\equiv \mathcal{G})$ is satisfiable.

III. $(\mathcal{F} \equiv \mathcal{G})$ is unsatisfiable.

IV. $(\mathcal{F} \equiv \mathcal{G})$ is not valid.

(A) **II** and **IV** only.

(B) **I** and **III** only.

(C) **II** only.

(D) **IV** only.

(E) none of them.

12. Given a sentence $\mathcal{F}$, we obtain $\mathcal{G}$ by removing all its quantifiers through **validity preserving** skolemization. Which of the following statements are **necessarily** true about $\mathcal{F}$ and $\mathcal{G}$?

I. $\neg\mathcal{G}$ is satisfiable if and only if $\neg\mathcal{F}$ is satisfiable.

II. $\mathcal{G}$ may not exist for any given $\mathcal{F}$.

III. It is always possible to skolemize in such a way that $\mathcal{G}$ and $\mathcal{F}$ are equivalent.

IV. If $\mathcal{F}$ and $\mathcal{G}$ are not equivalent, then $\mathcal{F}$ is not valid.

(A) **IV** only.

(B) **I** and **IV** only.

(C) **III** only.

(D) **II** only.

(E) **II** and **III** only.

13. Which of the following statements is **necessarily** true about a first-order theory $\mathcal{T}$?

I. If $\mathcal{T}$ is finitely axiomatizable, then $\mathcal{T}$ is decidable.

II. If $\mathcal{T}$ is not finitely axiomatizable, then $\mathcal{T}$ is undecidable.

III. If $\mathcal{T}$ is not finitely axiomatizable, then $\mathcal{T}$ does not have a finite model.

IV. If $\mathcal{T}$ is finitely axiomatizable, then $\mathcal{T}$ has a finite model.

(A) **I** and **III** only.

(B) **II** and **IV** only.

(C) **IV** only.

(D) **III** only.

(E) none of them.

14. Consider the sentence

$$(\forall x)(\exists y)[(\exists z)(\forall w)\varphi(x,y,z,w) \rightarrow (\forall z)(\exists w)\psi(x,y,z,w)].$$

Which of the following can appear as the results of one or more steps of **validity-preserving** skolemization?

I. $(\exists y)[(\exists z)(\forall w)\varphi(a,y,z,w) \rightarrow (\forall z)(\exists w)\psi(a,y,z,w)]$.

II. $(\exists y)[(\forall w)\varphi(a,y,f(y),w) \rightarrow (\forall z)(\exists w)\psi(a,y,z,w)]$.

III. $(\exists y)[(\forall w)\varphi(a,y,f(y),w) \rightarrow (\exists w)\psi(a,y,g(y),w)]$.

IV. $(\forall x)(\exists y)[(\forall w)\varphi(x,y,f(y),w) \rightarrow (\exists w)\psi(x,y,g(y),w)]$.

(A) **I** only.

(B) **III** only.

(C) **I** and **III** only.

(D) **I**, **II** and **III** only.

(E) all of them.

15. Consider the following terms:

$$\begin{aligned} t_1 &: \ f(x, g(x)) \\ t_2 &: \ f(y, g(y)) \\ t_3 &: \ f(g(z), g(y)) \end{aligned}$$

Which of the following statements are **necessarily** true?

I. t_1 and t_2 are unifiable.

II. t_2 and t_3 are unifiable.

III. t_1 and t_3 are unifiable.

IV. The tuple (t_1, t_2, t_3) is unifiable.

(A) **I** only.

(B) **I** and **II** only.

(C) **I** and **III** only.

(D) **I**, **II** and **III** only.

(E) all of them.

16. Let $\mathcal{F}, \mathcal{G}$ be arbitrary predicate logic formulas, and $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ be defined as follows:

$$\begin{aligned} \mathcal{S}_1 &: (\forall *)(\mathcal{F} \rightarrow \mathcal{G}) \\ \mathcal{S}_2 &: (\forall *)\mathcal{F} \rightarrow (\forall *)\mathcal{G} \\ \mathcal{S}_3 &: (\exists *)(\mathcal{F} \rightarrow (\forall *)\mathcal{G}) \end{aligned}$$

Which of the following statements are **necessarily** true?

