Comprehensive Exam: Algorithms and Concrete Mathematics Autumn 2003

1. **[10 pts]** Prove a tight asymptotic bound on the behavior of $T(n) = T(n/3) + T(n/9) + n^3$, where $T(n) \leq b$ for $n < n_0$, where n_0 and b are some given constants. For simplicity, solve for n being a power of 9.

   Simplest approach is to guess $T(n) = \Theta(n^3)$ and prove by induction. First notice that $T(n) = \Omega(n^3)$ just by examining the recurrence. Now assume $T(n) \leq cn^3$ for some constant c. This constant should be large enough to satisfy initial conditions. Now the inductive step is:

   $$T(n) = T(n/3) + T(n/9) + n^3 \leq c(n/3)^3 + c(n/9)^3 + n^3 = cn^3(1/3^3 + 1/9^3 + 1/c)$$

   The claim follows since the expression in the parenthesis is smaller than 1 for large enough c.

2. **[10 pts]** Given integers $a_1, a_2, \ldots, a_n$, give a randomized algorithm that outputs all pairs (i, j) such that $a_i = a_j$. Your algorithm should have expected running time $O(n+K)$ where K is the number of pairs output. Prove the upper bound on the running time.

   Use hashing with chaining, with size of the hash table constant factor larger than n. Moreover, construct each chain as a chain of linked lists, each list holding equal elements. In other words, if a_j hashes into position q, look through the linked list and find the list that corresponds to the value of a_j. Then, for each element a_k of this list, output (a_j, a_k). The claim follows since the expected number of lists attached to a single hash cell is constant.

3. **[10 pts]** Assume that you are given a "black box" implementation of a comparisons-based data structure that supports "extract minimum" and "insert element". You are also told that such data structure can be constructed on n elements in time $\Theta(n \log\log n)$. Let $g(n)$ be the amortized time it takes to execute "extract min", i.e. extract the smallest element from the above data structure and delete it from the data structure. Is it possible that $g(n) = \log\log n$? What is the (asymptotically) fastest possible $g(n)$?

   By using this black-box, one can sort by building the structure, and using extract-min n times. Since total time for sorting in comparisons-only model is $\Omega(n \log n)$, the extraction should take at least $\Omega(\log n)$ amortized per extracted element.

4. **[15 pts]** You are given a graph $G = (V, E)$ with non-negative weights on edges, i.e. $w : E \to R^+$. You are also told that all weights are distinct, i.e. $\forall e_1, e_2 \in E, e_1 \neq e_2 : w(e_1) \neq w(e_2)$.

   (a) **[10 pts]** Prove that the Minimum Spanning Tree in G is unique.

   (b) **[5 pts]** Say you succeeded in proving (a). Now assume that you were given tree T that is the minimum spanning tree for G with weight w. Explain how to use T in order to compute T', which is a minimum spanning tree for G with weights $w'(e) = [w(e)]^2, \forall e \in E$.

a. Let T be some MST. Consider Kruskal's MST algorithm executing on the given data to construct T'. We will show that T and T' are identical. Assume not. Now let e be the "first difference". More precisely, the first edge where Kruskal's decision was different from T. Observe that it is not possible that the first difference is of the form "Kruskal's algorithm rejected e, but e is in T". This is due to the fact that up until e, all edges taken by Kruskal's algorithm are also in T. Thus, the only reason Kruskal's algorithm rejected e was because it closes a cycle, which means that e can not belong to T either.

So the only conclusion is that e is not in T but it is in T'. Add e to T and consider the created cycle. Note that at least one of the edges on this cycle (call it e') was considered *after* e, since when e was considered, Kruskal's algorithm acceped it, which means it did not close a cycle at that moment. Thus, by adding e and deleting e', we improved the weight of T (recall that all weights are different), leading to a contradiction.

b. Since MST is unique, we can assume that the given tree T was constructed by Kruskal's algorithm running on the original weights. Observe that squaring of the weights does not change the sorted order of the edges, since all weights are positive. Hence, Kruskal's algorithm will still produce T, even after the edge weights are squared.

