Comprehensive Exam: Algorithms and Concrete Mathematics

Autumn 2002

This is a one hour closed-book exam and the point total for all questions is 100.

1. (*10 points*) Let $f(n) = 2^{\sqrt{\log n}}$ and $g(n) = n^{\epsilon}$, for some constant ϵ. Which one of the following claims is true:

   (a) $f(n) = O(g(n))$
   (b) $f(n) = o(g(n))$
   (c) $f(n) = \Omega(g(n))$
   (d) $f(n) = \omega(g(n))$

   Prove your answer.

2. (*15 points*) Prove a tight asymptotic bound on the behavior of $T(n)$.
   $T(n) = T(\lceil 0.4n \rceil) + T(\lceil 0.5n \rceil) + n$, where for $k \leq 10$ we are given that $T(k) = \Theta(1)$.
   Do not disregard the "ceiling" operations, i.e. provide a derivation that takes them into account or explicitly shows that they do not change the result.

3. (*20 points*) Prove that the following algorithm generates a random permutation of the numbers $1, 2, \ldots, n$:

   ```
   Set A[i] <- i;
   For i := 1,2,...,n :
      j <- Random uniformly distributed integer in range [i,n];
      Swap(A[i],A[j]);
   endfor
   ```

   In particular, prove for all i, j that $\Pr(A[i] = j) = 1/n$.

4. (*20 points*) You are given a graph $G(V, E)$ with n nodes and two length functions: $l_1 : E \to \{1, 2, \ldots, n\}$ and $l_2 : E \to \{1, 2, \ldots, n\}$.

   Given two integers a and b, and given two nodes s and t, give an algorithm to check if there is a path from s to t whose length according to the function l_1 is at most a, and whose length according to the second function l_2 is at most b. Prove the running time of your algorithm and show that it is polynomial in n. [Hint: make use of the fact that lengths are integers in a polynomial range.]

5. (*15 points*) A dominating set in a graph $G(V, E)$ is a set of vertices $S \subseteq V$ such that every vertex in $V - S$ is adjacent to at least one vertex in S.

   Consider the following greedy algorithm for dominating set: Find the vertex with largest degree. Include it in the dominating set, and delete it and all its neighboring vertices from the graph. Again find the vertex with largest degree in the remaining graph, include it in the dominating set, delete it and its neighbors from the graph, and so on. Repeat till the graph is empty.

   Show that there exist graphs with n vertices that have $O(\sqrt{n})$-size optimum dominating set and where the above algorithm produces a (much worse) $O(n)$-size dominating set. Partial credit for any other graph families that show that the above algorithm is not good.

6. (*20 points*) Suppose you are allowed only equality comparisons among objects. In other words, there is no ordering defined over the objects; you can only check if two objects are equal or not. You are given n objects in an array. Show an $O(n)$-time deterministic algorithm to find out if there is an object appearing strictly more than $n/2$ times in the array. For simplicity, you may assume n is a power of 2. 10 points for a randomized algorithm.

Stanford University
Computer Science Department

Fall 2002 Comprehensive Exam in Artificial Intelligence

1. **Closed Book:** no notes, textbooks, laptops, Internet access, etc.
2. **Write only in the Blue Books: No credit** for answers written on these exam pages.
3. **Write Magic number** on the cover of **EACH blue book**.
4. **The exam is timed for one hour.**

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.

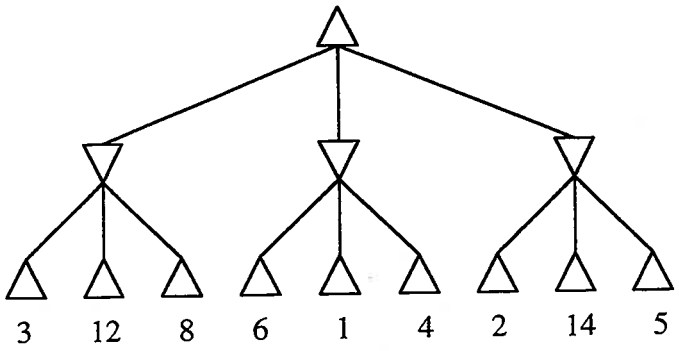
B. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures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

2002 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) Use Minimax search to solve this problem. Write the backed up values next to the nodes. What is the best first move for the maximizing player (left middle, or right)? What is the best response for the minimizing player (left middle, or right)?

(b) Now solve the problem using α-β pruning search. Mark the nodes that are *not* evaluated by the algorithm.

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

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

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

(c) If $\Gamma \models \varphi$ and $\Delta \models (\varphi \Rightarrow \psi)$, then $\Gamma \cup \Delta \models \psi$.

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

(e) If $\Gamma \models \varphi$ and $\Delta \models \neg\varphi$, then $\Gamma \cup \Delta \models (\varphi \Rightarrow \neg\varphi)$.

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

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

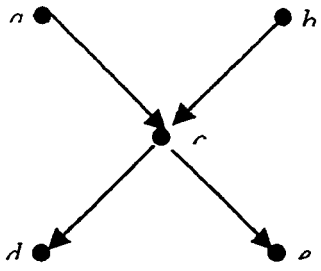
Note that this is a question about resolution. You will get zero points for proving it in any other way.

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.

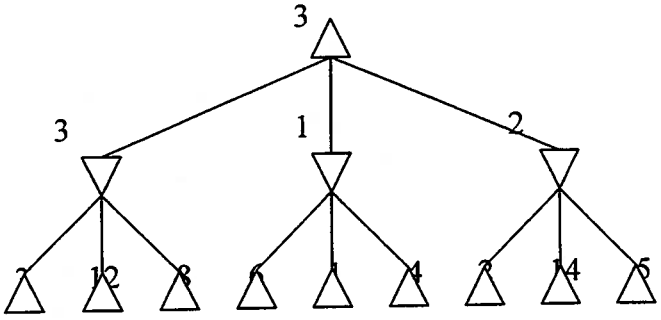
5. Belief Networks. (20 points) Consider the belief network shown below.



What is the joint probability of *-a*, *-b*, *c*, *d*, *e*? You may assume that p(a)=0.1, p(b)=0.2, p(cla,b)=0.9, p(cla,-b)=0.9, p(cl-a,b)=0.3, p(cl-a,-b)=0.1, p(dlc)=0.9, p(d,-c)=0.1, p(e,c)=0.7, p(e,-c)=0.1.

2002 Comprehensive Examination
Artificial Intelligence

1. Search.



(a) Left is the best move for the maximizer. Left is the best response for the minimizer.
(b) Unexpanded nodes are those with values 4, 14, and 5.

2. Logic.
(a) True
(b) False
(c) True
(d) False
(e) True

3. Automated Reasoning.

1. {p(x,f(x))}
2. {p(f(x),x)}
3. {-p(x,y), -p(y,z), p(x,z)}
4. {-p(c,c)}
5. {-p(c,y),-p(y,c)}
6. {p(f(c),c)}
7. {}

4. Probability.

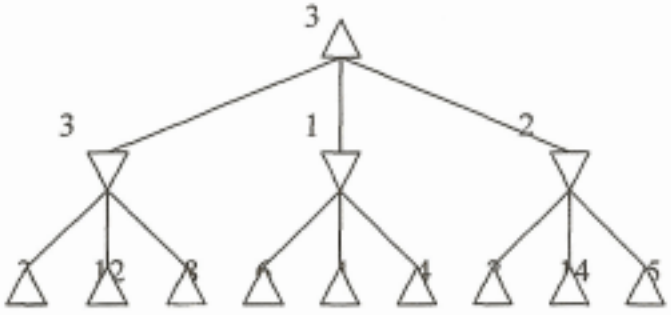
$$p(b1|r)=p(r|b1)*p(b1)/p(r)$$
$$=2/11*1/3 \ / \ (2/11+4/9+3/10)*1/3$$
$$=0.1963$$

5. Belief Networks.

$$p(-a, -b, c, d, e)=p(-a)*p(-b)*p(c|-a,-b)* p(d|c)*p(e,c)$$
$$= 0.9*0.8*0.1*0.9*0.7$$
$$= 0.04536$$

2002 Comprehensive Examination
Artificial Intelligence

1. Search.



(a) Left is the best move for the maximizer. Left is the best response for the minimizer.
(b) Unexpanded nodes are those with values 4, 14, and 5.

2. Logic.
(a) True
(b) False
(c) True
(d) False
(e) True

3. Automated Reasoning.

1. {p(x,f(x))}
2. {p(f(x),x)}
3. {-p(x,y), -p(y,z), p(x,z)}
4. {-p(c,c)}
5. {-p(c,y),-p(y,c)}
6. {p(f(c),c)}
7. {}

4. Probability.

$$p(b1|r)=p(r|b1)*p(b1)/p(r)$$
$$=2/11*1/3 \ / \ (2/11+4/9+3/10)*1/3$$
$$=0.1963$$

5. Belief Networks.

$$p(-a, -b, c, d, e)=p(-a)*p(-b)*p(c|-a,-b)* p(d|c)*p(e,c)$$
$$= 0.9*0.8*0.1*0.9*0.7$$
$$= 0.04536$$

Comprehensive Exam: Autumn 2002-03

Automata and Formal Languages (60 points)

Problem 1. [10 points]

Recall that a string x is called a *substring* of another string w if x appears consecutively within w (i.e. $w = yxz$ for some strings y,z).

1. [3 points] Give a regular expression for the language $L = \{ w \in \Sigma^* \mid$ the string *papi* is a substring of $w\}$, where Σ is the English alphabet.

2. [7 points] Give a deterministic finite automaton for L. (Partial credit for a nondeterministic finite automaton if you cannot get a DFA.)

Problem 2. [10 points]

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

1. Nondeterministic and deterministic finite automata recognize the same set of languages.

2. Nondeterministic and deterministic pushdown automata recognize the same set of languages.

3. Nondeterministic and deterministic Turing machines recognize the same set of languages.

4. The intersection of two recursive languages is recursive.

5. The intersection of two context-free languages is context-free.

Problem 3. [15 points] Classify each of the following languages as being in one of the following classes of languages: *regular, context-free, recursive, recursively enumerable*. You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the appropriate class for the language of a PDA is context-free. You will receive 3 points for each correct answer and –2 points for each incorrect answer.

1. Call a string *w* over the alphabet $\Sigma = \{s,r\}$ *well-formed* if every prefix of *w* contains at least as many occurrences of letter *s* as of letter *r*. What is the appropriate class for the language of well-formed strings ?

2. The language of well-formed strings *w* over $\Sigma = \{s,r\}$ such that in each prefix of *w* the number of occurrences of *s* does not exceed the number of occurrences of *r* by more than 10.

3. The language of strings over {*s,r,s',r'*} that are well-formed with respect to both pairs (*s,r*) and (*s',r'*), i.e. each prefix contains at least as many occurrences of *s* as of *r*, and at least as many occurrences of *s'* as of *r'*.

4. The set of encodings of Turing machines *M* whose time complexity is not bounded by n^2; i.e., $L = \{ M \mid$ there exists an input string *w* such that *M* performs more than $|w|^2$ steps on input *w*$\}$

5. The language $L = \{w \in \Sigma^* \mid \exists x \in L_1, \exists y \in L_2 \text{ such that } x=wy\}$, where L_1 is regular and L_2 is recursively enumerable.

Problem 4. [12 points]
Classify each of the following problems as being in one of the following classes: *P (polynomial-time solvable), decidable but not known to be in P* (this class includes eg. NP-complete and PSPACE-complete problems), *undecidable.* You will receive 3 points for each correct answer and –2 points for each incorrect answer.

1. *Input*: Deterministic finite automata *A, B.* *Question*: $L(A) \subseteq L(B)$?

2. *Input*: Pushdown automaton *A*, deterministic finite automaton *B.* *Question*: $L(A) \subseteq L(B)$?

3. *Input*: Deterministic finite automaton *A*, pushdown automaton *B.* *Question*: $L(A) \subseteq L(B)$?

4. *Input*: Nondeterministic finite automata *A, B.* *Question*: $L(A) \subseteq L(B)$?

Problem 5. [13 points]
Prove that the FEEDBACK NODE SET problem is NP-complete, using the fact that the NODE COVER problem is NP-complete.
The definitions of these problems are recalled below.

FEEDBACK NODE SET
Input: A directed graph *H* and a positive integer *k.*
Question: Is there a set *F* of at most *k* nodes such that removing from the graph the nodes of *F* and their incident edges leaves an acyclic graph ? (Such a set *F* is called a *feedback node set* of *H).*

NODE COVER
Input: An undirected graph *G* and a positive integer *k.*
Question: Is there a set *C* of at most *k* nodes such that every edge of *G* is incident to at least one node of *C* ? (Such a set *C* is called a *node cover* of *G.*)

Comprehensive Exam: Autumn 2002-03

Automata and Formal Languages (60 points)
Sample Solutions

Problem 1. [10 points]

Recall that a string x is called a *substring* of another string w if x appears consecutively within w (i.e. $w = yxz$ for some strings y,z).

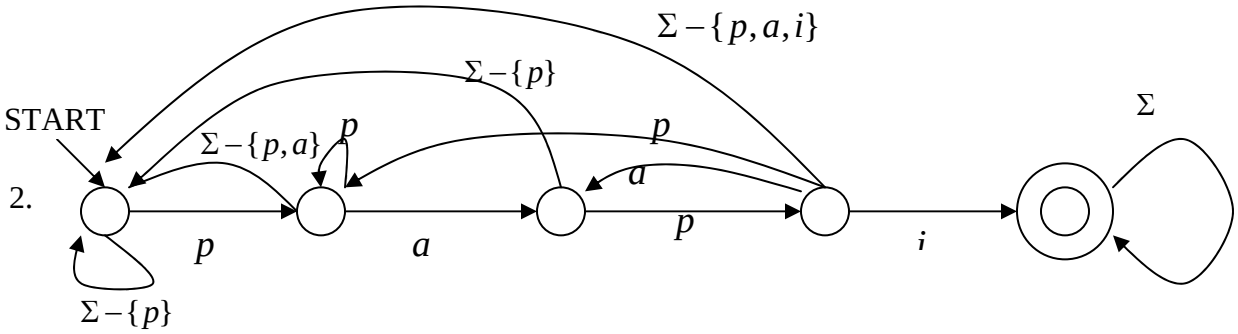
1. [3 points] Give a regular expression for the language $L = \{ w \in \Sigma^* \mid$ the string *papi* is a substring of $w\}$, where Σ is the English alphabet.