I. $\mathcal{S}_1$ implies $\mathcal{S}_2$.

II. If $\mathcal{S}_1$ is valid, then $\mathcal{S}_2$ is also valid.

III. If $\mathcal{S}_3$ is valid, then $\mathcal{S}_1$ is also valid.

IV. If $\mathcal{S}_1$ is valid, then $\mathcal{S}_3$ is also valid.

(A) **I** and **II** only.

(B) **I** and **III** only.

(C) **II** only.

(D) **I**, **II** and **IV** only.

(E) all of them.

17. Which of the following can appear as the result of an application of AG or GA resolution rule between assertion

$$\mathcal{A}: \qquad \big(p(x,y) \vee q(a,a)\big) \rightarrow q(y,b)$$

and goal

$$\mathcal{G}: \qquad q(a,z) \vee r(z)?$$

I. $\neg(p(x,y) \rightarrow q(y,b))$.

II. $p(x,a) \vee q(a,a)$.

III. $r(b)$.

IV. $r(a) \wedge (\neg q(y,b))$.

(A) **I**, **II** and **IV** only.

(B) **I**, **II** and **III** only.

(C) **I**, **III** and **IV** only.

(D) **II**, **III** and **IV** only.

(E) all of them.

Section B

For each question in this section you need to write "yes" if you think the statement holds or "no" if you think it does not. You receive **2** points if you answer correctly. However **2** points will be **deducted** for wrong answers. There is no penalty for leaving questions unanswered.

18. $(\exists x)\big(\varphi(x) \to (\forall y)\varphi(y)\big)$ is a valid first-order sentence.

19. $(\exists x)(\forall y)(\exists z)(x + y = z)$ is a valid first-order sentence.

20. If a sentence φ is only satisfiable in infinite models, then $\neg\, \varphi$ is satisfiable.

21. Any first-order theory which is recursive is axiomatizable.

22. If $\mathcal{A} = \langle A, < \rangle$ is a well-founded structure, then there is no infinite chain of the form

$$\ldots < a_{-i} < \ldots < a_{-1} < a_0 < a_1 < \ldots < a_i < \ldots$$

23. Let $\mathcal{F}$ and $\mathcal{G}$ be two first-order unifiable expressions. Then there exist infinitely many mgu's of $\mathcal{F}$ and $\mathcal{G}$.

24. There are infinitely many idempotent variable permutations.

25. A tuple of three expressions is unifiable if and only if any two of those expressions are unifiable.

26. It is not possible to write a first-order sentence $\mathcal{F}$ such that $\mathcal{F}$ is true and only true in all finite models.

27. Recall that $\mathrm{Th}(\mathfrak{M})$ denotes the first-order theory of the structure $\mathfrak{M}$. Let $\mathfrak{N} = \langle N, 0, 1, +, * \rangle$ denote the structure of natural numbers with addition and multiplication. Then $\mathrm{Th}(\mathfrak{N})$ is a complete theory.

28. If $\mathcal{T}$ is a complete first-order theory, then for every cardinality κ, $\mathcal{T}$ has, up to isomorphism, exactly one model.

29. $\{x \leftarrow f(x), y \leftarrow x\}$ is a mgu of x and $f(y)$.

30. If θ is a mgu of $\mathcal{F}$ and $\mathcal{G}$, and τ is **not** a variable permutation, then $\theta \circ \tau$ can not be a mgu of $\mathcal{F}$ and $\mathcal{G}$.

2004 Comprehensive Examination in Logic
Answers

Section A

Answer 1. C

Answer 2. D

Answer 3. D

Answer 4. E

Answer 5. E

Answer 6. B

Answer 7. E

Answer 8. C

Answer 9. B

Answer 10. D

Answer 11. A

Answer 12. B

Answer 13. E

Answer 14. E

Answer 15. E

Answer 16. D

Answer 17. A

Section B

Answer 18. T

Answer 19. F

Answer 20. T

Answer 21. T

Answer 22. T

Answer 23. T

Answer 24. F

Answer 25. F

Answer 26. T

Answer 27. T

Answer 28. F

Answer 29. T

Answer 30. F

Stanford University
Computer Science Department

<u>Fall 2004 Comprehensive Exam in NETWORKS</u>

1. **Closed book/ no laptops & notes. Answer in the Blue Book**
2. **The exam is timed for one hour.**
3. **Write your Magic Number on this sheet & on the Blue Book.**

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 un-permitted 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.