5. **[15 pts]** You are given n villages situated along a highway. Let $x_1, x_2, \ldots, x_n$ represent the positions of these villages on the highway (the highway is a straight line). We need to build hospitals in k of these villages. Let d_i denote the distance from the i-th village to the nearest hospital, and let $D = d_1 + d_2 + \ldots + d_n$. Our goal is to minimize D.

   (a) **[5 pts]** Give an efficient algorithm to solve the problem for $k = 1$.

   (b) **[5 pts]** Consider an optimum placement for some k. Prove that this optimum solution can be viewed as consisting of two optimum solutions for smaller problems, where one problem is optimum placement of $k - 1$ hospitals to cover some villages and the other problem is placing a single hospital to cover the rest of the villages. State the "smaller problems" completely.

   (c) **[5 pts]** Give a polynomial time algorithm to find the smallest possible value of D for a given k.

a. Median-index village is the answer to this question. To see this, consider a hospital with q_1 villages to the left of the hospital and q_2 villages to the right, with $n = q_1 + 1 + q_2$. Now move the hospital one village left, say x miles. All villages on the left improve distance by x, while all villages on the right (including the one where the hospital was up until now) increase their distance by the same x. Total change in D is $-q_1 x + x + q_2 x$, which is a gain as long as $q_1 > q_2$.

b. First of all, assume that the villages are sorted with x_1 being the leftmost and x_n being the rightmost. If they are not sorted, then sort and rename.

Since villages are assigned to closest hospital when computing D, the set of villages assigned to a specific hospital "has no holes", i.e. if x_i and x_j are assigned to a specific hospital, then all villages between x_i and x_j are assigned to the same hospital as well.

In particular, consider optimum solution, and consider all the villages assigned to the rightmost hospital. Per our discussion above, these are villages starting from some index

$q+1$ and up to n. The rest of hospitals ($k-1$ of them) are covering villages x_1 through x_q. We claim that these $k-1$ hospitals are the best solution to cover villages x_1 through x_q with $k-1$ hospitals. If not, then take the better solution, add the k-th hospital with its villages, and you will get a better solution than the original optimum one, which is a contradiction.

c. Let $D(q,p)$ be the best way to cover villages x_1 through x_q using p hospitals. Previous discussion implies that in order to compute this value, we should compare all values $D(i,p-1)+Q(i+1,q)$, where $Q(s,t)$ is the best cost of using a single hospital (in the best possible way as per (a)) to cover villages x_s through x_t.

Observe that we are looking for $D(n,k)$, which is the answer to the posed question. This value can be computed using dynamic programming. First, one can compute $D(i,1)$ for all i using median as in (a). Then one can compute $D(i,2)$ for all i, $D(i,3)$, etc. There are nk elements in the dynamic programming table, with each taking $O(n)$ to compute, leading to $O(n^2k)$ algorithm.

Stanford University
Computer Science Department

Fall 2003 Comprehensive Exam in Artificial Intelligence

1. **Closed Book, No Laptop & Notes Write only in the Blue Book provided.**
2. **The exam is timed for one hour.**
3. **Write your Magic Number on this sheet & on the Blue Book.**

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

A. The Honor Code is an undertaking of the students, individually and collectively:
   1. that they will not give or receive aid in examinations; that they will not give or receive 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----------

2003 Comprehensive Examination
Artificial Intelligence

1. Search. (20 points) Consider a search tree with uniform branching factor 2 and depth d, and consider a search problem for which there is a single solution in the tree at depth k. Give expressions for the worst case cost of finding the solution, in terms of nodes visited, for (a) breadth-first search, (b) depth-first search, and (c) iterative deepening starting at depth 0 and incrementing by 1 on each iteration. Comments on dealing with fence-posts: The root of the tree is at depth 0, and the cost of finding a solution includes the visit to the solution node.

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

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

$$\forall x.(p(x) \Rightarrow \exists y.(q(x,y) \vee q(y,x)))$$
$$\forall x.\forall y.((q(x,y) \vee q(y,x)) \Rightarrow r(x))$$

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

4. Probability. (20 points) Suppose there are three chests, each with two drawers. The first chest has a gold coin in each drawer. The second has a gold coin in one drawer and a silver coin in the other drawer. The third chest has a silver coin in each drawer.