2. [7 points] Give a deterministic finite automaton for L. (Partial credit for a nondeterministic finite automaton if you cannot get a DFA.)

Solution:

1. $\Sigma^* papi \Sigma^*$

2.



Problem 2. [10 points]

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

1. Nondeterministic and deterministic finite automata recognize the same set of languages.

2. Nondeterministic and deterministic pushdown automata recognize the same set of languages.

3. Nondeterministic and deterministic Turing machines recognize the same set of languages.

4. The intersection of two recursive languages is recursive.

5. The intersection of two context-free languages is context-free.

Solution:

1. True

2. False

3. True

4. True

5. False

Problem 3. [15 points] Classify each of the following languages as being in one of the following classes of languages: *regular, context-free, recursive, recursively enumerable*. You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the appropriate class for the language of a PDA is context-free. You will receive 3 points for each correct answer and –2 points for each incorrect answer.

1. Call a string w over the alphabet $\Sigma = \{s,r\}$ *well-formed* if every prefix of w contains at least as many occurrences of letter s as of letter r. What is the appropriate class for the language of well-formed strings ?

2. The language of well-formed strings w over $\Sigma = \{s,r\}$ such that in each prefix of w the number of occurrences of s does not exceed the number of occurrences of r by more than 10.

3. The language of strings over $\{s,r,s',r'\}$ that are well-formed with respect to both pairs (s,r) and (s',r'), i.e. each prefix contains at least as many occurrences of s as of r, and at least as many occurrences of s' as of r'.

4. The set of encodings of Turing machines M whose time complexity is not bounded by n^2; i.e., $L = \{ M \mid$ there exists an input string w such that M performs more than $|w|^2$ steps on input $w\}$

5. The language $L = \{w \in \Sigma^* \mid \exists x \in L_1, \exists y \in L_2 \text{ such that } x{=}wy\}$, where L_1 is regular and L_2 is recursively enumerable.

Solution:

1. context-free
2. regular
3. recursive
4. recursively enumerable
5. regular

Problem 4. [12 points]
Classify each of the following problems as being in one of the following three classes: *P (polynomial-time solvable), decidable but not known to be in P* (this class includes eg. NP-complete and PSPACE-complete problems) , *undecidable.* You will receive 3 points for each correct answer and –2 points for each incorrect answer.

1. *Input*: Deterministic finite automata *A, B.* *Question*: $L(A) \subseteq L(B)$?
2. *Input*: Pushdown automaton *A,* deterministic finite automaton *B.* *Question*: $L(A) \subseteq L(B)$?
3. *Input*: Deterministic finite automaton *A,* pushdown automaton *B.* *Question*: $L(A) \subseteq L(B)$?
4. *Input*: Nondeterministic finite automata *A, B.* *Question*: $L(A) \subseteq L(B)$?

Solution:

1. P
2. P
3. undecidable
4. decidable but not known to be in P

Problem 5. [13 points]
Prove that the FEEDBACK NODE SET problem is NP-complete, using the fact that the NODE COVER problem is NP-complete.
The definitions of these problems are recalled below.

FEEDBACK NODE SET
Input: A directed graph H and a positive integer k.
Question: Is there a set F of at most k nodes such that removing from the graph the nodes of F and their incident edges leaves an acyclic graph ? (Such a set F is called a *feedback node set* of H).

NODE COVER
Input: An undirected graph G and a positive integer k.
Question: Is there a set C of at most k nodes such that every edge of G is incident to at least one node of C ? (Such a set C is called a *node cover* of G.)

Solution:

First show that the FEEDBACK NODE SET problem is in NP. To see this, note that we can guess a set F of at most k nodes, remove them from the graph along with their incident edges, and then check in polynomial time that the resulting graph is acyclic.

To show that the FEEDBACK NODE SET problem is NP-hard, we give a polynomial time reduction from the NODE COVER problem. Given an instance (G,k) of the NODE COVER problem, we construct an instance (H,k) of the FEEDBACK NODE SET problem as follows. The directed graph H has exactly the same nodes as G. For each undirected edge $[u,v]$ of G, the directed graph H has two directed edges $u \to v$ and $v \to u$. We claim that G has a node cover of size k if and only if H has a feedback node set of size k.

(if) Suppose that H has a feedback node set F of size k. Note that for each edge $[u,v]$ of G, the corresponding two directed edges $u \to v$ and $v \to u$ that were introduced in H form a cycle. Therefore, for each edge $[u,v]$ of G, the feedback node set F of H must contain at least one of the nodes u,v. Thus, the set F is a node cover of G.

(only if) Suppose that G has a node cover C of size k. Then every directed edge of H is incident to at least one node of C. Therefore, removing the set C of nodes and their incident edges from H, will leave a graph with no edges, which is of course acyclic. Thus, C is a feedback node set of H.

COMPILERS COMPREHENSIVE EXAM
Autumn 2002

This is a 30 minute, closed book exam. Please mark your answers in the blue book.

1) Give a context free grammar that will describe the set of strings that are palindromes over the alphabet {0, 1, ..., 9} (2 pts)

2) Given the following code:

```
1: #include <stdio.h>
2:
3: int main () {
4:   int foo [10];
5:   foo++;
6:   printf ("Hello world!);
7:   return 7
8: }
```

Clearly this code has some errors in it. Assuming each error in the code would be reported by the compiler (and that one error would not affect the next), what lines would cause the compiler to error and which stage of compilation would detect the error (lexing, syntax or semantic analysis)? Justify your responses. (4 pts)

3) Is the following grammar LL(1)? If not, can the language that it describes be expressed with an LL(1) grammar? Justify your response. (7 pts)

S --> Tu | wx
T --> Sq | vvS

4) In one or two sentences only, briefly describe why the class of LR(1) grammars can describe more languages than the class of LL(1) grammars. (8 pts)

5) Sometimes the basic compiler optimizations can actually slow down the code being compiled. Write a snippet of code that demonstrates how the application of simple compiler opimizations may cause the generated code to be slower than the "unoptimized" version. Justify your response. (9 pts)

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.

By placing your number below you indicate that you agree in the spirit of the Stanford Honor Code to neither give nor receive unpermitted aid on this exam.

Number: _____

Problem	Max Score	Your Score
1	48	
2	25	
3	27	
TOTAL	100	

Problem 1: Short Answer [4 points each, 48 points total]

A. Cache A is an 8K-byte direct-mapped cache. Cache B is a 16K-byte four-way set associative cache. Cache C is a 32K-byte four-way set associative cache. All caches have sixteen-byte lines, start in the same initial state of all lines invalid, use an LRU replacement policy, and see the same sequence of memory references. After this sequence of references, which of the following statements are true. Circle all that apply.

(a) Cache B will always contain every line in cache A
(b) Cache C will always contain every line in cache B
(c) Cache C will always contain every line in cache A

B. Compared to an 8K-byte direct-mapped cache, what type of misses will a 16K-byte direct-mapped cache have fewer of? ? Circle all that apply.

(a) compulsory
(b) conflict
(c) capacity

C. Adding a cache memory to a system changes which of the following memory parameters? Circle all that apply and denote the direction of change with an up arrow or a down arrow.

(a) Memory latency
(b) Memory bandwidth
(c) Memory address space
(d) Memory reliability

D. In a 32K-byte four-way set-associative cache with 32-byte blocks, how large is the index field used to address the cache array? (write down the number of bits)

_____

E. If the cache of question D is physically tagged and physical addresses are 40-bits long, what is the minimum possible length the tag may be for correct operation? (write down the number of bits)

_____

F. Which instruction set will give the best code density?

(a) An accumulator instruction set
(b) A RISC instruction set with fixed 32-bit instruction formats
(c) A CISC instruction set with variable-length instruction formats
(c) A stack instruction set

G. Which mean should be used to combine execution times of programs? Circle all that apply.

(a) Arithmetic mean
(b) Geometric mean
(c) Harmonic mean

H. In the steady state, what will be the prediction accuracy of a two-bit branch predictor on the sequence TTTTNTTTTN (T=taken, N=not taken)?

I. Which will have better throughput, a machine that uses scoreboarding (with no bypassing) to avoid read-after-write hazards, or a machine that uses bypassing to avoid these hazards?

(a) Scoreboarding with no bypassing
(b) Bypassing

J. A processor has a six-stage pipeline with stages Fetch, Decode, Register Read, Execute, Memory, Register Write and is able to execute one instruction per cycle in the absence of branches. All branches are predicted as not taken. When a branch is taken, the condition and address are both generated during the Execute pipeline stage. If conditional branches account for 10% of all instructions and 30% of conditional branches are taken, what will be the execution rate of this processor? Express your answer in clocks per instruction (CPI).

_____

K. A machine with register renaming is able to reorder instructions without regard to what type of dependencies? Circle all that apply.

(a) Data dependencies
(b) Output dependencies
(c) Anti-dependencies
(d) Control dependencies

L. An instruction for protected subsystem entry (sometimes called system call) must change which subset of the following five things atomically? (circle all that apply)

(a) The contents of a data register
(b) The program counter (sometimes called instruction pointer)
(c) The contents of the page table
(d) The contents of the cache
(e) The privilege level

Problem 2: Pipeline Architecture [25 points total]

Suppose you have a CPU with a 6 stage pipeline containing the following stages:

F: instruction fetch
D: instruction decode
R: register read (and branch address calculation)
A: execute ALU operations (including determination of branch condition)
M: memory load and store
W: write back to register file

The register file cannot read a value that is being written in the same cycle. The machine has bypass paths from the output of the A, M, and W stages to both inputs of the ALU.

A. [5 points] What is the latency of an unconditional branch on this machine? Assume no prediction or speculation is employed. (Note: Latency is defined as the number of cycles from when this instruction is executed until the next instruction is executed. E.g., the latency of a simple instruction with no dependencies is one.)

B. [5 points] What is the latency of a conditional branch? Again assume no speculation about branch direction or distance.

C. [5 points] The instruction register at each pipeline stage is denoted by IR_{stage}; for example IR_A is the instruction register at the ALU stage. The source and destination fields of the IR are denoted IR.A (source 1), IR.B (source 2), and IR.C (the destination). For example, the destination field of the IR at the M stage is IR_M.C. Each IR also has a valid field IR.V that indicates when it contains a valid instruction. Using this notation, write a logical expression that indicates when the bypass path from the output of the M stage to the first (A) input of the A stage should be activated.

D. [10 points] Consider the following instruction sequence:

```
Brz                     r5, EXIT
Add                     r1 <- r5 + r6
Load                    r2 <- [r1 + 4]
Add                     r4 <- r2 + r3
```

Assuming that the branch is not taken and that no instructions are fetched speculatively, how many cycles does this take to execute from the time the IP points to the branch instruction until the result of the add is written to the register file? Explain your answer. You may want to draw a timeline.

Problem 3. Instruction Issue [27 points total]

Consider the following instruction sequence:

```
1     LD    X,R1
2     ADD   R1,#5,R2        ; R2 <- R1 + 5
3     LD    Y,R3
4     ADD   R2,R3,R4        ; R4 <- R2 + R3
5     LD    Z,R5
6     ADD   R1,R5,R6        ; R6 <- R1 + R5
7     MUL   R6,R4,R7        ; R7 <- R6 * R4
8     ADD   R7,#3,R8        ; R8 <- R7 + 3
```

You may assume that all arithmetic operations have two-cycle latency and all loads have four-cycle latency. That is, the result of an arithmetic (memory) operation is available two (four) cycles after that operation enters the first execution stage of the machine. Also assume that there is full bypassing, that all loads hit in the cache, and that an arbitrary number of loads (and arithmetic operations) can be in flight at a single time. Hint: you need only consider the execution and memory stages of the pipeline to answer this question.

A. [7 points] How many cycles does this sequence take to execute on a single-issue in-order machine? Measure time from the cycle that the first instruction issues (enters the execution stage) to the cycle in which last instruction completes. For example, the load of instruction 1 issues in cycle 1 and completes in cycle 4. This enables the ADD of instruction 2 to issue in cycle 5 and complete in cycle 6. Show your work.

B. [7 points] If issue order and resources are not limited, what is the shortest time this sequence of instructions could take? Show your work.

C. [7 points] How many cycles does this sequence take to execute on an in-order machine with multiple issue (assume an issue width as wide as you need)? Show your work

D. [6 points] Can you statically reorder the code to give the wide in-order machine of part C the performance bound of part B? If so, show the new ordering.

Stanford University
Computer Science Department

Fall 2002 Comprehensive Exam in Databases

1. **OPEN Book & Notes:** no laptops, Internet access, etc.
2. **Write only on the answer sheet provided.**
3. **The exam is timed for one hour.**
4. **Write your Magic Number on this sheet & on the Exam paper.**

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.

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

Stanford University Computer Science Department
2002 Comprehensive Exam in Databases

- The exam is *open book and notes*.
- There are 7 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	TOTAL
Max. points	6	12	6	10	6	8	12	60
Points								

1. **(6 points)** Consider an Entity-Relationship diagram with a 3-way relationship R connecting entity sets A, B, and C. In the E/R diagram, there are arrows entering B and C from R, but no arrow entering A. Suppose the number of entities currently in entity sets A, B, and C are a, b, and c, respectively. Also suppose that there are currently r triples in the relationship set for R.

   (a) It must be that $r \leq ab$. Explain why.

   (b) What other nontrivial constraints on values a, b, c, and r must hold?

2. **(12 points)** Suppose we have a relation $R(A, B, C, D, E, F, G, H, I, J)$ with the following nine functional dependencies:

$$
\begin{array}{rcl}
FI & \rightarrow & C \\
AC & \rightarrow & I \\
EJ & \rightarrow & H \\
GI & \rightarrow & E \\
CJ & \rightarrow & E \\
DE & \rightarrow & B \\
EF & \rightarrow & H \\
BE & \rightarrow & A \\
IJ & \rightarrow & D
\end{array}
$$

   (a) Compute the closure $(CFGJ)^+$, that is, the set of attributes that must be equal in any two tuples that agree on all of C, F, G, and J.