***Magic Number*---------------------------------**

Comprehensive Exam: Networks (60 points) Closed Book: Autumn 2004

1. (15 points total) *End to end*

   (a) (5 points) Define the so-called "end-to-end principle" as applied to the Internet.

   (b) (5 points) John Kerry claims that putting the transport layer protocol implementation in end systems is an evil corporate plot to drive up the costs and increase the "digital divide". His proposal is having a global coalition of "smart" routers that ensure that applications have reliable communication without having to run a transport protocol. How would you advise the senator before he goes public with this proposal?

   (c) (5 points) Howard Dean, objecting to TCP as a creation of the US military, claims that it (TCP) violates the end-to-end principle by providing 3-way handshake. Describe whether this is true or false, or a little of both, justifying your answer.

2. (15 points total) *Transport Protocol Design*

   (a) (8 points) Joe Conservative claims that TCP works well for his applications over the WAN but is inefficient for his LAN applications. Bill Liberal claims that TCP works for his applications over the LAN but is inefficient for his WAN applications. Describe how they can both be right, tying your answer to specific aspects of the TCP design.

   (b) (7 points) TCP has been improved with selective acknowledgment and fast retransmit. Pick one of these, describe briefly how it works, including a specific scenario in which it helps and one in which it does not help performance.

3. (15 points total) *Congestion Control*

   (a) (7 points) Larry Roberts, Bob Metcalfe and more recently Simon Lam (all recognized Internet heroes) have warned that the Internet is at risk of "congestion collapse" as a datagram network. Define "congestion collapse" and describe how this can happen, at least in theory, in the Internet.

   (b) (8 Points) The military becomes concerned about this congestion collapse issue and then is really panicked when they learn that "congestion control is implemented by voluntary actions of endsystem clients", as they describe it to you. Describe what this means, whether it is true and how you would advise them to deal with it, i.e. is there a better alternative approach, refinements or just fine the way it is.

4. (15 points total) *Ethernet*

   (a) (6 points) Describe each field in the Ethernet header and what properties each field has and why this field is required with those properties, if in fact it is.

   (b) (5 points) Describe how CSMA-CD works and how this access protocol is affected by the data rate of Ethernet moving from 10 Mbps to support all the way up to 10 Gbps.

   (c) (4 points) Peterson and Davie say: "it might seem that a wireless protocol would follow the exactly the same CSMA-CD algorithm as Ethernet" as a lead-in to why not. Describe why not and what 802.11 does about it.

The End

Computer Science Department
Stanford University

Comprehensive Examination in Numerical Analysis
Fall 2004

Note: You may use a result you are asked to prove in subsequent questions even if you have not been able to prove it.

1. Norms and orthogonality [10pts]

1. [4pts] Let $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ be an orthonormal basis for R^n and $\mathbf{A}$ a $n \times n$ matrix. If $\mathbf{A}\mathbf{q}_1, \mathbf{A}\mathbf{q}_2, \ldots, \mathbf{A}\mathbf{q}_n$ is an orthonormal set as well, prove that $\mathbf{A}$ must be orthogonal.

2. Let $\mathbf{A}, \mathbf{B}$ be $n \times n$ nonsingular matrices satisfying $\|\mathbf{A}\mathbf{x}\|_2 = \|\mathbf{B}\mathbf{x}\|_2$ for every $\mathbf{x} \in R^n$.

   (a) [3pts] Show that $\mathbf{A}$ and $\mathbf{B}$ have the same singular values.
   (*Hint:* Show that $\mathbf{A}^T\mathbf{A} = \mathbf{B}^T\mathbf{B}$)

   (b) [3pts] Show that $\mathbf{A} = \mathbf{Q}\mathbf{B}$ for an orthogonal matrix $\mathbf{Q}$.

2. Optimization and least squares [10 pts]

If $\mathbf{A}$ is an $m \times n$, $m > n$ matrix with full column rank and $\mathbf{b} \in R^m$, we know that the least squares solution to the overdetermined system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is given by the system of normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x}_0 = \mathbf{A}^T\mathbf{b}$ and corresponds to the vector $\mathbf{x}_0 \in R^n$ that minimizes $\phi(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.