(a) Three of the six drawers are selected at random. What is the probability that all three drawers contain gold coins?
(b) A chest is chosen at random, and a drawer is opened. If the drawer contains a gold coin, what is the probability that the other drawer also contains a gold coin? Be careful.

5. Learning. (20 points) Consider the following training set for a decision-tree learning problem.

Example	*a*	*b*	*c*	*d*	*e*	Goal
x_1	1	0	0	0	0	1
x_2	1	1	1	1	0	0
x_3	0	0	0	1	0	1
x_4	1	1	1	0	0	1
x_5	1	0	0	1	0	0
x_6	0	0	0	0	0	0

(a) Draw a decision tree of minimal depth that correctly classifies the examples in this dataset.

(b) How much information is needed to classify an example in this case? (Reminder: the amount of information needed to classify an example is $-(p*\log p)-(n*\log n)$, where p is the probability of a positive answer and n is the probability of a negative answer.)

(c) How much information is needed to classify an example *given* that a is 1? What if a is 0?

(d) So, what is the information gain from attribute a? What is the information gain from d given a?

AI Comp 2003 solutions (by CS PhD 2006)

Peerapong Dhangwatnotai

October 25, 2006

1 Search

a) $2^{k+1} - 1$
b) $2^{d+1} - 2^{d-k+1} + 1$ (complete tree except descendants of the solution node)
c) $2^{k+2} - k - 3$. We can get this from summing up the cost of breadth-first search with max depth = 0, 1, ..., k.

$$\begin{aligned}
\sum_{i=0}^{k} \text{complete tree of depth i} &= \sum_{i=0}^{k} 2^{i+1} - 1 && (1)\\
&= \left(\sum_{i=0}^{k+1} 2^i\right) - 1 - \sum_{i=0}^{k} 1 && (2)\\
&= 2^{k+2} - 1 - 1 - (k+1) && (3)\\
&= 2^{k+2} - k - 3 && (4)
\end{aligned}$$

2 Logic

a) True.
b) False. $\phi = p \vee q$. $\Gamma = \{p\}$. $\Delta = \{q\}$
c) True.
d) True.
e) True. Tautology

3 Automated Reasoning

1. $\neg\forall x(p(x) \Rightarrow r(x))$ (negate the conclusion and add to our premises)
2. $\forall x(p(x) \Rightarrow \exists y(q(x,y) \vee q(y,x)))$ (premise)
3. $\forall x\forall y((q(x,y) \vee q(y,x)) \Rightarrow r(x))$ (premise)
4. $\neg\forall x\neg p(x) \vee r(x)$ (eliminate implication from 1)
5. $\forall x(\neg p(x) \vee \exists y(q(x,y) \vee q(y,x)))$ (eliminate implication from 2)
6. $\forall x\forall y(\neg(q(x,y) \vee q(y,x)) \vee r(x))$ (eliminate implication from 3)
7. $\exists x(p(x) \wedge \neg r(x))$ (push negation inside, 4)
8. $\forall x\forall y((\neg q(x,y) \wedge \neg q(y,x)) \vee r(x))$ (push negation inside, 6)
9. $\neg p(x) \vee (q(x, f(x)) \vee q(f(x), x))$ (skolemize, 5)

10. $(p(C) \wedge \neg r(C))$ (skolemize, 7)
11. $((\neg q(x, f(x)) \wedge \neg q(f(x), x)) \vee r(x))$ (skolemize, 8)
12. $(\neg q(x, f(x)) \vee r(x)) \wedge (\neg q(f(x), x) \vee r(x))$ (distribute $\vee$, 11)
13. $p(C)$ (split up $\wedge$, 10)
14. $\neg r(C)$ (split up $\wedge$, 10)
15. $(\neg q(x, f(x)) \vee r(x))$ (split up $\wedge$, 12)
16. $(\neg q(f(x), x) \vee r(x))$ (split up $\wedge$, 12)
17. $\neg q(C, f(C))$ (resolving 15 and 14 by setting x to C)
18. $\neg q(f(C), C)$ (resolving 16 and 14 by setting x to C)
19. $q(C, f(C)) \vee q(f(C), C)$ (resolving 9 and 13 by setting x to C)
20. $q(f(C), C)$ (resolving 19 and 17)
21. $\perp$(false) (resolving 18 and 20)
22. $\forall x(p(x) \Rightarrow r(x))$ (Negation of one of the premises is true)