(Problem continues on next page)

(b) Three of the ten attributes must be in any key. Which three are they, and why? (There is a brief, simple, explanation that does not require you to find all the keys.)

(c) Find the smallest key, and explain why there is no other key of *the same size* or smaller.

3. **(6 points)** Suppose relation $R(A, B, C, D, E)$ satisfies the multivalued dependencies $A \twoheadrightarrow B$ and $B \twoheadrightarrow D$. Also suppose that R contains the tuples $(0, 1, 2, 3, 4)$ and $(0, 5, 6, 7, 8)$. List the tuples in the smallest possible relation that R could be.

4. **(10 points)** In the following, all outerjoins should be assumed "natural."

   (a) If we use the set model of relations, then it is possible to express the full outerjoin of two relations as the union of the left-outerjoin and right-outerjoin of these two relations. Explain why this formula produces the full outerjoin.

   (b) Show that the the formula from (a) does not work if we use the bag (multiset) model of relations.

   (c) For the bag model, is there any way to express the full outerjoin in terms of the left- and right-outerjoin and other operations of relational algebra? Either give such a formula or show that none exists.

5. **(6 points)** Consider relations $R(A, B, C)$ and $S(A, B, C)$ with no assumptions about keys. For the following two statements using relational algebra, state the constraint or property that is specified by the given statement. Please be specific about the actual attributes involved. In both cases the correct answer corresponds to a specific database concept and can be stated in a few words. Much longer answers will be considered incorrect.

   Note: Symbol $\ltimes$ is the *semijoin* operator. $E_1 \ltimes E_2 \equiv \pi_{schema(E_1)}(E_1 \bowtie E_2)$.

   (a) $\pi_A(R) \ltimes \pi_A(S) = \pi_A(R)$

   (b) $\sigma_{B1 \neq B2}(\rho_{R1(A,B1,C1)}(R) \bowtie \rho_{R2(A,B2,C2)}(R)) = \emptyset$

6. **(8 points)** Consider the following relations in a SQL database:

```
PhDstud(name,year,email)   // name is a key
Aligned(student,adviser)   // student is a foreign key
                              referencing PhDstud.name
```

   Suppose the owner (creator) of these relations is a user named "Kathi," and Kathi wants to grant to a user named "Rajeev" the ability to read all information about all first-year PhD students who are not aligned with an adviser (and only those students). Specify a command or sequence of commands that achieves this goal.

7. **(12 points)** Consider a relation `Visit(inspector,restaurant,date)` recording visits inspectors make to certain restaurants on certain dates. Assume no more than one inspector visits any given restaurant on a given date. Values for the `date` attribute can be compared using =, <, <=, etc., and aggregate functions `min`, `max`, and `count` may be applied to the `date` column. Write a SQL query that returns the two most recent visit dates to each restaurant.

   The result of your query should be an ordered two-column relation, where each restaurant appearing in the original `Visit` relation has two adjacent tuples, with the more recent visit for each restaurant listed first. If a restaurant has only one visit, your result should still include a single tuple for that restaurant. For example, we might see the following result, where "Thai Cafe" has only one visit:

restaurant	date
Bytes	4/1/02
Bytes	3/31/02
Coffee House	9/15/01
Coffee House	8/1/00
Thai Cafe	1/10/02
Tresidder	3/10/01
Tresidder	3/9/01

   Your SQL query will be graded on simplicity as well as correctness.

Stanford University Computer Science Department
2002 Comprehensive Exam in Databases – SAMPLE SOLUTION

1. **(6 points)** Consider an Entity-Relationship diagram with a 3-way relationship R connecting entity sets A, B, and C. In the E/R diagram, there are arrows entering B and C from R, but no arrow entering A. Suppose the number of entities currently in entity sets A, B, and C are a, b, and c, respectively. Also suppose that there are currently r triples in the relationship set for R.

   (a) It must be that $r \leq ab$. Explain why.

   Answer: The arrow into C tells us that for any A-B pair, there can be at most one associated C in the R relationship set. Thus, there can be at most ab triples in that set.

   (b) What other nontrivial constraints on values a, b, c, and r must hold?

   Answer: $r \leq ac$ holds as well. Nothing else that does not follow from $r \leq ab$ and $r \leq ac$ holds.

2. **(12 points)** Suppose we have a relation $R(A, B, C, D, E, F, G, H, I, J)$ with the following nine functional dependencies:

$$\begin{array}{rcl} FI & \rightarrow & C \\ AC & \rightarrow & I \\ EJ & \rightarrow & H \\ GI & \rightarrow & E \\ CJ & \rightarrow & E \\ DE & \rightarrow & B \\ EF & \rightarrow & H \\ BE & \rightarrow & A \\ IJ & \rightarrow & D \end{array}$$

   (a) Compute the closure $(CFGJ)^+$, that is, the set of attributes that must be equal in any two tuples that agree on all of C, F, G, and J.

   Answer: $CEFGHJ$.

   (b) Three of the ten attributes must be in any key. Which three are they, and why? (There is a brief, simple, explanation that does not require you to find all the keys.)

   Answer: F, G, and J must be in every key, since they do not appear on the right side of any FD.

(c) Find the smallest key, and explain why there is no other key of *the same size* or smaller.

Answer: $FGIJ$ is the smallest key. It is easy to check that its closure is all the attributes, so it is a superkey. Every key contains FGJ, but FGJ by itself is not a key — it is its own closure. Thus, at least 4 attributes are necessary, and therefore $FGIJ$ is a key, not just a superkey. We have only to check that there are no other 4-attribute keys. These could only be FGJ and one of A, B, C, D, E, or H. We could check each set, and see that its closure is less than all the attriubutes, but there is a quicker argument. Observe the first two FD's, and note that these are the only ways to add I or C to a closure. But each requires that the other be present in the closure already. Thus, only $CFGJ$ could be a 4-attribute key. But we ruled that out in part (a).

3. **(6 points)** Suppose relation $R(A, B, C, D, E)$ satisfies the multivalued dependencies $A \twoheadrightarrow B$ and $B \twoheadrightarrow D$. Also suppose that R contains the tuples $(0, 1, 2, 3, 4)$ and $(0, 5, 6, 7, 8)$. List the tuples in the smallest possible relation that R could be.

   Answer: The eight tuples described as follows. A is 0. B is 1 or 5. D is 3 or 7. C and E are either 2 and 4, respectively, or 6 and 8, respectively.

4. **(10 points)** In the following, all outerjoins should be assumed "natural."

   (a) If we use the set model of relations, then it is possible to express the full outerjoin of two relations as the union of the left-outerjoin and right-outerjoin of these two relations. Explain why this formula produces the full outerjoin.

   Answer: A tuple in the outerjoin is either (1) in the (inner) join, or (2) it is a padded tuple from the left argument, or (3) it is a padded tuple from the right argument. The left outerjoin gives us (1) and (2), while the right outerjoin gives us (1) and (3). Thus, their union gives us exactly (1), (2), and (3).

   (b) Show that the the formula from (a) does not work if we use the bag (multiset) model of relations.

   Answer: The tuples of the inner join (1) are counted twice when we take the bag union. Thus, as long as the inner join is nonempty, there will be more tuples in the union of the left- and right-outerjoins than in the full outerjoin.

   (c) For the bag model, is there any way to express the full outerjoin in terms of the left- and right-outerjoin and other operations of relational algebra? Either give such a formula or show that none exists.

   Answer: The union of the left- and right-outerjoins, minus the (inner) join has exactly the right number of occurrences of each tuple.

5. **(6 points)** Consider relations $R(A, B, C)$ and $S(A, B, C)$ with no assumptions about keys. For the following two statements using relational algebra, state the constraint or property that

is specified by the given statement. Please be specific about the actual attributes involved. In both cases the correct answer corresponds to a specific database concept and can be stated in a few words. Much longer answers will be considered incorrect.

Note: Symbol $\ltimes$ is the *semijoin* operator. $E_1 \ltimes E_2 \equiv \pi_{schema(E_1)}(E_1 \bowtie E_2)$.

(a) $\pi_A(R) \ltimes \pi_A(S) = \pi_A(R)$

Answer: Referential integrity from $R.A$ to $S.A$

(b) $\sigma_{B1 \neq B2}(\rho_{R1(A,B1,C1)}(R) \bowtie \rho_{R2(A,B2,C2)}(R)) = \emptyset$

Answer: Functional dependency $A \rightarrow B$ in R

6. **(8 points)** Consider the following relations in a SQL database:

```
PhDstud(name,year,email)   // name is a key
Aligned(student,adviser)   // student is a foreign key
                              referencing PhDstud.name
```

Suppose the owner (creator) of these relations is a user named "Kathi," and Kathi wants to grant to a user named "Rajeev" the ability to read all information about all first-year PhD students who are not aligned with an adviser (and only those students). Specify a command or sequence of commands that achieves this goal.

Answer:

```
create view Slackers as
  (select * from PhDstud
   where year = 1
    and name not in (select student from Aligned));
grant select on Slackers to Rajeev;
```

7. **(12 points)** Consider a relation `Visit(inspector,restaurant,date)` recording visits inspectors make to certain restaurants on certain dates. Assume no more than one inspector visits any given restaurant on a given date. Values for the `date` attribute can be compared using `=`, `<`, `<=`, etc., and aggregate functions `min`, `max`, and `count` may be applied to the `date` column. Write a SQL query that returns the two most recent visit dates to each restaurant.

The result of your query should be an ordered two-column relation, where each restaurant appearing in the original `Visit` relation has two adjacent tuples, with the more recent visit for each restaurant listed first. If a restaurant has only one visit, your result should still include a single tuple for that restaurant. For example, we might see the following result, where "Thai Cafe" has only one visit:

restaurant	date
Bytes	4/1/02
Bytes	3/31/02
Coffee House	9/15/01
Coffee House	8/1/00
Thai Cafe	1/10/02
Tresidder	3/10/01
Tresidder	3/9/01

Your SQL query will be graded on simplicity as well as correctness.

Answer:

```
select restaurant, date
from Visit V1
where (select count(*) from Visit V2
       where V2.restaurant = V1.restaurant
       and V2.date >= V1.date) <= 2
order by restaurant, date desc
```

Computer Graphics Comprehensive Exam

Computer Science Department
Stanford University
Fall 2002

MAGIC NUMBER: _____

Note: This 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.

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 the 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)

1. [Total: 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]. Image composition.

1B [5 points]. Diffuse reflection.

1C [5 points]. Key-frame animation.

1D [5 points]. Hierarchical modeling.

2. [20 points] Transformations.

Computer graphics relies heavily on 3D transformations. Most common are the geometric transformations such as rotations, translations, scales, etc. In the following, T(dx,dy,dz) refers to a translation by (dx,dy,dz), Rx(a) refers to a rotation about the x-axis by a degrees, Ry(a) and Rz(a) refer to rotations about the y- and z-axis, respectively.

The order of transformations may matter. Also, sometimes the order may be rearranged, but the arguments will change. Describe whether the following statements are true or false.

T(1,0,0) Rx(45) = Rx(45) T(1,0,0)?

T(1,0,0) T(0,2,0) = T(0,2,0) T(1,0,0)?

Rx(45) Ry(30) = Ry(30) Rx(45)?

Rx(45) Rx(30) = Rx(30) Rx(45)?

T(1,0,0) Rz(180) = T(-1,0,0) Rz(0)

Transformations have inverses. State whether the following formulas for the inverse transformations are true or false.

$Rz(45)^{-1} = Rz(-45)$?

$Rz(180)^{-1} = Rz(180)$?

$T(1,1,1)^{-1} = T(1,1,1)$?

$[Rz(45)\ T(1,0,0)]^{-1} = T(1,0,0)\ Rz(45)$?

$[Rx(45)\ Ry(30)]^{-1} = Ry(-30)\ Rx(-45)$?

3. [20 points] Ray tracing.

One of the most general methods for rendering is ray tracing. At the core of a ray tracer is a procedure to find ray-surface intersections. The inputs to the procedure are a ray and a formula for a surface, the output is the point of intersection.

Assume is ray is given by the following parametric equations:

$$x = x0 + t * x1$$
$$y = y0 + t * y1$$
$$z = z0 + t * z1;$$

(x0,y0,z0) is the origin of the ray, and (x1,y1,z1) is the direction of the ray. As t increases, the points on the ray move from the origin along the direction.

Work out a procedure for computing the point of intersection of a ray with a cone. A cone is given by the following implicit function $x^2+y^2-z^2 = 0$. In general, a ray intersects a cone in two points. Return the closest point in the direction of the ray.

4. [20 points] Polygonal Meshes.

Polygonal meshes are widely used in computer graphics. A polygonal mesh consists of vertices that define the positions of 3D points, and topological information that organize the points into polygons. There are many different types of mesh data structures, depending on the operations that need to be performed.

1. Describe a simple data structure to store a triangular mesh so that vertices are not duplicated. In your description, define C structs for vertices and triangles.

2. Refine the data structure described above so that neighboring triangles may be found in constant time.

5. [20 points] Rasterization.

In your new job as chief graphics software architect, you receive a request to write a procedure to draw the following curve:

$$y^2 = x^3$$

To simplify the problem, consider only positive y values. This procedure should draw n pixels by stepping along the curve starting at the origin. Each pixel should be under the curve, or on the curve. No pixels should be above the curve. For example, (0,0) is on the curve, whereas (0,1) is above the curve; so we would choose (0,0) as the nearest pixel to the curve. The next few points would be (1,1), (2,2), (3,3), (3,4), (3,5), ... The procedure should ensure that each point is connected (that is, is adjacent) to the previous point. The procedure should also be written as efficiently as possible. That is, use no multiplications except by constants in the inner loop. To draw the curve, write down an implicit function that defines the set of points on the curve, and then incrementally trace out the curve. Work out the math, and then fill in the following template:

```
drawcurve(int n)
{
  int x = 0;
  int y = 0;




  for(i=0; i<n; i++) {
    drawpoint(x,y);









  }
}
```

1. [Total: 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]. Image composition.

Image composition is a method for combining two or more images into a single image. The combination may be done with a variety of operators including blend, over, sum, logical or, etc.

1B [5 points]. Diffuse reflection.

In diffuse reflection, incident light is scattered equally in all directions. This holds for all incoming directions.