1. [4pts] Consider the modified functional

$$\hat{\phi}(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \mathbf{x}^T\mathbf{B}\mathbf{x} - 2\mathbf{c}^T\mathbf{x}$$

   where $\mathbf{B}$ is an $n \times n$ symmetric and positive definite matrix, and $\mathbf{c} \in R^n$. Show that we can find the value of $\mathbf{x}$ that minimizes $\hat{\phi}(\mathbf{x})$ by solving the following modification of the system of normal equations

$$\left(\mathbf{A}^T\mathbf{A} + \mathbf{B}\right)\mathbf{x}_0 = \mathbf{A}^T\mathbf{b} + \mathbf{c}$$

2. [1pt] Explain why we can find a symmetric, positive definite matrix $\mathbf{M}$ such that $\mathbf{B} = \mathbf{M}^2 = \mathbf{M}^T\mathbf{M}$.

3. [4pts] Show that the value of $\mathbf{x}$ that minimizes $\hat{\phi}(\mathbf{x})$ can alternatively be found by finding the least squares solution of the following modification of the original overdetermined system

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{M} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{b} \\ \mathbf{M}^{-1}c \end{pmatrix}$$

where $\mathbf{M}$ is the matrix defined in (3)

4. [1pts] Why would you prefer either of the approaches described in (1) or (3) to solve the given minimization problem?

3. Differential equations [10pts]

Consider the scalar ordinary differential equation $y' = \lambda y,\ \lambda \in R$ and the following methods for solving it

$$\begin{aligned} \text{Forward Euler} &: \quad y_{k+1} = y_k + hy'_k \\ \text{Backward Euler} &: \quad y_{k+1} = y_k + hy'_{k+1} \\ \text{Trapezoidal} &: \quad y_{k+1} = y_k + \frac{h}{2}(y'_k + y'_{k+1}) \end{aligned}$$

1. [4pts] Prove that taking one step of forward Euler to get from y_k to y_{k+1} followed by a step of backward Euler to get from y_{k+1} to y_{k+2} is equivalent to taking one trapezoidal step from y_k to y_{k+2} (note that the integration interval will be equal to $2h$ for a step from y_k to y_{k+2}).

2. [4pts] Prove that we get the same result if we use backward Euler for the first step and forward Euler for the second.

3. [2pts] Explain why both methods described in (1) and (2) are stable for any $\lambda < 0$

Comprehensive Exam: Programming Languages Autumn 2004

This is a 30-minute closed-book exam and the point total for all questions is 30.

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	Total
Score					
Max	3	4	11	12	30

1. (3 points) *Type inference*

   What is the ML type of this function?

```
fun length nil = 0
|   length (first::rest) = 1 + length rest
```

2. (4 points) *Scoping*

   What is the difference between static and dynamic scope? Name a language or language feature that that uses static scope and one that uses dynamic scope.

3. (11 points) *Typing and Garbage Collection*

   Andrew Appel proposed an interesting scheme for representing the run-time type information used in garbage collection. There are two parts to his approach. First, the compiler generates tables that contain the types of global and stack allocated variables. Second, the compiler generates a compact representation of all the types in the form of a type graph. This graph contains the parse trees (or parse graphs) of all the types used in the program. The idea is that together, these two pieces of information should be sufficient to find the type of a data item at run time.

   For example, consider this simple program with one type and one stack allocated variable

```
TYPE t = RECORD
              val: INTEGER;
              next: REF t;
            END;
VAR v: REF t; (* v is a stack allocated variable of type REF t *)
```

   where `REF` indicates a pointer. At garbage collection time, the collector looks at all the global variables and stack allocated variables. In this example, that's just the variable `v`. Since the collector knows the types of all stack allocated variables, it knows that `v` points to an object of type `t`. Since `v` is on the stack, the collector knows that the object `v` points to is reachable. Using the type table, the collector can determine that the data pointed to by `v` has type `t`. The type graph tells the collector that this object contains a `next` field of type `REF t`, so the collector can follow the `next` field and mark the next object as reachable.

(a) (3 points) What problem is this solving? Is the goal here to improve the running time of programs, reduce the run-time space requirements or programs, provide additional programming flexibility, or accomplish something else?

(b) (2 points) Would this work better for Lisp, Scheme, or ML? (Don't worry about polymorphic functions in ML.) Why?

(c) (3 points) Could you use this approach for garbage collecting C programs? If so, how is way of representing type information and using it at collection time helpful (in comparison with the way Scheme or Lisp garbage collectors usually work)? What problem does it not solve?

(d) (3 points) Do you think this will work well for ML program that use polymorphism? Explain what problem might arise with polymorphism.