4 Probability

a) $1/\binom{6}{3}$
b)

$$p(d_2 = g|d_1 = g) = \frac{p(d_2 = g, d_1 = g)}{p(d_1 = g)} \quad (5)$$

$$= \frac{1/3}{1/3 \cdot 1 + 1/3 \cdot 1/2} \quad (6)$$

$$= \frac{2}{3} \quad (7)$$

5 Learning

a) There are two answers. One makes "a" the root node and "d" the child nodes. But you can switch them and make "d" the root node. The tree has depth 1 (beginning with depth 0).
b) Since the probability that "Goal" is 1 is $\frac{1}{2}$, $p = \frac{1}{2}$ and $n = \frac{1}{2}$. Therefore, $-(p * logp) - (n * logn) = 1$.
c) If a is 1, the probability that Goal is 1 is $\frac{1}{2}$. If a is 0, the probability that Goal is 1 is $\frac{1}{2}$. Therefore, information needed is 1 for both cases.
d) Info gain from a is 0. Since the tree correctly classifies all examples, information gain from d given a is 1 - (info gain from a) = 1.

Comprehensive Exam Solutions Autumn 2003-04

Automata and Formal Languages (60 points)

Problem 1. [10 points]

Consider the following grammar G over the alphabet $\Sigma = \{0,1\}$, where S is the start symbol of the grammar.

$S \rightarrow \varepsilon \mid 0T \mid 1U$
$T \rightarrow 0T \mid 1S$
$U \rightarrow 0S$

1. [2 points] Give a derivation of the string 1001.

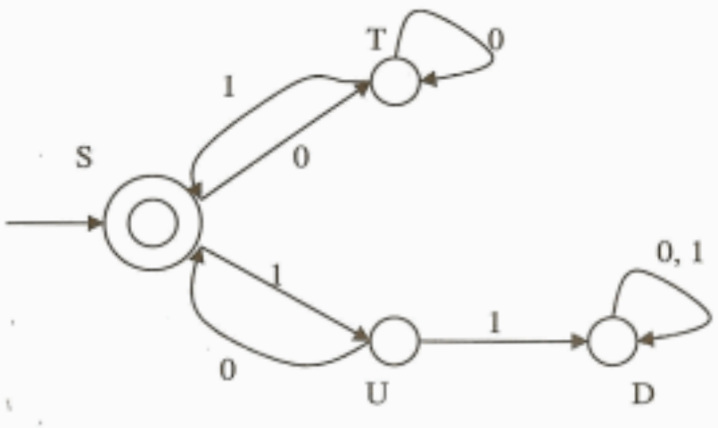
2. [4 points] Give a deterministic finite automaton for the language of G.

3. [4 points] Give a regular expression for the language of G.

Solution.

1. $S \Rightarrow 1U \Rightarrow 10S \Rightarrow 100T \Rightarrow 1001S \Rightarrow 1001$

2. The DFA is shown in the following figure. It is ok to omit state D.



3. $(10 + 00^*1)^*$

Problem 2. [10 points]
For each of the following statements, write TRUE if the statement is true for all languages L,M that satisfy the hypothesis; otherwise, write FALSE. You will receive 2 points for each correct answer and –2 points for each incorrect answer.

1. If L is a nonregular language and M is a regular language then their concatenation LM is not regular. (Recall that the concatenation is $LM = \{xy \mid x \in L, y \in M\}$.)

2. All finite languages L are regular.

3. All context-free languages L are in the class P (Polynomial Time).

4. If L, M are languages in PSPACE then their difference L-M is also in PSPACE. (Recall that $L - M = \{x \mid x \in L \text{ and } x \notin M\}$.)

5. If L, M are recursively enumerable languages then L-M is also recursively enumerable.

Solution.

1. FALSE

2. TRUE

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, all languages*. 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. The set of strings over $\{0,1\}$ in which the number of 0's is divisible by 5.