1C [5 points]. Key-frame animation.

To animate an articulated character, the character is put into a particular pose at a sequence of times. The pose of the character at inbetween times are created by interpolating the parameters that control the pose. Key-frame animation can also be done in other ways. For example, different shapes may be assigned to different times and interpolated in a variety of different ways.

1D [5 points]. Hierarchical modeling.

A hierarchical model is a model that is composed of other models. In computer graphics this is represented as a DAG. The nodes consist of a transformation and pointers to other models. Hierarchical models are used to model parts made of subparts. For example, a body is comprised of a torso, two legs and two arms.

2. [20 points] Transformations.

Computer graphics relies heavily on 3D transformations. Most common are the geometric transformations such as rotations, translations, scales, etc. In the following, T(dx,dy,dz) refers to a translation by (dx,dy,dz), Rx(a) refers to a rotation about the x-axis by a degrees, Ry(a) and Rz(a) refer to rotations about the y- and z-axis, respectively.

The order of transformations may matter. Also, sometimes the order may be rearranged, but the arguments will change. Describe whether the following statements are true or false.

T(1,0,0) Rx(45) = Rx(45) T(1,0,0)? **FALSE**

T(1,0,0) T(0,2,0) = T(0,2,0) T(1,0,0)? **TRUE**

Rx(45) Ry(30) = Ry(30) Rx(45)? **FALSE**

Rx(45) Rx(30) = Rx(30) Rx(45)? **TRUE**

T(1,0,0) Rz(180) = T(-1,0,0) Rz(0) **FALSE**

Transformations have inverses. State whether the following formulas for the inverse transformations are true or false.

$Rz(45)^{-1}$ = Rz(-45)? **TRUE**

$Rz(180)^{-1}$ = Rz(180)? **TRUE**

$T(1,1,1)^{-1}$ = T(1,1,1)? **FALSE**

$[Rz(45)\ T(1,0,0)]^{-1}$ = T(1,0,0) Rz(45)? **FALSE**

$[Rx(45)\ Ry(30)]^{-1}$ = Ry(-30) Rx(-45)? **TRUE**

3. [20 points] Ray tracing.

One of the most general methods for rendering is ray tracing. At the core of a ray tracer is a procedure to find ray-surface intersections. The inputs to the procedure are a ray and a formula for a surface, the output is the point of intersection.

Assume is ray is given by the following parametric equations:

$$x = x0 + t * x1$$
$$y = y0 + t * y1$$
$$z = z0 + t * z1;$$

(x0,y0,z0) is the origin of the ray, and (x1,y1,z1) is the direction of the ray. As t increases, the points on the ray move from the origin along the direction.

Work out a procedure for computing the point of intersection of a ray with a cone. A cone is given by the following implicit function $x^2+y^2-z^2 = 0$. In general, a ray intersects a cone in two points. Return the closest point in the direction of the ray.

First, substitute the ray equation into the implicit equation for the cone. This yields the following formula:

$$(x0+t*x1)^2+(y0+t*y1)^2-(z0+t*z1)^2$$

$$= (x0^2+y0^2-z0^2)+2*(x0*x1+y0*y1-z0*z1)*t+(x1^2+y1^2-z1^2)*t^2$$

$$= a * t^2 + b * t + c = 0$$

This equation may be solved using the quadratic formula.

$$t = [-b +/- sqrt(b^2-4*a*c)] / (2*a);$$

The discriminant $D = sqrt(b^2-4*a*c)$ determines the nature of the solution. If $D < 0$, there are no real roots and the ray does not intersect the cone. If $D == 0$, there is a single root and the ray is tangent to the cone. If $D > 0$, there are two intersections.

The smallest, positive real root represents the closest intersection.

To find the point of intersection, t is plugged into the equation of the ray,

5. [20 points] Rasterization.

In your new job as chief graphics software architect, you receive a request to write a procedure to draw the following curve:

$$y^2 = x^3$$

To simplify the problem, consider only positive y values. This procedure should draw **n** pixels by stepping along the curve starting at the origin. Each pixel should be under the curve, or on the curve. No pixels should be above the curve. For example, (0,0) is on the curve, whereas (0,1) is above the curve; so we would choose (0,0) as the nearest pixel to the curve. The next few points would be (1,1), (2,2), (3,3), (3,4), (3,5), … The procedure should ensure that each point is connected (that is, is adjacent) to the previous point. The procedure should also be written as efficiently as possible. That is, use no multiplications except by constants in the inner loop. To draw the curve, write down an implicit function that defines the set of points on the curve, and then incrementally trace out the curve. Work out the math, and then fill in the following template:

```
drawcurve(int n)
{
  int x = 0;
  int y = 0;
  int x3 = x2 = 0;   // keep track of x*x and x*x*x
  int y2 = 0;        // keep track of y*y

  for(i=0; i<n; i++) {
    drawpoint(x,y);

    y2 += 2*y + 1;
    y  += 1;         // always move up

    if( y2 < x3 ) { // if above curve, move right

      x3 += 3*x2 + 3*x + 1;
      x2 += 2*x + 1;
      x  += 1;

    }
  }
}
```

4. [20 points] Polygonal Meshes.

Polygonal meshes are widely used in computer graphics. A polygonal mesh consists of vertices that define the positions of 3D points, and topological information that organize the points into polygons. There are many different types of mesh data structures, depending on the operations that need to be performed.

1. Describe a simple data structure to store a triangular mesh so that vertices are not duplicated. In your description, define C structs for vertices and triangles.

A simple data structure is

```
struct Vertex { float x, y, z; };

struct Triangle { struct Vertex *v1, *v2, *v3; };
```

2. Refine the data structure described above so that neighboring triangles may be found in constant time.

This can be done by modifying the above data structure to include pointers to the triangles across each edge.

```
struct Triangle {
   struct Vertex   *v1, *v2, *v3;
   struct Triangle *t1, *t2, *t3;
};
```

It important to define what triangle is adjacent across what edge. For example, `t1` is the triangle across the edge defined by `v2` and `v3`.

Comprehensive Examination in Logic
Stanford University
Department of Computer Science
November 2002

INSTRUCTIONS

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

You should mark your answers only in the **answer sheet** that is provided with this part of the Comprehensive Examination. Be sure to write your **magic number** on the answer sheet.

This exam is **open book** and is composed of **44 questions** on **8 pages**. plus one answer sheet. For each question, write either **YES** or **NO** in the corresponding box of the answer sheet, or leave it blank. You will receive $+1$ point for each correct answer, -1 point for each incorrect answer, and **0** points for a blank (or crossed out) answer. You have **60 minutes** to complete the exam.

NOTATION

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 particular, recall that "theory" means "a set of sentences closed by logical consequence". The "theory of a model $\mathfrak{M}$" is the set of all sentences true in $\mathfrak{M}$.

Do not turn this page until instructed to do so.

An AI system for error diagnosis works by keeping a knowledge base, about the functioning of the artifact it diagnoses, as a set of sentences of first-order logic. When a set of observable behaviors are given as inputs, again as sentences of first-order logic, we can ask whether a certain component is necessarily broken. The system uses a first-order reasoning algorithm to look for a proof of this fact. Replacing a working component of the artifact is inadmissible. for cost reasons, but we can admit some undetected broken components. Then:

1. The reasoning algorithm should necessarily be sound.

2. The reasoning algorithm should necessarily be complete.

Answer "yes" or "no".

3. Is $((p \rightarrow q) \rightarrow q) \rightarrow q$ a tautology of propositional logic?

4. Suppose you are given a machine that you can feed any propositional formula and tells you, in constant time, whether that formula is a tautology. Could you then build a machine that decides validity of first-order sentences?

Assume $\alpha \rightarrow \beta \vee \gamma$ is valid. Then necessarily:

5. $\alpha \wedge \neg\beta \rightarrow \gamma$ is valid?

6. $\alpha \vee \beta \vee \gamma$ is satisfiable?

Which of the following are valid formulas of first-order logic?

7. $\forall x\ p(x) \vee \forall x\ q(x) \rightarrow \forall x\ (p(x) \vee q(x))$?

8. $\forall x\ (p(x) \vee q(x)) \rightarrow \forall x\ p(x) \vee \forall x\ q(x)$?

9. $\forall y\ (\ \exists x\ (p(y) \wedge q(x)) \leftrightarrow p(y) \wedge \exists x\ q(x)\ )$?

Consider the first-order language with equality and one binary predicate symbol R. A model of this language can be seen as a graph.

10. Can the property of a graph G *"There is a path of length at most 2 between any given two vertices of G."* be expressed with a first-order sentence over this language?

11. Does the following sentence express the property *"G is symmetric and transitive"*?

$$\forall x\ \forall y\ \forall z\ [[R(x,y) \wedge \neg R(y,x) \rightarrow \neg R(x,y)] \wedge [R(z,x) \wedge R(z,y) \rightarrow R(y,x)]]$$

12. Does the following sentence express the property *"G is acyclic"*?

$$\forall x\ \forall y\ \forall z\ [\neg R(x,x) \wedge [R(z,x) \wedge R(z,y) \rightarrow R(y,x)]]$$

Consider validity-preserving skolemization.

13. Does $\exists x\ (p(x,x) \rightarrow \forall y\ \exists z\ (q(x,y,z) \vee p(z,x)))$ correctly skolemize to $p(x,x) \rightarrow q(x,f(x),z) \vee p(z,x)$?

14. Does $\forall x\ \exists y\ (y > x \wedge \neg\exists u\ \exists v\ (\neg u = y \wedge \neg u = 1 \wedge u \cdot v = y))$ skolemize to $y > a \wedge \neg(\neg f(y) = y \wedge \neg f(y) = 1 \wedge u \cdot v = y)$? ($1, a, b, c$ are constants, u, v, x, y, z are variables, etc.: no tricks).

Consider the following terms t_1, t_2, and t_3:

$$\begin{array}{ll} t_1: & f(h(r,s,t), h(w,x,y), t, w) \\ t_2: & f(h(g(u,v), r, s), h(x,y,w), g(v,a), v) \\ t_3: & f(h(g(u,v), s, w), h(x,y,w), t, t) \end{array}$$

where $r, s, t, u, v, w, x, y, z$ are variables and a is a constant symbol. Which of the following sets are unifiable?

15. $\{t_1, t_2\}$

16. $\{t_1, t_3\}$

17. $\{t_2, t_3\}$

18. $\{t_1, t_2, t_3\}$

Consider the following deductive tableau, where x, y, and z are variables, and a is a constant:

	A	G
1		$p(a,x) \vee p(x,a)$
2	$p(z,y) \longrightarrow p(y,y)$	

Which of the following rows are results of an application of resolution according to the polarity strategy?

19.

		$\bot$

20.

		$p(a,a)$

21.

		$p(z,a)$

Let C be the first-order theory of complex numbers with addition, multiplication.

22. Does C have a finite model?

23. Does C have a countably infinite model?

24. Does C have an uncountably infinite model?

Consider the first-order theory, N, of the standard model of natural numbers with addition and multiplication.

25. Is N complete?

26. Is N decidable?

Let φ be an arbitrary sentence of first-order logic, and T an arbitrary axiomatizable theory over the same language as φ. Assume you have a resolution theorem prover that follows a resolution strategy S: you give the prover the axioms of T (including the Reflexivity of Equality axiom) and try to prove φ; the prover starts producing resolvents. Five weeks later it reports that no new resolvent can be produced and no proof has been found. Which of the following statements are correct?

27. A faster prover, or running the prover for a longer time, might find a proof of φ.

28. Either S is incomplete or T is incomplete.

29. If S is complete and T is complete, then T contains the sentence $\neg\varphi$.

Is there a first-order theory (with equality) that:

30. has exactly one model up to isomorphism?

31. has exactly one infinite model up to isomorphism?

32. has exactly two infinite models up to isomorphism?

Does the following hold?

33. For a set of first-order sentences, A, and a sentence φ, $A \models \varphi$ if and only if there is a finite subset B of A such that $B \models \varphi$.

34. A theory T is satisfiable if and only if it has a finite satisfiable subset.

35. Every theory is a subset of some consistent theory.

36. Let φ be a sentence that has no finite models. Then $\neg\varphi$ has a countably infinite model.

Does the following hold?

37. Let T be an incomplete theory, over a countable language, with infinite models and no finite models. Then there are at least two non-isomorphic models of T for every infinite cardinality.

38. There is a first-order sentence φ_{inf} such that, for every model $\mathfrak{M}$, $\mathfrak{M} \models \varphi_{\text{inf}}$ if and only if $|\mathfrak{M}|$ is infinite.

39. If $\sqsubset$ is well-founded over the set A, then every nonempty subset of $A \times A$ has a minimal element according to the relation $\prec$ defined by $(a, b) \prec (a', b') \Leftrightarrow a \sqsubset a'$ or $b \sqsubset b'$.

Answer "yes" or "no". In the context of deductive tableaux:

40. Is there a first-order theory for which resolution alone, without skolemization, is complete for computing validity?

41. Is there a first-order theory for which skolemization alone, without resolution, is complete for computing validity?

---

42. Let A be a set of first-order sentences and $\varphi[(\exists x\ \psi)^-] \in A$ (the minus sign denotes polarity). Let $A' = (A \setminus \{\varphi[(\exists x\ \psi)^-]\}) \cup \{\varphi[(\forall x\ \psi)^-]\}$. Is it true that if A is inconsistent, then A' is inconsistent?

Suppose you are visiting a forest in which every inhabitant is either a knight or a knave. Knights always speak the truth and knaves always lie.

43. You witness the following conversation among three inhabitants A, B, and C:

> A: At least one of the three of us is a knave.
> B: C is a knight.

Do you now know for sure who among A, B, and C are the knights?

44. Inspector Craig of Scotland Yard was called to the Forest of Knights and Knaves to help find a criminal named Arthur York. What made the process difficult was that it was not known whether Arthur York was a knight or a knave.

One suspect was arrested and brought to trial. Inspector Craig was the presiding Judge. Here is a transcript of the trial:

> CRAIG: What do you know about Arthur York?
> DEFENDANT: Arthur York once claimed that I was a knave.
> CRAIG: Are you by any chance Arthur York?
> DEFENDANT: Yes.