4. (12 points) *Method Lookup.*
Java and C++ use different implementations of method lookup for objects.

(a) (4 points) Suppose a Java program constains a declaration `XClass x = new ...` and invokes `XClass` method `m` on object `x`. Explain how the code for `m` is found at run time.

(b) (4 points) If a C++ program calls virtual member function `f` of object `x`, how is the code for `f` located? Describe the main data structure and briefly explain how it is used.

(c) (4 points) Describe the trade-off's between the two approaches. Which is more flexible? Which is more efficient? Why is each one appropriate for the language in which it is used.

2004 Programming Languages Comp. Solutions

written by 2006 Ph.D. First-Years

1. *Type inference:* `'a list -> int`

2. *Scoping:* Static scope is resolved at compile-time (use the value of the variable declared in the closest enclosing block); dynamic scope is resolved at execution-time (use the value of the variable closest to the current function's activation record on the stack). C++ is a language that uses static scope. Exceptions are a language feature that uses dynamic scope.

3. *Typing and Garbage Collection:*

   (a) This technique is solving the problem of tags needing to be kept alongside pointers in order to make precise garbage collection work. In order for garbage collection to be precise, the garbage collector must know which blocks of memory are actually pointers so that it can follow pointers to access other reachable objects. Keeping these tables around obviates the need to tag each pointer at run-time, thus giving space savings.

   (b) This will work better for ML because it has static types. Lisp and Scheme are dynamically-typed languages, so there is no way to build up a compile-time type table.

   (c) Yes, this is helpful for C programs because you can know which memory contents are actually pointers, instead of having to make conservative guesses (that any sufficiently large integer value that refers to some valid address range on the heap is a pointer). However, it does not solve the problem of pointers to the middle of objects, casts of integers to pointers (and vice versa), and the use of `void*` pointers to emulate object-oriented functionality.

   (d) No, it won't work with polymorphism because a variable can have more than one type at run-time, so there is no way to build up a table at compile-time that maps a variable to one type.

4. *Method Lookup:*

   (a) It is found via a lookup in a method dictionary where the dictionary for `XClass` contains an entry that maps the method name `m` to the address of the first instruction of that method (otherwise, its superclass should contain an entry for `m`). The dictionary for each class also contains a pointer to the dictionary of its superclass, so that if an entry isn't found, then the superclass's dictionary is searched, and so on until either the method is found or an exception results.

   (b) The main data structure is a `vtable`, which is an array containing addresses of the first instruction in each method. It is used by having the compiler hardcode in a fixed constant offset from the beginning of the `vtable` so that overridden methods in a subclass have the same offset as their respective methods in the superclass.

(c) The C++ `vtable` approach is more efficient because a direct array lookup is very fast, whereas the Java method dictionary method is more flexible because it can support dynamic class loading, etc. The `vtable` approach is more appropriate for C++ because performance is paramount for C++ programs, but for Java programs, flexibility and extensibility are more important.

Stanford University Computer Science Department

Fall 2004 Comprehensive Exam in Software Systems

1. **CLOSED BOOK**: no notes, textbook, computer, PDA, Internet access, etc.
2. **WRITE ONLY IN BLUE BOOKS: No credit** for answers written on these exam pages.
3. **EXPLAIN YOUR REASONING.** Answers with no explanation are insufficient.
4. The exam is designed to take about an hour if you budget 1 point per minute.
5. If you need to make assumptions to answer a question, *state them clearly*.

---

1) [5] Consider a virtual memory system with a single-level page table. The (dimensionless) page fault rate for a particular workload is *r;* the average DRAM access time is *d;* the average time to service a page fault is *f;* the (dimensionless) TLB hit rate is *h*. Write the expression for the average memory access time in this system.

2) [5] Consider three CPU scheduling disciplines: shortest-job-first, preemptive round-robin, and first-come-first-served. For each of these, if it is *starvation-free*, explain why; if it's not, *briefly* describe a scenario in which starvation could occur.

3) [4] Suppose a particular process makes a total of *p* page references, of which $n<=p$ are to distinct pages. (The ordering of page references is not known.) The process is allocated *m* frames of physical memory, initially all empty. In terms of these variables, give a *lower bound* and an *upper bound* on the number of page faults this process will experience, no matter what page-replacement strategy is used, and *explain your reasoning.*

4) [4] Debuggers like *gdb* let you set breakpoints that are triggered whenever a program variable, say *V,* is accessed or modified. How is this implemented? Under what circumstances, if any, does the usual implementation incur a performance cost when variables *other* than *V* are accessed? (To simplify the explanation, you may assume that we're only referring to global variables whose memory placement is known at load time.)