2. The set of strings over $\{a,b,c\}$ that contain the same number of a's, b's and c's.

3. $L = \{a^i b^j c^k \mid j = i + k,\ i, j, k \geq 0\}$.

4. The set of encodings of deterministic Turing machines M such that $L(M) \neq \varnothing$.

5. The language $L = \{x \in \{0,1\}^* \mid xx \in M\}$ where M is a regular language.

Solution.

1. regular

2. recursive

3. context-free

4. recursively enumerable

5. regular

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*: Context free grammar G and deterministic finite automaton A.
   Question: $L(G) \cap L(A) = \varnothing$?

2. *Input*: Context free grammar G and deterministic finite automaton A over alphabet Σ.
   Question: $L(G) \cup L(A) = \Sigma^*$?

3. *Input*: Deterministic finite automaton A.
   Question: Is $L(A)$ finite ?

4. *Input*: Encoding of a deterministic Turing machine M, input string w of length n.
   Question: Does M use more than n^2 space on input w ?

Solution.

1. in P

2. undecidable

3. in P

4. decidable but not known to be in P

Problem 5. [13 points]

a. [10 points] Give a polynomial time reduction from the SATISFIABILITY problem to the following LINEAR INTEGER PROGRAMMING PROBLEM

Input: A set of linear inequalities of the form $\sum_{i=1}^{n} a_i x_i \geq c$ or $\sum_{i=1}^{n} a_i x_i \leq c$, where the a_i's and c are integer constants and $x_1, x_2, \cdots x_n$ are variables.

Question: Does there exist an assignment of integers to each of the variables that makes all the inequalities true?

Justify briefly the correctness of your reduction.

b. [3 points] Can you conclude from part (a) that LINEAR INTEGER PROGRAMMING is an NP-complete problem? Justify briefly your answer.

Solution.

a. We are given an instance F of the SATISFIABILITY problem consisting of m clauses $C_1, \ldots, C_m$ over n Boolean variables $y_1, \cdots, y_n$, where each clause is a disjunction of literals. Construct an instance L of the LINEAR INTEGER PROGRAMMING problem as follows. We have an integer variable x_j for each Boolean variable y_j, $j = 1, \cdots, n$ of F.

There are two inequalities for each variable and one inequality for each clause of F. The inequalities corresponding to variable y_j are $x_j \geq 0$ and $x_j \leq 1$.

The inequality corresponding to clause C_i is a follows:

$\sum\{x_j \mid y_j \text{ appears positively in } C_i\} + \sum\{(1 - x_j) \mid y_j \text{ appears negated in } C_i\} \geq 1$.

We claim that F is satisfiable *iff* L has an integral solution.

(*if*) Suppose that F is satisfiable and let τ be a satisfying truth assignment. Set each variable x_j equal to 1 if y_j is true in τ and set $x_j = 0$ otherwise. Since τ satisfies F, it follows that all inequalities of L are satisfied.

(*only if*) Suppose that L has an integral solution. Then every variable x_j is set to 0 or 1. Define a truth assignment for the variables of F by setting y_j to true if $x_j = 1$ and to false if $x_j = 0$. Every clause C_i of F contains a true literal because otherwise the corresponding inequality of L would be violated (the left hand side would be 0).

b. Part a is not enough to conclude that LINEAR INTEGER PROGRAMMING is NP-complete; it shows that the problem is NP-hard. For NP-completeness we need to show also that LINEAR INTEGER PROGRAMMING is in NP.

This is in fact one of the few problems where membership in NP is not completely trivial. (The reason it's nontrivial is that an argument has to be made that if there exists a solution, then there is one where all the variables are assigned integers whose number of bits is polynomially bounded in the size of the input.)

But the problem in the comp did not ask to prove actually this part.

Compilers Comprehensive Exam
Fall 2003

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

1. (5 points) Assume we have a statically typed language with polymorphic types and type inference (in the style of ML or Haskell). In one or two sentences explain how a self-application, such as

   $$f(f)$$

   is typed. Depending on what assumptions you make, you can answer this question either so that type inference succeeds or that it fails, but in either case you should pinpoint why it succeeds or fails.