Is the defendant Arthur York?

ANSWER SHEET
Comprehensive Examination in LOGIC
November 2004

MAGIC NUMBER: _____

1		2		3		4		5	
6		7		8		9		10	
11		12		13		14		15	
16		17		18		19		20	
21		22		23		24		25	
26		27		28		29		30	
31		32		33		34		35	
36		37		38		39		40	
41		42		43		44			

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.

I acknowledge and accept the Honor Code. (Signed) _____

Comprehensive Examination in Logic

Stanford University

Department of Computer Science

November 2002

Solutions

INSTRUCTIONS

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

You should mark your answers only in the **answer sheet** that is provided with this part of the Comprehensive Examination. Be sure to write your **magic number** on the answer sheet.

This exam is **open book** and is composed of **44 questions** on **8 pages**, plus one answer sheet. For each question, write either **YES** or **NO** in the corresponding box of the answer sheet, or leave it blank. You will receive **+1** point for each correct answer, **−1** point for each incorrect answer, and **0** points for a blank (or crossed out) answer. You have **60 minutes** to complete the exam.

NOTATION

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 particular, recall that "theory" means "a set of sentences closed by logical consequence". The "theory of a model $\mathfrak{M}$" is the set of all sentences true in $\mathfrak{M}$.

Do not turn this page until instructed to do so.

An AI system for error diagnosis works by keeping a knowledge base, about the functioning of the artifact it diagnoses, as a set of sentences of first-order logic. When a set of observable behaviors are given as inputs, again as sentences of first-order logic, we can ask whether a certain component is necessarily broken. The system uses a first-order reasoning algorithm to look for a proof of this fact. Replacing a working component of the artifact is inadmissible, for cost reasons, but we can admit some undetected broken components. Then:

1. The reasoning algorithm should necessarily be sound.

Answer. YES. Soundness means every sentence that is proved follows from the assumptions, which is what we need to prevent detecting unexisting errors.

2. The reasoning algorithm should necessarily be complete.

Answer. NO. Completeness means every sentence that follows from the assumptions is provable, but we can admit some undetected broken components, thus completeness is not required.

Answer "yes" or "no".

3. Is $((p \rightarrow q) \rightarrow q) \rightarrow q$ a tautology of propositional logic?

Answer. NO. The falsifying valuation, obtainable by the falsification method, makes p true and q false.

4. Suppose you are given a machine that you can feed any propositional formula and tells you, in constant time, whether that formula is a tautology. Could you then build a machine that decides validity of first-order sentences?

Answer. NO. This would imply decidability of validity for first-order logic.

Assume $\alpha \rightarrow \beta \vee \gamma$ is valid. Then necessarily:

5. $\alpha \wedge \neg\beta \rightarrow \gamma$ is valid?

Answer. YES. Consider an arbitrary interpretation. If $\alpha \wedge \neg\beta$ holds in that interpretation, then since $\alpha \rightarrow \beta \vee \gamma$ holds by assumption, $\beta \vee \gamma$ holds, and therefore, since β does not hold, γ does. Thus $\alpha \wedge \neg\beta \rightarrow \gamma$ holds in every interpretation.

6. $\alpha \vee \beta \vee \gamma$ is satisfiable?

Answer. NO. Consider for example $\alpha \equiv \beta \equiv \gamma \equiv p \wedge \neg p$.

Which of the following are valid formulas of first-order logic?

7. $\forall x\ p(x) \vee \forall x\ q(x) \rightarrow \forall x\ (p(x) \vee q(x))$?

Answer. YES.

8. $\forall x\ (p(x) \vee q(x)) \rightarrow \forall x\ p(x) \vee \forall x\ q(x)$?

Answer. NO.

9. $\forall y\ (\ \exists x\ (p(y) \wedge q(x)) \leftrightarrow p(y) \wedge \exists x\ q(x)\ )$?

Answer. YES.

Consider the first-order language with equality and one binary predicate symbol R. A model of this language can be seen as a graph.

10. Can the property of a graph G *"There is a path of length at most 2 between any given two vertices of G."* be expressed with a first-order sentence over this language?

Answer. YES. For example,
$\forall x\ \forall y\ (x = y \vee R(x,y) \vee \exists z\ (R(x,z) \wedge R(z,y)))$.

11. Does the following sentence express the property *"G is symmetric and transitive"*?

$$\forall x\ \forall y\ \forall z\ [[R(x,y) \wedge \neg R(y,x) \rightarrow \neg R(x,y)] \wedge [R(z,x) \wedge R(z,y) \rightarrow R(y,x)]]$$

Answer. YES. The first conjunct is equivalent to $R(x,y) \rightarrow R(y,x)$, i.e., symmetry, in view of the propositional tautology $(p \rightarrow q) \leftrightarrow (p \wedge \neg q \rightarrow p)$. The second conjunct is equivalent to transitivity once symmetry is assumed, because $R(z,y)$ is then equivalent to $R(y,z)$.

12. Does the following sentence express the property *"G is acyclic"*?

$$\forall x\ \forall y\ \forall z\ [\neg R(x,x) \wedge [R(z,x) \wedge R(z,y) \rightarrow R(y,x)]]$$

Answer. NO. In fact, acyclicity cannot be expressed in first-order logic.

Consider validity-preserving skolemization.

13. Does $\exists x\ (p(x,x) \rightarrow \forall y\ \exists z\ (q(x,y,z) \vee p(z,x)))$ correctly skolemize to $p(x,x) \rightarrow q(x,f(x),z) \vee p(z,x)$?

Answer. YES.

14. Does $\forall x\ \exists y\ (y > x \wedge \neg\exists u\ \exists v\ (\neg u = y \wedge \neg u = 1 \wedge u \cdot v = y))$ skolemize to $y > a \wedge \neg(\neg f(y) = y \wedge \neg f(y) = 1 \wedge u \cdot v = y)$? ($1, a, b, c$ are constants, u, v, x, y, z are variables, etc.: no tricks).

Answer. NO. The last u and v should be $f(y)$ and $g(y)$, respectively.

Consider the following terms t_1, t_2, and t_3:

$$\begin{array}{ll} t_1: & f(h(r,s,t),h(w,x,y),t,w) \\ t_2: & f(h(g(u,v),r,s),h(x,y,w),g(v,a),v) \\ t_3: & f(h(g(u,v),s,w),h(x,y,w),t,t) \end{array}$$

where $r, s, t, u, v, w, x, y, z$ are variables and a is a constant symbol. Which of the following sets are unifiable?

15. $\{t_1, t_2\}$

Answer. YES.

16. $\{t_1, t_3\}$

Answer. YES.

17. $\{t_2, t_3\}$

Answer. NO.

18. $\{t_1, t_2, t_3\}$

Answer. NO.

Consider the following deductive tableau, where x, y, and z are variables, and a is a constant:

	A	G
1		$p(a,x) \vee p(x,a)$
2	$p(z,y) \rightarrow p(y,y)$	

Which of the following rows are results of an application of resolution according to the polarity strategy?

19.

		$\perp$

Answer. NO. This resolvent can be obtained only by violating the polarity strategy.

20.

		$p(a,a)$

Answer. NO. Obtaining this resolvent would involve using unifier that is not most general.

21.

		$p(z,a)$

Answer. YES. Resolve the two (positive) p-atoms in Goal 1 and the (negative) p-atom in the consequent of Assertion 2, with most general unifier $\{x \leftarrow a, y \leftarrow a\}$. In fact, this is the only resolvent according to the polarity strategy.

Let C be the first-order theory of complex numbers with addition, multiplication.

22. Does C have a finite model?

Answer. NO. The theory contains a formula saying there are n different elements in the domain, for every n, thus every domain must be infinite.

23. Does C have a countably infinite model?

Answer. YES. By the Löwenheim-Skolem-Tarski Theorem, since the language is finite and there is an uncountably infinite model (next question).

24. Does C have an uncountably infinite model?

Answer. YES. The complex numbers are one.

Consider the first-order theory, N, of the standard model of natural numbers with addition and multiplication.

25. Is N complete?

Answer. YES. The theory of any single structure is complete.

26. Is N decidable?

Answer. NO. If it were, it would be an axiomatizable extension of Peano Arithmetic, thus violating Gödel's Incompleteness Theorem.

Let φ be an arbitrary sentence of first-order logic, and T an arbitrary axiomatizable theory over the same language as φ. Assume you have a resolution theorem prover that follows a resolution strategy S; you give the prover the axioms of T (including the Reflexivity of Equality axiom) and try to prove φ; the prover starts producing resolvents. Five weeks later it reports that no new resolvent can be produced and no proof has been found. Which of the following statements are correct?

27. A faster prover, or running the prover for a longer time, might find a proof of φ.

Answer. NO.

28. Either S is incomplete or T is incomplete.

Answer. NO. If S is any complete strategy and T is any complete consitent theory, a φ such that $T \models \neg\varphi$ will do.

29. If S is complete and T is complete, then T contains the sentence $\neg\varphi$.

Answer. YES. Since S is complete, φ is not in the theory T (otherwise, the prover would eventually produce a proof, but none has been found and no new resolvent can be produced). Therefore, since T is complete, $\neg\varphi$ must be in the theory T.

Is there a first-order theory (with equality) that:

30. has exactly one model up to isomorphism?

Answer. YES. For example, the theory with only the axiom $\forall x\ \forall y\ x = y$.

31. has exactly one infinite model up to isomorphism?

Answer. NO. It follows from the Löwenheim-Skolem-Tarski Theorem that if a first-order theory has infinite models then it has infinite models of any sufficiently large cardinality. Two models of different cardinality cannot be isomorphic.

32. has exactly two infinite models up to isomorphism?

Answer. NO. The reason is exactly the same as in the answer to the previous question.

Does the following hold?

33. For a set of first-order sentences, A, and a sentence φ, $A \models \varphi$ if and only if there is a finite subset B of A such that $B \models \varphi$.

Answer. YES. This is the Compactness Theorem, or, if you remember the Compactness Theorem in a different form, it follows from completeness.

34. A theory T is satisfiable if and only if it has a finite satisfiable subset.

Answer. NO. Then every theory would be satisfiable, since the empty set is satisfiable and is a subset of every theory. The right formulation is "...all of its finite subsets are satisfiable."

35. Every theory is a subset of some consistent theory.

Answer. NO. An inconsistent theory, no matter what you add to it, stays inconsistent.

36. Let φ be a sentence that has no finite models. Then $\neg\varphi$ has a countably infinite model.

Answer. YES. If φ is false in every finite model, then $\neg\varphi$ is true in every finite model, thus, by compactness, $\neg\varphi$ must be true in an infinite model.

Does the following hold?

37. Let T be an incomplete theory, over a countable language, with infinite models and no finite models. Then there are at least two non-isomorphic models of T for every infinite cardinality.

Answer. YES. Take a φ such that neither φ nor $\neg\varphi$ is a logical consequence of T. Let $\mathfrak{M}$ and $\mathfrak{N}$ be (countably infinite) models of $T \cup \{\varphi\}$ and $T \cup \{\neg\varphi\}$ respectively. Then $\mathfrak{M}$ and $\mathfrak{N}$ are models of T, and they are not isomorphic.

38. There is a first-order sentence φ_{inf} such that, for every model $\mathfrak{M}$, $\mathfrak{M} \models \varphi_{\text{inf}}$ if and only if $|\mathfrak{M}|$ is infinite.

Answer. NO. If there were such a sentence, then $\neg\varphi_{\text{inf}}$ would hold exactly in all finite model; by compactness, it would have an infinite model, violating its own definition.

39. If $\sqsubset$ is well-founded over the set A, then every nonempty subset of $A \times A$ has a minimal element according to the relation $\prec$ defined by $(a, b) \prec (a', b') \Leftrightarrow a \sqsubset a'$ or $b \sqsubset b'$.

Answer. NO. For example, consider $A = \{0, 1\}$. $\sqsubset = \{(0, 1)\}$, and let the subset of $A \times A$ be $\{(0, 1), (1, 0)\}$.

Answer "yes" or "no". In the context of deductive tableaux:

40. Is there a first-order theory for which resolution alone, without skolemization, is complete for computing validity?

Answer. YES. For example, the inconsistent theory with axiom $\bot$.

41. Is there a first-order theory for which skolemization alone, without resolution, is complete for computing validity?

Answer. YES. For example, the inconsistent theory with axiom $\bot$.

---

42. Let A be a set of first-order sentences and $\varphi[(\exists x\ \psi)^-] \in A$ (the minus sign denotes polarity). Let $A' = A \setminus \{\varphi[(\exists x\ \psi)^-]\} \cup \{\varphi[(\forall x\ \psi)^-]\}$. Is it true that if A is inconsistent, then A' is inconsistent?

Answer. NO. For a counterexample, take $A = \{\varphi\} = \{(\exists x\ x = a) \rightarrow \bot\}$. The reverse is true, that is, if A' is inconsistent then A is inconsistent. The reason is left as an exercise.

Suppose you are visiting a forest in which every inhabitant is either a knight or a knave. Knights always speak the truth and knaves always lie.

43. You witness the following conversation among three inhabitants A, B, and C:

> A: At least one of the three of us is a knave.
> B: C is a knight.

Do you now know for sure who among A, B, and C are the knights?

Answer. YES. A is the only knight: A cannot be a knave, because then he would have spoken the truth by saying at least one of the three was a knave. Thus A is a knight and she spoke the truth. Thus either B or C is a knave. If B is a knight, then C is a knight, too, thus B is a knave. And hence, since B is lying, C is a knave.

Alternatively, and more formally, if a stays for "A is a knight", $\neg a$ for "A is a knave", and so on, the conversation tells us that

$$a \leftrightarrow \neg a \vee \neg b \vee \neg c$$
$$b \leftrightarrow c$$

From this it is now provable, in propositional logic, that $a \wedge \neg b \wedge \neg c$.

44. Inspector Craig of Scotland Yard was called to the Forest of Knights and Knaves to help find a criminal named Arthur York. What made the process difficult was that it was not known whether Arthur York was a knight or a knave.

One suspect was arrested and brought to trial. Inspector Craig was the presiding Judge. Here is a transcript of the trial:

> CRAIG: What do you know about Arthur York?

DEFENDANT: Arthur York once claimed that I was a knave.
CRAIG: Are you by any chance Arthur York?
DEFENDANT: Yes.