5) [4] Give an example of a scenario where memory-mapped I/O makes more sense than programmed I/O, and *why* memory-mapped would be better. Then give an example of the opposite case.

6) [4] To successfully prevent user programs from causing damage to other programs or the OS, hardware support is required. Name **two** hardware mechanisms in modern CPU's that supports this goal, and for each one, describe what specific kind(s) of damage it prevents.

7) [4] Why would some OS's support multiple page sizes instead of just one page size? What additional page-management issues does this raise?

8) [4] Describe the mechanism of *priority inheritance* and give an example of the kind of problem it's intended to solve.

9) [3] With respect to remote procedure calls (RPC), what is serialization (or marshalling) and why is it necessary? Are there cases where it is unnecessary?

10) [3] Describe one failure mode that might occur if a non-preemptive scheduler is used, and how it would be avoided with a preemptive scheduler.

11) [3] A particular email message you're sending is so sensitive that you wish to both encrypt and sign it (both using public-key cryptography). Under what circumstances, if any, would you encrypt it first and then sign it? Under what circumstances, if any, would you sign it first and then encrypt it? (In other words, what's the practical effect of doing it one way vs. the other?)

12) [3] True or false: all side-effect-free operations are idempotent, but not all idempotent operations are side-effect-free. Explain your answer concisely but completely (i.e. if true, explain why each part is true; if false, explain which part(s) are false and/or give counterexamples).

13) [2] To avoid replay attacks, one can either use a randomly-generated nonce or a physical timestamp. Give one advantage of using a nonce over a timestamp, and one advantage of using a timestamp over a nonce. (You may assume that the resolution of physical timestamps is sufficient to avoid timestamp value collisions.)

14) [2] Describe one type of file access control that can be performed with access-control lists (such as AFS uses) but cannot be performed with file permissions (such as traditional Unix filesystems use).

15) [2] You're asked to take an existing Java program and rewrite it in C++. What benefit would you *gain* by doing so? What benefit, if any, would you *lose* by doing so?

16) [2] What's the difference between a credential and a capability?

17) [2] What's the difference between thrashing and deadlock?

18) [2] Give one example of how a filesystem might become corrupted, ***other than corruption of data in the files themselves***

19) [2] Your C program has a bug that causes it to accidentally dereference a nonexistent array element, e.g. a[10] where array a[] has been statically declared as containing 8 elements. What happens when this bug occurs at runtime, and why?

THE END

Stanford University Computer Science Department

Fall 2004 Comprehensive Exam in Software Systems

SOLUTIONS

1. **CLOSED BOOK**: no notes, textbook, computer, PDA, Internet access, etc.
2. **WRITE ONLY IN BLUE BOOKS: No credit** for answers written on these exam pages.
3. **EXPLAIN YOUR REASONING.** Answers with no explanation are insufficient.
4. The exam is designed to take about an hour if you budget 1 point per minute.
5. If you need to make assumptions to answer a question, *state them clearly*.

---

1) [5] Consider a virtual memory system with a single-level page table. The (dimensionless) page fault rate for a particular workload is *r;* the average DRAM access time is *d;* the average time to service a page fault is *f;* the (dimensionless) TLB hit rate is *h*. Write the expression for the average memory access time in this system.

This was tricky to compute, so I gave credit for either of two methods. If you break it down by whether a given access page faults or not, keeping in mind that a page fault takes time f plus two memory lookups (TLB miss + the actual access), you get $r(f + 2d) + (1-r)(hd + (1-h)2d)$. If you break it down by whether a given access hits in the TLB or not, you get $hd + (1-h)((1-r)2d + r(2d+f))$

2) [5] Consider three CPU scheduling disciplines: shortest-job-first, preemptive round-robin, and first-come-first-served. For each of these, if it is *starvation-free*, explain why; if it's not, *briefly* describe a scenario in which starvation could occur.

SJF is not starvation-free: short jobs could continue to arrive that always get favored over pending longer jobs. FCFS is starvation free as long as jobs terminate in bounded time. Preemptive RR is starvation-free because each process necessarily gets CPU time due to timer interrupts.