2. (5 points) Consider the following nested loops (written in C). In one sentence give one reason an optimizer might choose to transform the first loop nest into the second. In one sentence give one reason an optimizer might choose to transform the second loop nest into the first.

```
for(i = 0; i < a, i++)
    for(j = 0; j < b; j++)
        A[i,j] = A[i][j+1] * 2;

for(j = 0; j < b; j++)
    for(i = 0; i < a, i++)
        A[i,j] = A[i][j+1] * 2;
```

3. (6 points) Consider the following flex-like specification. Parentheses are used to show the association of operations and are not part of the

input alphabet.

$\texttt{aa}^*$	{ `return Token1;` }
$\texttt{c(a}\vert\texttt{b)}^*$	{ `return Token2;` }
$\texttt{ab}^*\texttt{c}$	{ `return Token3;` }
$\texttt{caa}^*$	{ `return Token4;` }
$\texttt{b}^*\texttt{aa}^*(\texttt{c}\vert\epsilon)$	{ `return Token5;` }

Show how the following string is partitioned into tokens. Label each lexeme with the integer of the correct token class.

`abcabcaabbaacccabaccbb`

4. (4 points) In one or two sentences explain why a bottom-up parser can handle the following grammar while a top-down parser cannot:

$$\begin{aligned} \text{Loop} \quad &\rightarrow \quad \textbf{do}\ \text{stmt}\ \textbf{while}\ \text{expr} \\ &\mid \quad \textbf{do}\ \text{stmt}\ \textbf{until}\ \text{expr} \\ &\mid \quad \textbf{do}\ \text{stmt}\ \textbf{forever} \end{aligned}$$

5. (10 points) Below are the "action" and "goto" tables for an LR parser. The "goto" table includes only moves of the parsing automaton on non-terminals; the moves on terminals are encoded in the shift moves of the "action" table. The actions should be interpreted as follows:

   - $s(n)$ shifts the input and goes to state n.
   - $r(n, T)$ pops n elements off of the stack and pushes the non-terminal T onto the stack. An r-action is a reduce move, given in a non-standard way.
   - *acc* means accept.
   - A blank is an error entry.

   The non-terminals of the grammar from which these tables were generated are A, B, and C. No two productions for A have the same number of symbols on the right-hand side; similarly, all productions for B and C have different lengths.

   What is the grammar from which these tables were produced?

State	*action*						*goto*		
	a	b	c	d	e	$	A	B	C
0	s(5)			s(4)			1	2	3
1		s(6)				acc			
2		r(1,A)	s(7)		r(1,A)	r(1,A)			
3		r(1,B)	r(1,B)		r(1,B)	r(1,B)			
4	s(5)			s(4)			8	2	3
5		r(1,C)	r(1,C)		r(1,C)	r(1,C)			
6	s(5)			s(4)				9	3
7	s(5)			s(4)					10
8		s(6)			s(11)				
9		r(3,A)	s(7)		r(3,A)	r(3,A)			
10		r(3,B)	r(3,B)		r(3,B)	r(3,B)			
11		r(3,C)	r(3,C)		r(3,C)	r(3,C)			

Compilers Comprehensive Exam
Fall 2003

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

1. (5 points) Assume we have a statically typed language with polymorphic types and type inference (in the style of ML or Haskell). In one or two sentences explain how a self-application, such as

   $$f(f)$$

   is typed. Depending on what assumptions you make, you can answer this question either so that type inference succeeds or that it fails, but in either case you should pinpoint why it succeeds or fails.

   Since f is a function, it has type a -> b for some unknowns a and b. Because f is also the argument, f must have type a. The type equation
   a -> b = a
   has no finite solution, but does have infinite solutions, which correspond to unification with the occurs check (for finite solutions) and without (for infinite solutions).

2. (5 points) Consider the following nested loops (written in C). In one sentence give one reason an optimizer might choose to transform the first loop nest into the second. In one sentence give one reason an optimizer might choose to transform the second loop nest into the first.

```
for(i = 0; i < a, i++)
    for(j = 0; j < b; j++)
        A[i][j] = A[i][j+1] * 2;
```

```
for(j = 0; j < b; j++)
    for(i = 0; i < a, i++)
        A[i][j] = A[i][j+1] * 2;
```

The first nested loop has the better cache performance, as memory elements are accessed in order. The second nested loop has no dependencies between iterations of the inner loop, and so is good for any number of other optimizations (e.g., instruction scheduling).