Is the defendant Arthur York?

Answer. NO. If the defendant is Arthur York, we get the following contradiction. Suppose he is Arthur York. Then he is a knight, since he claimed to be Arthur York. That would mean that his first answer to Craig was also true, which means that he, Arthur York, once claimed that he was a knave. But that is impossible! Therefore the defendant is not Arthur York, although he is, of course, a knave.

Alternatively, if a means "the defendant is Arthur York", k means "the defendant is a knight", and b means "Arthur York is a knight", we can formalize the information as

$$k \leftrightarrow (b \leftrightarrow \neg k) \text{ (first answer)}$$
$$k \leftrightarrow a \text{ (second answer)}$$
$$a \rightarrow (k \leftrightarrow b) \text{ (if he is AY then } k \text{ and } b \text{ equal)}$$

and from these prove $\neg a$.

Stanford University
Computer Science Department

Fall 2002 Comprehensive Exam in Networks

1. **Closed Book:** no notes, textbooks, laptops, Internet access, etc.
2. **Write only in the Blue Books: No credit** for answers written on these exam pages.
3. **Write Magic number** on the cover of **EACH blue book**.
4. **The exam is timed for one hour.**

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.

Comprehensive Exam: Networks (60 points) **Autumn 2002**

1. (15 points total) *TCP transport protocol* TCP has a 32-bit sequence number field and a 16-bit window

   (a) (5 points) Give a quantitative argument, as the designers of TCP might have done in 1980, that 32-bits is a good choice of size for the sequence number field.

   (b) (5 Points) In the same vein, why is 16 bits a reasonable choice for the window parameter.

   (c) (5 points) Quantify the limitations and concerns that these sizes raise based on how the technology has changed since 1980, i.e. lower cost memory, faster processors, optical links, etc.

2. (15 points total) *Virtual Circuits vs. Datagrams* Alexander Graham Bell returns from the dead to set us straight on (virtual) circuit switching, given he considers we went off the rails in using datagrams in the Internet. Give your cogent response to each of the points that this old cadaver makes:

   (a) (5 points) "Hot dang, this datagram stuff requires huge packet headers that incur excessive bandwidth overhead. Circuit switching allows you to use itsty bitsy tiny little circuit IDs."

   (b) (5 Points) "This new fangled datagram nonsense cost much more to do packet classification and forwarding decisions in each switch because of the large clumsy headers. With circuit switching, the circuit id can be an index into a table that directly indicates the next hop."

   (c) (5 points) "This datagram nonsense does not allow you to reserve bandwidth so you cannot know whether your packets are going to get through. Good grief, the only guarantees are death and taxes, and I'm already dead."

3. (15 points total) *Ethernet*

   (a) (5 points) Describe the Ethernet media access control (MAC) protocol, CSMA-CD, comparing it to the Aloha network and p-persistent protocols.

   (b) (5 points) Original Ethernet was at 10 Mbps with limitations on cable length and packet size. Now, IEEE is busy standardizing 10 Gbps Ethernet. Describe how these parameters need to change, if at all, if you insist on using the MAC protocol above at this higher speed.

   (c) (5 points) Peterson and Davie say: "it might seem that a wireless protocol would follow the exactly the same algorithm as the Ethernet" as a lead-in to why not. Describe why not and what 802.11 does about it.

4. (15 points total) *End to end.*

   (a) (6 points) Describe the "end-to-end" argument in networking,
   illustrating how a file transfer program should behave if it was truly "end-to-end".

(b) (6 Points) Osama Bin Laden, pointing out yet another hypocrisy of the West, argues that if we really believed in end-to-end, we would not mess around with Ethernet CRCs, IP checksums, TCP checksums and TCP retransmissions. George W. Bush claims it's an important part of our way of life. Describe why Bin Laden's wrong in this particular instance.

(c) (3 Points) Give an example of how Internet protocols are not completely consistent with the end-to-end argument.

The End

20

Answers to Comprehensive Exam: Networks (60 points) Autumn 2002

1. (15 points total) *TCP transport protocol* TCP has a 32-bit sequence number field and a 16-bit window

   (a) (5 points) Give a quantitative argument, as the designers of TCP might have done in 1980, that 32-bits is a good choice of size for the sequence number field.

   Answer: The key issue is being able to detect old delayed segments showing up. The 32 bits allows 4 gigabytes to be transmitted before wrap, which is 8*4G/10M = 3200 seconds or almost one hour at 10 Mbps, which was high-speed back in those days. And, a TCP segment is not expected to live in the next more than a few seconds so this should catch old delayed segment even just using half the range.

   (b) (5 Points) In the same vein, why is 16 bits a reasonable choice for the window parameter.

   Answer: This allows up to 64 kilobytes to be outstanding, while on a LAN with 10 Mbps data rate, this allows a RTT of up to about 50 milliseconds, whereas the actual RTT is microseconds. WAN data rates tend to be much lower even though the RTT is longer.

   (c) (5 points) Quantify the limitations and concerns that these sizes raise based on how the technology has changed since 1980, i.e. lower cost memory, faster processors, optical links, etc.

   Answer: At 10 Gbps, you can wrap in 3.2 seconds, and usually one provides half the range for new packets and half for old, so you are in trouble if segments live from more than 1.6 seconds. With WANs, this is getting rather close to expected max. packet lifetimes, which can be hundreds of milliseconds. Similarly, for the window parameter, multi-gigabit links are arising that have 100 Ms RTT, so we cannot operate full-speed because we dont have the delay-bandwidth product.

2. (15 points total) *Virtual Circuits vs. Datagrams* Alexander Graham Bell returns from the dead to set us straight on (virtual) circuit switching, given he considers we went off the rails in using datagrams in the Internet. Give your cogent response to each of the points that this old cadaver makes:

   (a) (5 points) "Hot dang, this datagram stuff requires huge packet headers that incur excessive bandwidth overhead. Circuit switching allows you to use itsty bitsy tiny little circuit IDs."

   Answer: Large packets, which dominate the traffic, are 1000-1500 bytes and the headers are only 56 bytes or so, so percentage overhead is not significant. Even with average packet size of 300 bytes means the overhead is 20 percent. Moreover, lots of bandwidth is wasted with virtual circuit setup, and connections/flows in the Internet tend to be very short on average.

   (b) (5 Points) "This new fangled datagram nonsense cost much more to do packet classification and forwarding decisions in each switch because of the large clumsy

headers. With circuit switching, the circuit id can be an index into a table that directly indicates the next hop."

Answer: Typically, only a small port of the header needs to be looked up for a forwarding decision, such as the 32-bite IPv4 address in a router, so the size is not much larger than a VCid. For higher-level classification, it would be more expensive to have a network-layer connection per higher-level connection (TCP) because they are so short-lived, and connections are short-lived — see above.

(c) (5 points) "This datagram nonsense does not allow you to reserve bandwidth so you cannot know whether your packets are going to get through. Good grief, the only guarantees are death and taxes, and I'm already dead."

Answer: There are no guarantees in either case, because a router can fail, At least with datagrams, your packets might get through by routing around the failure, rather than having to re-setup the whole virtual circuit. Moreover, you can do RVSP on top of datagrams if you insist on reservations. (Thanks for the telephone, in any case.)

3. (15 points total) *Ethernet*

(a) (5 points) Describe the Ethernet media access control (MAC) protocol, CSMA-CD, comparing it to the Aloha network and p-persistent protocols.

Answer: Listen before transmit, transmit if idle or wait for idle. Listen while transmitting and abort if collision and retry as above. Aloha does not listen for collisions. A p-persistent protocol only transmits with probability p if the line is idle, ie. Ethernet is 1-persistent.

(b) (5 points) Original Ethernet was at 10 Mbps with limitations on cable length and packet size. Now, IEEE is busy standardizing 10 Gbps Ethernet. Describe how these parameters need to change, if at all, and why if you insist on using the MAC protocol above at this higher speed.

Answer: You need to either decrease the max. cable length by a factor of 1000 or else increase the min packet size by a factor of a 1000 (or some combination of the two i.e. 100/10) so that transmitter is still listening/transmitting when there can be a collision. This is not practical to do. 64Kbytes min packet size!

(c) (5 points) Peterson and Davie say: "it might seem that a wireless protocol would follow the exactly the same algorithm as the Ethernet" as a lead-in to why not. Describe why not and what 802.11 does about it.

Answer: In the wireless environment, nodes cannot necessary hear the transmission of every other node that might interfere with its transmission, unlike Ethernet. The 802.11 solution is to request-to-send to the destination before sending, and getting clear to send back from destination, with the hope that others close by will see this and avoid colliding.

4. (15 points total) *End to end.*

(a) (6 points) Describe the "end-to-end" argument in networking, illustrating how a file transfer program should behave if it was truly "end-to-end".

Answer: End-to-end semantics can only be ensured by end-to-end mechanisms and checks because intermediate/lower-level mechanisms can fail in ways that only end-to-end mechanisms can detect. Lower-level mechanisms are at best, optimizations. An end-to-end file transfer ensures that a file was accurately transferred to a remote disk by transmitting a file-level checksum on the file to the receiver and having the receiver read back the file, recompute the checksum and verify against the supplied one. This catches any intermediate failures (except low prob. checksum fooling).

(b) (6 Points) Osama Bin Laden, attempting to point out yet another hypocrisy of the West, argues that if we really believed in end-to-end, we would not mess around with Ethernet CRCs, IP checksums, TCP checksums and TCP retransmissions. Describe why Bin Laden's wrong in this particular instance.

Answer: The TCP checksum and retransmission mechanism is an optimization over having to retransmit larger units, such as the whole file in a file transfer case, when a packet is dropped or corrupted. The Ethernet CRC helps further to detect corrupted packets, given that the TCP checksum is not very strong, and Ethernet CRC is normally computed in hardware. IP checksums might be argued helpful to avoid forwarding packets whose headers have been trashed, another optimization.

(c) (3 Points) Give an example of how Internet protocols are not completely consistent with the end-to-end argument.

Answer: Standard FTP does not include an end-to-end file-level checksum. Also, telnet has no application-level check other than the user noticing missing characters.

The End

Computer Science Department
Stanford University
Comprehensive Examination in Numerical Analysis
Fall 2002

1. (10 points) Consider a system of linear equations

$$Ax = b \tag{1}$$

where A is a real $n \times n$ non-singular matrix, $b \in R^n$.

a) (3 points) Describe how to use the LU decomposition (assume it is given) to solve the system of linear equations (the Gaussian elimination). Why might one need pivoting?

b) (4 points) Describe how to use the QR decomposition (assume it is given) for solving (1). In what cases might one use the QR decomposition rather than the LU decomposition?

c) (3 points) Describe an iterative method to solve (1) in case the matrix A is a symmetric positive definite and discuss its convergence properties. In what cases would iterative methods prevail over the direct ones?

2. (10 points) Consider a system of ordinary differential equations

$$\begin{pmatrix} x \\ y \end{pmatrix}_t = A \begin{pmatrix} x \\ y \end{pmatrix}, \quad \begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix}, \quad A \in R^{2x2}. \tag{2}$$

a) (3 points) Assume A has a complete set of eigenvectors. How do eigenvalues influence stability (definition of stability: $\left|\begin{matrix} x(t) \\ y(t) \end{matrix}\right| \le K \left|\begin{matrix} \varphi_1 \\ \varphi_2 \end{matrix}\right|$ for $t \ge 0$, where K is independent of time and initial data) of this problem?

b) (2 points) Assume A has only one eigenvalue with algebraic multiplicity 2. Do the stability conditions remain the same?

c) (5 points) Let $A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. Describe how to solve this system with the forward Euler scheme. Is it a good numerical method to solve this problem? If not, which method would you use instead?
Hint: Solve (2) and consider the analytical stability of its solution.

3. (10 points) Numerical Integration.

a) (3 points) Write the formula for the composite trapezoidal rule scheme.

b) (2 points) Define the degree of accuracy for a numerical quadrature scheme.

c) (5 points) Derive Simpson's method using a Taylor polynomial and in the process show that it has degree of accuracy 3.

Solutions

1 a) Denote $r = Ax - b$; $x = \arg\min \|r\|_2 = \arg\min r^T r = \arg\min(x^T A^T Ax - 2(A^T b)^T x + b^T b)$

Differentiating with respect to x and setting the result to zero, we get $2A^T Ax - 2A^T b = 0$, or

$$A^T Ax = A^T b \qquad (2)$$

A is of full column rank => $A^T A$ is nonsingular => (2) has a unique solution.

$A^T A$ is symmetric and positive definite => we can compute its Cholesky factorization

$$A^T A = LL^T,$$

where L - lower triangular. This allows us to reduce (2) to solving the triangular systems

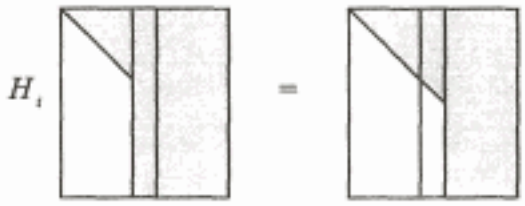
$$Ly = A^T b, \qquad L^T x = y.$$

b)First we want to decompose A into the product $Q\begin{bmatrix} R \\ 0 \end{bmatrix}$, $Q : m \times m$ orthogonal, $R : n \times n$ upper-triangular; since A is of full column rank, R is nonsingular.

To reduce A to the upper-triangular form, we successively apply Householder transformations

$$H_n \ldots H_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}, \qquad \text{where } H_i = I - 2\frac{v_i v_i^T}{v_i^T v_i} \quad \Rightarrow \quad \text{orthogonal and symmetric.}$$

H_i reflects a vector against the hyperplane $v_i^{\perp}$. We choose v_i so that H_i zeros out the subdiagonal part of the ith column $\tilde{a}_i$ of the current state of A - $(H_{i-1} \ldots H_1 A)$



If $\tilde{a}_i' = \{\tilde{a}_i$ with the upper i-1 entries set to 0$\}$, then $v_i = \tilde{a}_i' \pm \|\tilde{a}_i'\|_2 e_i$, where the sign is chosen so as to avoid cancellation.