3) [4] Suppose a particular process makes a total of p page references, of which $n<=p$ are to distinct pages. (The ordering of page references is not known.) The process is allocated m frames of physical memory, initially all empty. In terms of these variables, give a *lower bound* and an *upper bound* on the number of page faults this process will experience, no matter what page-replacement strategy is used, and *explain your reasoning.*

No matter what, at least n page faults will occur since each page is touched at least once and the process's frames all start out empty. If $m>=n$, then after these n faults there will be no evictions, so the upper bound is n. If $m<n$, in the worst case every access after the m' th could still cause a fault (systematic eviction) so the worst case is p. It was OK to say the worst case is p as long as you qualified it with "if $m<n$".

4) [4] Debuggers like *gdb* let you set breakpoints that are triggered whenever a program variable, say *V,* is accessed or modified. How is this implemented? Under what circumstances, if any, does the usual implementation incur a performance cost when variables *other* than *V* are accessed? (To simplify the explanation, you may assume that we're only referring to global variables whose memory placement is known at load time.)

Page tables are modified to force a trap (usually due to an illegal page access) when the page containing *V* is accessed. Of course, since there may be other variables also stored on this page, a check is required (comparing the page offset of the address) to confirm whether the access is to *V*; if not, it's a "false alarm". So a performance penalty is incurred on every access to the page containing *V*. Note that debuggers aren't compilers, and they don't require you to recompile your program to put in guard code. Saying that "exceptions are thrown" when the variable is accessed is just rephrasing the question, so it doesn't count. (*What* triggers the exception?)

5) [4] Give an example of a scenario where memory-mapped I/O makes more sense than programmed I/O, and *why* memory-mapped would be better. Then give an example of the opposite case.

A typical use of memory-mapped I/O is a video framebuffer since the framebuffer size is static and we're modifying entries in place. A typical use of PIO is short data transfers for character-mode devices or for accessing device registers that are used to synthesize a stream of data when sampled over time. Note that speed is not the main issue.

6) [4] To successfully prevent user programs from causing damage to other programs or the OS, hardware support is required. Name **two** hardware mechanisms in modern CPU's that supports this goal, and for each one, describe what specific kind(s) of damage it prevents.

Virtual memory, supported by the MMU, prevents processes from stomping on or reading each others' data. Privilege levels prevent user processes from executing certain instructions or code blocks that are reserved exclusively to the OS for inter-process resource management.

7) [4] Why would some OS's support multiple page sizes instead of just one page size? What additional page-management issues does this raise?

Using very large pages for OS code and shared read-only data reduces the number of page table entries and TLB entries that must be used to manage them. (Saying "reduces internal fragmentation" is not enough, since that's a basic principle of pages. The issue is *what situations* would benefit from this property.) However, it requires keeping track of whether pages "overlap", i.e. whether a given frame is part of a larger or a smaller page (some processors solve this by forbidding overlap, so that a single region can either be one big page or chopped up into small pages, but not both), and complicates physical address computation (the number of bits to use for page number and offset will depend on page length), page table walking and TLB lookup.

8) [4] Describe the mechanism of *priority inheritance* and give an example of the kind of problem it's intended to solve.

Consider threads 1, 2, 3 with decreasing priorities. 3 is running but has a resource needed by 1 (so 1 is blocked and 3 is about to release the resource so 1 can run). But 2 is on the ready queue, so 2 is scheduled and pre-empts 3. Now 1 cannot proceed because the resource it needs is still held by a lower-priority but not-running thread (3). **Note:** you need 3 threads to illustrate this.

9) [3] With respect to remote procedure calls (RPC), what is serialization (or marshalling) and why is it necessary? Are there cases where it is unnecessary?

Marshalling involves converting procedure call arguments (or return values) into a common representation for communication between machines that may have different architectures; for example, the packing of strings and the representation of floating point and integer values may differ between two machines, so marshalling converts them to a common intermediate form. It also follows pointers from data structures to collect all logical data structure members into a contiguous representation. If it is known that the representations of data structures and packing semantics are identical on both machines (which is a stronger requirement than just having the two machines use the same ISA), the first step can be omitted but the second is still needed.

10) [3] Describe one failure mode that might occur if a non-preemptive scheduler is used, and how it would be avoided with a preemptive scheduler.

One example: a program goes into an infinite loop and never does a blocking I/O or other operation that would yield the CPU. A preemptive scheduler would regain control on the next timer interrupt.

11) [3] A particular email message you're sending is so sensitive that you wish to both encrypt and sign it (both using public-key cryptography). Under what circumstances, if any, would you encrypt it first and then sign it? Under what circumstances, if any, would you sign it first and then encrypt it? (In other words, what's the practical effect of doing it one way vs. the other?)

Encrypt first, then sign, if it's OK for others to know who the sender is. Sign first, then encrypt, if you want only the designated receiver to be able to verify that you are the sender.

12) [3] True or false: all side-effect-free operations are idempotent, but not all idempotent operations are side-effect-free. Explain your answer concisely but completely (i.e. if true, explain why each part is true; if false, explain which part(s) are false and/or give counterexamples).

True. If an operation has no side effects, by definition executing it once is the same as executing it many times. On the other hand, an operation such as setting a variable to a specific value *does* have a side effect, yet is still idempotent.

13) [2] To avoid replay attacks, one can either use a randomly-generated nonce or a physical timestamp. Give one advantage of using a nonce over a timestamp, and one advantage of using a timestamp over a nonce. (You may assume that the resolution of physical timestamps is sufficient to avoid timestamp value collisions.)

Nonces better because synchronized clocks aren't required. Timestamps better because you don't have to remember every nonce you've ever seen.

14) [2] Describe one type of file access control that can be performed with access-control lists (such as AFS uses) but cannot be performed with file permissions (such as traditional Unix filesystems use).

One example: I can delegate (or revoke) write permission to a subset of users who are not all part of a common administrative group; you could allow everyone except one user to access a file.

15) [2] You're asked to take an existing Java program and rewrite it in C++. What benefit would you *gain* by doing so? What benefit, if any, would you *lose* by doing so?

Gain: C++ is generally a lot faster than Java since it's not interpreted. Lose: Java is type-safe, C++ isn't. Note, you lose *binary* portability but not necessarily *source* portability. If you claimed portability as your only lost property, you had to specify *binary* portability.

16) [2] What's the difference between a credential and a capability?

A credential proves that something (or someone) is what it says it is; a certificate signed by a trusted certificate issuer is an example. A capability allows the holder (who may be anyone) to take the particular action or use the particular resource specified by the capability, i.e. it is not specific to a principal.

17) [2] What's the difference between thrashing and deadlock?

Thrashing: processes can in theory make progress, but their aggregate resource needs exceed the ability of the scheduler; more time is spent shuffling resources than doing work, so processes make little or no forward progress. Deadlock: Processes cannot make progress even in principle since there is a loop in their logical waits-for resource graph.

18) [2] Give one example of how a filesystem might become corrupted, ***other than corruption of data in the files themselves***

The directory entry pointing to an inode or other metadata block may be lost. Or, the file metadata may not match the file's actual characteristics (e.g., length in bytes) because a failure occurred after the file was updated but before the metadata was updated. Trashed “file descriptors” don’t count, since they are not part of the filesystem and can usually be destroyed without resulting in filesystem inconsistency (though it may result in lost writes). Similarly, synchronization-related race conditions between clients writing a file do not damage the filesystem, they just leave it in a consistent-but-wrong (from program’s point of view) state.

19) [2] Your C program has a bug that causes it to accidentally dereference a nonexistent array element, e.g. a[10] where array a[] has been statically declared as containing 8 elements. What happens when this bug occurs at runtime, and why?

You get a segmentation fault or illegal access fault, *from the OS.* C++ doesn’t have runtime bounds checking so *you don’t get an array bounds exception.*

THE END

Systems Comp Fall 04 Results

Max: 60 Mean: 42.9 Median: 42 **Pass: 36 (60%)**

Regrade policy: You must provide a written explanation of where you think the grading error was made. Entire exam will be regraded, which may result in either raising or lowering total score.

Magic Number	Score	pass/fail
29	47	PASS
47	41	PASS
20	45	PASS
30	56	PASS
59	42	PASS
33	38	PASS
15	36	PASS
16	36	PASS
10	36	PASS
4	38	PASS
8	41	PASS
23	47	PASS
19	38	PASS
45	57	PASS
74	45	PASS
27	46	PASS
42	48	PASS
14	33	FAIL
9	45	PASS