Now, using this decomposition $Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$, $Q^T = H_n \ldots H_1$,

$$x = \arg\min \|Ax - b\|_2 = \arg\min \|Q^T Ax - Q^T b\|_2 = \arg\min \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} x - Q^T b \right\|_2;$$

let $Q^T b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, $b_1 : n \times 1$, $b_2 : (m-n) \times 1$;

$$x = \arg\min \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right\|_2^2 = \arg\min(\|Rx - b_1\|_2^2 + \|b_2\|_2^2) = \arg\min \|Rx - b_1\|_2 = R^{-1} b_1$$

c)

	Computations	Accuracy
Normal equations	$A^T A$ - $\approx mn^2$ flops; Cholesky factorization - $\approx \frac{n^3}{3}$ flops Triangular systems - $O(n^2)$ $\approx mn^2 + \frac{n^3}{3}$ flops	relative error in x is proportional to $(cond(A))^2$
Householder transformations	$\approx 2mn^2 + \frac{2}{3}n^3$ flops	relative error in x is proportional to $cond(A) + \|r\|_2 (cond(A))^2$

For nearly square problems, $m \approx n$, the two methods require about the same amounts of work, but for $m >> n$ normal equations method is about 2 times cheaper than Householder method. On the other hand, the Householder method is more accurate.

2 a) Via elementary symbolic calculations we obtain $y(t) = e^{\lambda t}$. As $t \to \infty$, $y(t) \to +0$.

---

b) $y_{k+1}(1 - \frac{\lambda h}{2}) = y_k(1 + \frac{\lambda h}{2})$, $\quad y_{k+1} = \frac{1 + \lambda h / 2}{1 - \lambda h / 2} y_k$, $\quad y_k = \left(\frac{1+\lambda h/2}{1-\lambda h/2}\right)^k y_0$

$y_k \to 0 \quad \Leftrightarrow \quad \left|\frac{1+\lambda h/2}{1-\lambda h/2}\right| < 1, \quad (2+\lambda h)^2 < (2-\lambda h)^2, \quad 4\lambda h < -4\lambda h, \quad \lambda h < 0,$

and since we assume $h > 0$, this is true for all negative λ.

---

c) We have $\quad y_{k+1} = y_k + \frac{f(t_k, y_k) + f(t_{k+1}, y_{k+1})}{2} h \qquad (1)$

For the true solution $y(t)$ we can write

$$y(t_{k+1}) = y(t_k) + \frac{y'(t_k) + y'(t_{k+1})}{2} h + \xi, \qquad (2)$$

where ξ is some unknown term. From Taylor expansion we have two formulas:

$\frac{f(x+h) + f(x-h)}{2} = f(x) + \frac{\theta h^2}{2}, \qquad |\theta| \le \|f''\|_C;$

$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{\theta_1 h^2}{6}, \qquad |\theta_1| \le \|f^{(3)}\|_C.$

Regrouping (2) and applying these formulas, we get:

$\frac{\xi}{h} = \frac{y(t_{k+1}) - y(t_k)}{h} - \frac{y'(t_{k+1}) + y'(t_k)}{2} = y'(t_{k+1/2}) + \frac{\eta h^2}{6} - y'(t_{k+1/2}) - \frac{\eta_1 h^2}{2} = (\frac{\eta}{6} - \frac{\eta_1}{2})h^2$, where

$|\eta| \le \|y^{(3)}\|_C, \quad |\eta_1| \le \|y''\|_C$. Hence $\quad |\xi| \le \frac{1}{2}(\|y''\|_C + \|y^{(3)}\|_C)h^3. \qquad (3)$

Now denoting $\delta_k = y_k - y(t_k)$ and subtracting (1)-(2), we get

$$\delta_{k+1} = \delta_k + \frac{f(t_k, y_k) - f(t_k, y(t_k))}{2} h + \frac{f(t_{k+1}, y_{k+1}) - f(t_{k+1}, y(t_{k+1}))}{2} h - \xi.$$

Then using Lipschitz-continuity of f,

$|\delta_{k+1}| \le |\delta_k| + \frac{Lh}{2}|\delta_k| + \frac{Lh}{2}|\delta_{k+1}| + |\xi|, \qquad (1 - \frac{Lh}{2})|\delta_{k+1}| \le (1 + \frac{Lh}{2})|\delta_k| + |\xi|,$

$$|\delta_{k+1}| \le \left|\frac{1+Lh/2}{1-Lh/2}\right||\delta_k| + \frac{|\xi|}{1-Lh/2} \le \left|\frac{1+Lh/2}{1-Lh/2}\right||\delta_k| + 2|\xi| \qquad \text{for} \quad h \le \frac{1}{L}.$$

To simplify this, consider the general situation: $x_{k+1} \le ax_k + b$. Applying this inequality recursively, we get $x_k \le a^k x_0 + b(a^{k-1} + a^{k-2} + \ldots + 1) = a^k x_0 + b\dfrac{a^k - 1}{a-1}$. The first term in our case disappears, because $y(0) = y_0 = 1$; therefore $|\delta_k| \le 2|\xi| \dfrac{\left|\frac{1+Lh/2}{1-Lh/2}\right|^k - 1}{\left|\frac{1+Lh/2}{1-Lh/2}\right| - 1}$.

Now since $k = t_k / h$, $\left|\dfrac{1+Lh/2}{1-Lh/2}\right|^k \xrightarrow[h\to 0]{} \dfrac{e^{\frac{Lt_k}{2}}}{e^{-\frac{Lt_k}{2}}} = e^{Lt_k}$, and hence

$$|\delta_k| \le 2|\xi| \frac{e^{Lt_k} + \varepsilon(h) - 1}{Lh}(1 - Lh/2) \le \frac{2|\xi|}{h}\frac{e^{Lt_k} + \varepsilon(h) - 1}{L},$$

where $\dfrac{2|\xi|}{h} \le (\|y''\|_C + \|y^{(3)}\|_C)h^2$, and $\varepsilon(h) \to 0$ as $h \to 0$. Thus $|\delta_k| \le Ch^2$ for sufficiently small h.

3 a) Let $l(x) = f(a) + (x-a)\dfrac{f(b)-f(a)}{b-a}$, then

$$I(f) = \int_a^b l(x)dx = f(a)(b-a) + \frac{(x-a)^2}{2}\bigg|_a^b \frac{f(b)-f(a)}{b-a} = (b-a)(f(a) + \frac{f(b)-f(a)}{2}) = (b-a)\frac{f(a)+f(b)}{2}$$

---

b) Let $I_0 = \int_a^b f(x)dx$, $\quad \Delta = b - a$.

$$f(a+t) = f(a) + f'(a)t + f''(\xi)\frac{t^2}{2}, \quad I_0 = f(a)\Delta + f'(a)\frac{\Delta^2}{2} + A\frac{\Delta^3}{6}, \qquad |A| \le \|f''\|_C$$

$$f(b-t) = f(b) - f'(b)t + f''(\xi_1)\frac{t^2}{2}, \quad I_0 = f(b)\Delta - f'(b)\frac{\Delta^2}{2} + B\frac{\Delta^3}{6}, \qquad |B| \le \|f''\|_C.$$

Averaging the two expressions for I_0, we obtain

$$I_0 = \frac{f(a)+f(b)}{2}\Delta + \frac{A+B}{2}\frac{\Delta^3}{6} = I + C\frac{\Delta^3}{6}, \quad \text{where} \quad C = \frac{A+B}{2}. \quad \text{This implies} \quad |I_0 - I| \le \|f''\|_C \frac{\Delta^3}{6}.$$

---

c) Let $I_0^{(i)}$, $I^{(i)}$ be the true integral and our estimate correspondingly for the ith subinterval. Then

$$I_0 = \sum_{i=1}^n I_0^{(i)}, \qquad I = \sum_{i=1}^n I^{(i)}.$$

$$|I_0 - I| \le \sum_{i=1}^n |I_0^{(i)} - I^{(i)}| \underset{(b)}{\le} \sum_{i=1}^n |C_i|\frac{\Delta_i^3}{6} \le \sum_{i=1}^n \|f''\|_C \frac{1}{6}\left(\frac{\Delta}{n}\right)^3 = \|f''\|_C \frac{\Delta^3}{6n^2} = \frac{\|f''\|_C (b-a)^3}{6n^2}.$$

Comprehensive Exam: Programming Languages Autumn 2002

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. 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	# 3	Total
Score					
Max	7	5	6	12	30

1. (7 points) *Type Inference*

   (a) (3 points) This part of the question asks you to write the ML type of an ML function. Recall that an ML function that takes two integer arguments and returns a Boolean value has type (`int` $*$ `int`) $\rightarrow$ `bool`. What is the ML type for the following function?

   ```
   fun applytwice(f, x) = f(f(x));
   ```

   (b) (2 points) Describe types `<type1>`, `<type2>` and `<type3>` that will make the analogous C code below statically typed. You may write the types using correct or mostly correct C syntax. In particular you can write `<type2> f` in a form where `f` appears in the middle of the type, if you prefer (and if this is correct).

   ```
   <type1> applytwice(<type2> f, <type3> x) {
             return f(f(x));
   };
   ```

   (c) (2 points) Why is the ML type more useful than the C type for `applytwice`? Specifically, what flexibility does the ML type system give, for functions with the kind of type you wrote in (a), that the C type system does not give you for the function in (b)?

2. (5 points) *ALGOL 60 Pass-By-Name*

   In pass-by-name, the result of a procedure call is the same as if the formal parameters were substituted into the body of the procedure. This rule for defining the result of a procedure call by copying the procedure and substituting for the formal parameters is called the *Algol 60 copy rule.* While the copy rule works well for pure functional programs, as illustrated by β-reduction in lambda calculus, the interaction with side effects to the formal parameter may cause unexpected results.

The following Algol 60 code declares a procedure `P` with one pass-by-name integer parameter. The line `integer x` does not declare local variables – this is just Algol 60 syntax declaring the type of the procedure parameter.

```
begin
   integer i;
   integer array A[1:2];

   procedure P(x);
      integer x;
      begin
         i := x;
         x := i
      end

   i := 1;
   A[1] := 2; A[2] := 3;
   P (A[i]);
   print (i, A[1], A[2])
end
```

(a) (2 points) Explain how the procedure call `P(A[i])` changes the values of `i` and `A` by writing the lines of Algol that you get by substituting the actual parameter for the formal parameter in `P(x)`.

(b) (3 points) What integer values are printed by the program using pass-by-name parameter passing?

3. (6 points) *Control Flow and Memory Management.* An *exception* is a command that aborts part of a computation and transfers control to a handler that was established at some earlier point in the computation.

   (a) (3 points) Why is it easier to write programs with exceptions in a language that provides garbage collection than a language where all data on the heap is explicitly manipulated by the program?

   (b) (3 points) In a concurrent programming language, we have two options:
   (i) raising an exception only affects the current thread, or
   (ii) raising an exception aborts all threads that were spawned below the control point associated with the exception handler.

   Which option would be easier to implement? Which would be more convenient to use? Explain briefly. (This question is only worth 3 points, so do not write more than 3 or 4 sentences.)

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

   (a) (4 points) If a Smalltalk program sends message `m` to object `x`, how is the code implementing `m` located? Describe the main data structure and briefly explain how it is used.

   (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? Your answer should be 3–5 sentences.

Comprehensive Exam: Programming Languages Autumn 2002

1. (7 points) *Type Inference*

   (a) (3 points) This part of the question asks you to write the ML type of an ML function. Recall that an ML function that takes two integer arguments and returns a Boolean value has type `(int * int) -> bool`. What is the ML type for the following function?

   ```
   fun applytwice(f, x) = f(f(x));
   ```

   Answer: `('a -> 'a) * 'a -> 'a`

   (b) (2 points) Describe types `<type1>`, `<type2>` and `<type3>` that will make the analogous C code below statically typed. You may write the types using correct or mostly correct C syntax. In particular you can write `<type2> f` in a form where `f` appears in the middle of the type, if you prefer (and if this is correct).

   ```
   <type1> applytwice(<type2> f, <type3> x) {
             return f(f(x));
   };
   ```

   Answer: One possibility is `<type1>=<type3>=int` and `<type2>`= int (*) (int). Some prefer to write `int (* f)(int)`

   (c) (2 points) Why is the ML type more useful than the C type for `applytwice`? Specifically, what flexibility does the ML type system give, for functions with the kind of type you wrote in (a), that the C type system does not give you for the function in (b)?

   Answer: The ML function is polymorphic, which means that you can apply `applytwice` to many types of arguments. In contrast, the C function can only be declared and used for one specific choice of types.

2. (5 points) *ALGOL 60 Pass-By-Name*

   In pass-by-name, the result of a procedure call is the same as if the formal parameters were substituted into the body of the procedure. This rule for defining the result of a procedure call by copying the procedure and substituting for the formal parameters is called the *Algol 60 copy rule*. While the copy rule works well for pure functional programs, as illustrated by β-reduction in lambda calculus, the interaction with side effects to the formal parameter may cause unexpected results.

   The following Algol 60 code declares a procedure `P` with one pass-by-name integer parameter. The line `integer x` does not declare local variables – this is just Algol 60 syntax declaring the type of the procedure parameter.

```
begin
   integer i;
   integer array A[1:2];

   procedure P(x);
      integer x;
      begin
         i := x;
         x := i
      end

   i := 1;
   A[1] := 2; A[2] := 3;
   P (A[i]);
   print (i, A[1], A[2])
end
```

(a) (2 points) Explain how the procedure call `P(A[i])` changes the values of `i` and `A` by writing the lines of Algol that you get by substituting the actual parameter for the formal parameter in `P(x)`.

Answer: The program is equivalent to the following one, obtained by replacing the call with a copy of the procedure body in which each occurrence of the formal parameter `x` is replaced by the actual paramter `A[i]`:

```
begin
   integer i;
   integer array A[1:2];
   i := 1;
   A[1] := 2; A[2] := 3;
   begin
      i := A[i];
      A[i] := i
   end;
   print (i, A[1], A[2])
end
```

(b) (3 points) What integer values are printed by the program using pass-by-name parameter passing?

Answer: This block resulting from the call sets `i` to 2 and then sets `A[2]` to 2, so the output is 2, 2, 2.

3. (6 points) *Control Flow and Memory Management.* An *exception* is a command that aborts part of a computation and transfers control to a handler that was established at some earlier point in the computation.

   (a) (3 points) Why is it easier to write programs with exceptions in a language that provides garbage collection than a language where all data on the heap is explicitly manipulated by the program?

   Answer: An exception might casue a program to bypass the "normal" code that deallocated memory and cleans up after a portion of the computation. After an exception, deallocation would have to be performed by the exception handler. However, when the only pointers to garbage are pointers that were in activation records that were eliminated as a result of the exception, it will be impossible for the handler to explicitly deallocate unreachable memory.

   (b) (3 points) In a concurrent programming language, we have two options:
   (i) raising an exception only affects the current thread, or
   (ii) raising an exception aborts all threads that were spawned below the control point associated with the exception handler.

   Which option would be easier to implement? Which would be more convenient to use? Explain briefly. (This question is only worth 3 points, so do not write more than 3 or 4 sentences.)

   Answer: Option (i) is easier to implement but option (ii) may be more convenient in some cases. In general, if we are doing some parallel computation and an error occurs in one branch, then it may make most sense to abort the entire computation. But it is hard to cleanly abort all the subprocesses that might be running on different processors.

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

   (a) (4 points) If a Smalltalk program sends message `m` to object `x`, how is the code implementing `m` located? Describe the main data structure and briefly explain how it is used.

   Answer: The method code for `m` is located by searching the method dictionary of the class of `x`, then any superclass of the class of `x`, then the method dictionary of any superclass of any superclass of the class of `x`, and so on. A Smalltalk *method dictionary* generally provides an association between method names (selectors) and method code, together with a pointer to any superclass method dictionaries.

   (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.

Answer: The main data structure is the *virtual function table* or vtable. The vtable is a structure of pointers to function bodies. The offset of a virtual member function is calculated at compile time, so that at run time it is only necessary to follow the pointer located by adding the vtable address and the offset.

(c) (4 points) Describe the trade-off's between the two approaches. Which is more flexible? Which is more efficient? Your answer should be 3–5 sentences.

Answer: The C++ mechanism is generally more efficient, but it requires compile-time type information. Smalltalk is a more flexible language, since it doesn't have compile-time type checking, but method lookup is less efficient. (Caching can be used to speed up the Smalltalk lookup, however.) Another factor is that if a class is changed, a C++ derived class or client program may need to be recompiled, since the compile-time-computed offsets may change. However, no such recompilation is needed in Smalltalk since the offsets are determined at run time.

Stanford University Computer Science Department

Fall 2002 Comprehensive Exam in Software Systems

1. **CLOSED BOOK**: no notes, textbook, computer, Internet access, etc.
2. **WRITE ONLY IN BLUE BOOKS: No credit** for answers written on these exam pages.
3. **WRITE MAGIC NUMBER** on the cover of **EACH** blue book.
4. The exam is designed to take less than an hour. Most answers should be quite short.
5. If you need to make assumptions to answer a question, *state them clearly*.
6. For questions that refer generically to "the OS": if you think the answer would differ substantially depending on the OS in question, assume any popular flavor of Unix (Linux, BSD, Solaris, etc.).

Short Answers (3 each, 30 total)

1) With respect to transmission of a message from principal A to principal B, distinguish what is meant by authenticity, privacy, and integrity with respect to that message.

2) What is the point of multilevel paging?

3) Give an example of a high-level language that uses explicit memory management, and state what facilities are provided for doing it.

4) What is an idempotent operation? Give an example of an idempotent operation, and contrast it with an example of a non-idempotent operation.

5) Describe how failure recovery can be made easier by the use of idempotent operations.

6) What's the difference between a preemptive and a non-preemptive process scheduler?

7) If a physical page is shared between two different processes, is it possible for the page to be read-write for one process and read-only for the other? If so, how, and if not, why not?

8) Describe the difference between memory-mapped I/O and programmed I/O. Mention one typical application of each.

9) What is the difference between deadlock and starvation in a resource-allocation system?

10) A RAID system can fail if two or more of its drives crash within the same hour. Suppose that in a given hour, a drive can fail with probability p, and that drive failures are independent. What is the probability of a k-drive RAID system failing in a given hour?

Slightly Longer Answers (5 each, 30 total)

11) When a user program makes a (privileged) system call, in what important way(s) is the calling sequence different from that of calling another user-level procedure?

12) Describe the *priority inversion* problem that arises in process scheduling or thread scheduling. Give a concrete example of circumstances that might cause priority inversion to occur.

13) Suppose you have a word-sized variable *Foo* that is shared among many concurrent threads. You would like to serialize accesses to this variable, to prevent conflicting updates. Unfortunately, you're writing in a language that doesn't provide language-level constructs for writing a monitor procedure (i.e., doesn't have features like Java's *synchronized* keyword).
(a) Assume your OS provides a mutex/locking facility. Using any suitable pseudocode syntax, write the pseudocode for *updateFoo(int newFoo)* using mutexes and/or locks.
(b) Assume your OS provides simple nonblocking atomic operations, like test-and-set or compare-and-swap, on word-sized operands. Write the pseudocode for *updateFoo(int newFoo)* using nonblocking atomic operations. (Note! For part (b), *do not* use nonblocking operations to implement locks! Write true nonblocking code.)

14) Suppose now that *Foo* is not an int but a data structure *FooStruct* that is several words long, and *updateFoo(FooStruct newFoo)* must atomically update the entire contents of this data structure. Can you still write only nonblocking code for *updateFoo*? If so, show the pseudocode. If not, explain why it can't be done. (Note: as in part (b) of the previous question, using atomic instructions to implement locks *does not* count as nonblocking code! Nonblocking code avoids spinwaits.)

15) Suppose that a CPU scheduling algorithm favors those processes that have used the least processor time in the recent past. Explain why this algorithm will favor I/O-bound processes; then explain why it will *not* permanently starve CPU-bound processes.

16) Typically, a true LRU replacement policy can be used for managing blocks in the buffer cache of the file system, whereas only approximations of LRU are used to keep track of pages in a virtual memory system. Why is this so?

17) Suppose a new CPU design has the following feature: there are multiple copies of the register file; at any given time, exactly one of these copies or "banks" can be accessed by user code; and a privileged instruction must be executed to change which "bank" is being used. How might the OS exploit such a CPU feature, and what benefit(s) would be gained?

Stanford University Computer Science Department
Fall 2002 Comprehensive Exam in Software Systems

Solutions by CS Ph.D. First Years (2006-10-19)

Short Answers (3 each, 30 total)

1.) With respect to transmission of a message from principal A to principal B, distinguish what is meant by authenticity, privacy, and integrity with respect to that message.

Authenticity – Was it really A who sent the message to B?
Privacy – Could a third party snoop and see the contents of the message or find out that the message was being sent?
Integrity – Did a third party alter the message while it was in transit?

2.) What is the point of multilevel paging?

To save a tremendous amount of space. Page tables are mostly empty; they are sparse data structures. If there were only one level, most slots would be blank.

3.) Give an example of a high-level language that uses explicit memory management, and state what facilities are provided for doing it.

C++. The **new** and **delete** operators are used to manage memory.

4.) What is an idempotent operation? Give an example of an idempotent operation, and contrast it with an example of a non-idempotent operation.

An idempotent operation is one whose results are identical no matter if it's executed once or many times. An example is 'x = x * 0'. A non-idempotent operation is 'x = x + 1'.

5.) Describe how failure recovery can be made easier by the use of idempotent operations.

If you experience a failure while you're in the middle of running the recovery code, there is no harm in running idempotent failure recovery operations again. This simplifies recovery code because it doesn't need to consider what happens if there is a failure during the recovery process itself.

6.) What's the difference between a preemptive and a non-preemptive process scheduler?

A preemptive scheduler forces processes to relinquish control of the CPU at certain times, while a non-preemptive scheduler doesn't; in a non-preemptive scheduler, the only way for some other process B to run while process A is running is if process A purposely yields control of the CPU.

7.) If a physical page is shared between two different processes, is it possible for the page to be read-write for one process and read-only for the other? If so, how, and if not, why not?

Yes, the page table entries for the two processes can point to the same physical page (usually mapped to different virtual addresses), and within each entry are permissions bits specific to the process. One process's entry for that page could be marked 'read-write' while another one's could be marked 'read-only'.

8.) Describe the difference between memory-mapped I/O and programmed I/O. Mention one typical application of each.

Memory-mapped I/O works by mapping the device into special memory addresses so that applications can interact with those devices by reading/writing those addresses. An example is a video card.

Programmed I/O requires special CPU instructions to read/write to a certain port and is not done through the memory system. Examples are legacy hard drives and mechanisms to write to a computer's CMOS memory.

9.) What is the difference between deadlock and starvation in a resource-allocation system?

Deadlock is when nobody gets a chance to run at all. Starvation is when one party never gets to run (or rarely gets to run) while other parties are running just fine. Starvation may be temporary and get relieved as time progresses (perhaps as load decreases), but deadlock is, by definition, permanent (until the system resets).

10.) A RAID system can fail if two or more of its drives crash within the same hour. Suppose that in a given hour, a drive can fail with probability *p*, and that drive failures are independent. What is the probability of a *k*-drive RAID system failing in a given hour?

P(*k*-drive RAID system failing in some hour) = P(2 or more drives crash within that hour)

P(2 or more drives crash) = 1 – P(exactly 1 drive crashes) – P(no drives crash)

P(exactly 1 drive crashes) = $k * p * (1 - p)^{k-1}$
P(no drives crash) = $(1 - p)^k$

Thus, P(failure) = P(2 or more drives crash) = $1 - k * p * (1 - p)^{k-1} - (1 - p)^k$

Slightly Longer Answers (5 each, 30 total)

11.) When a user program makes a (privileged) system call, in what important way(s) is the calling sequence different from that of calling another user-level procedure.

When calling a regular user-level procedure, the program saves caller-save registers onto the stack, pushes the return address and arguments onto the stack, and directly jumps into the code of the called procedure. However, when making a system call, the program sets up some register state and simply issues an interrupt to trap into the kernel. The kernel starts executing in a privileged state, saves the process's state so that it can be restored later, does whatever the system call requires, and returns back to the user program in a non-privileged state.

12.) Describe the *priority inversion* problem that arises in process scheduling or thread scheduling. Give a concrete example of circumstances that might cause priority inversion to occur.

Priority inversion is when a lower-priority process holds a lock (or some other resource) that a higher-priority process is waiting on, so the higher-priority process doesn't get to execute, thus inverting their priorities. A concrete example is the following: Suppose that there is a high priority process H and a low priority process L. L grabs a lock and then H tries to grab it. H can't run because L has the lock. However, now a medium priority process M appears, and because it has higher priority than L, it runs more often. L will eventually finish what it's doing and release the lock, but in the meantime, M runs much more often than H, even though it has lower priority than H, hence the priority inversion.

13.) Suppose you have a word-sized variable *Foo* that is shared among many concurrent threads. You would like to serialize accesses to this variable, to prevent conflicting updates. Unfortunately, you're writing in a language that doesn't provide language-level constructs for writing a monitor procedure (i.e., doesn't have features like Java's *synchronized* keyword).

(a) Assume your OS provides a mutex/locking facility. Using any suitable pseudocode syntax, write the pseudocode for *updateFoo(int newFoo)* using mutexes and/or locks.

```
int Foo;
mutex fooMutex;
updateFoo(int newFoo) {
  lock(&fooMutex);
  Foo = newFoo;
  unlock(&fooMutex);
}
```

(b) Assume your OS provides simple nonblocking atomic operations, like test-and-set or compare-and-swap, on word-sized operands. Write the pseudocode for *updateFoo(int*

newFoo) using nonblocking atomic operations. (Note! For part (b), *do not* use nonblocking operations to implement locks! Write true nonblocking code.)

```
int Foo;
updateFoo(int newFoo) {
  CAS(&Foo, Foo, newFoo); // compare-and-swap
}
```

14.) Suppose now that *Foo* is not an int but a data structure *FooStruct* that is several words long, and *updateFoo(FooStruct newFoo)* must atomically update the entire contents of this data structure. Can you still write only nonblocking code for *updateFoo*? If so, show the pseudocode. If not, explain why it can't be done. (Note: as in part (b) of the previous question, using atomic instructions to implement locks *does not* count as nonblocking code! Nonblocking code avoids spinwaits.)

No, because there is no way to atomically write more than 1 word. There might be interruptions from other threads while a *FooStruct* is in the process of being updated.

15.) Suppose that a CPU scheduling algorithm favors those processes that have used the least processor time in the recent past. Explain why this algorithm will favor I/O-bound processes; then explain why it will *not* permanently starve CPU-bound processes.

It will favor I/O-bound processes because those are likely to be blocking (waiting) on I/O lots of the time, and hence qualify for 'using the least processor time in the recent past.' However, it will not starve CPU-bound processes because after those I/O-bound processes get a chance to run for a bit, the CPU-bound processes now quality as 'using the least processor time in the recent past.' Also, because those processes are I/O-bound, they are likely to just do a little bit of work and then block for a long time, giving plenty of time for the CPU-bound processes to execute.

16.) Typically, a true LRU replacement policy can be used for managing blocks in the buffer cache of the file system, whereas only approximations of LRU are used to keep track of pages in a virtual memory system. Why is this so?

Speed. True LRU requires timestamps and is much more costly in terms of speed than an LRU approximation (such as the clock algorithm). When dealing with the buffer cache, the order of magnitude in speed is bounded by hard disk access times, which are much slower than CPU calculation times, so it's reasonable to implement true LRU. However, when dealing with virtual memory, which must be accessed more often and needs to be much faster, true LRU is simply too slow.

17.) Suppose a new CPU design has the following feature: there are multiple copies of the register file; at any given time, exactly one of these copies or "banks" can be accessed by user code; and a privileged instruction must be executed to change which "bank" is being used. How might the OS exploit such a CPU feature, and what benefit(s) would be gained?

The OS can exploit this feature to perform context switches more efficiently. Normally, to perform a context switch, the registers for the exiting process must be saved away in memory, usually somewhere in its own stack, and the register values for the entering process must be loaded from memory into the actual registers in the register file. This can be a slow operation. With multiple register files, the OS can simply execute a privileged instruction to switch banks when processes are being switched.