3. (6 points) Consider the following flex-like specification. Parentheses are used to show the association of operations and are not part of the input alphabet.

aa^*	{ return Token1; }
$c(a\|b)^*$	{ return Token2; }
ab^*c	{ return Token3; }
caa^*	{ return Token4; }
$b^*aa^*(c\|\epsilon)$	{ return Token5; }

Show how the following string is partitioned into tokens. Label each lexeme with the integer of the correct token class.

abcabcaabbaacccabaccbb

abc	abc	aa	bbaac	c	caba	c	cbb
3	3	1	5	2	2	2	2

4. (4 points) In one or two sentences explain why an LR(1) parser can handle the following grammar while a LL(1) parser cannot:

Loop → do stmt while expr
| do stmt until expr
| do stmt forever

A top-down parser must decide which production to use when it sees the terminal 'do', whereas a bottom-up parser makes this decision only after the entire right-hand side of the production is on the stack.

5. (10 points) Below are the "action" and "goto" tables for an LR parser. The "goto" table includes only moves of the parsing automaton on non-terminals; the moves on terminals are encoded in the shift moves of the "action" table. The actions should be interpreted as follows:

   - $s(n)$ shifts the input and goes to state n.
   - $r(n, T)$ pops n elements off of the stack and pushes the non-terminal T onto the stack. An r-action is a reduce move, given in a non-standard way.
   - *acc* means accept.
   - A blank is an error entry.

   The non-terminals of the grammar from which these tables were generated are A, B, and C. No two productions for A have the same number of symbols on the right-hand side; similarly, all productions for B and C have different lengths.

   What is the grammar from which these tables were produced?

State	*action*						*goto*		
	a	b	c	d	e	$	A	B	C
0	s(5)			s(4)			1	2	3
1		s(6)				acc			
2		r(1,A)	s(7)		r(1,A)	r(1,A)			
3		r(1,B)	r(1,B)		r(1,B)	r(1,B)			
4	s(5)			s(4)			8	2	3
5		r(1,C)	r(1,C)		r(1,C)	r(1,C)			
6	s(5)			s(4)				9	3
7	s(5)			s(4)					10
8		s(6)			s(11)				
9		r(3,A)	s(7)		r(3,A)	r(3,A)			
10		r(3,B)	r(3,B)		r(3,B)	r(3,B)			
11		r(3,C)	r(3,C)		r(3,C)	r(3,C)			

```
A -> A b B
A -> B
B -> B c C
B -> C
C -> d A e
C -> a
```

The essence of the problem is to discover what can be on the stack when a reduction is about to happen. One way to solve the problem is to reconstruct the parsing DFA from the table and read the moves. A simpler way is to reason as follows. In state 9 there is a reduce move r(3,A), so we know there is a production A -> XYZ for some X, Y, and Z. How could the DFA get into state 9? It could get there from a 'goto B' out of state 6. Therefore, Z = B. One way to get to state 6 is via a 'shift b' action from state 8, so Y = b. State 8 is reached from a goto on a reduce to A, so X = A and the production is A -> AbB.

The reasoning for the other productions is similar.

Computer Science Comprehensive Examination
Computer Architecture
[100 points]

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

Number: _____

Problem	Max Score	Your Score
1	33	
2	33	
3	34	
TOTAL	100	

Problem 1: Memory Systems [33 Points 4 points each + 1 free]

A. Compared to a 16K direct-mapped cache, what type of misses will a 16K fully associative cache have fewer of? Circle all that apply.

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

(b) conflict misses

B. Suppose that cache A, a 16K-byte four-way set associative cache, cache B, a 16K-byte direct-mapped cache, and cache C, a 4K-byte direct-mapped cache are all referenced with an identical address sequence. All caches use a true least-recently used (LRU) replacement policy. Which of the following statements are true: (circle all that apply)

(a) A will contain a superset of the data in B
(b) A will contain a superset of the data in C
(c) B will contain a superset of the data in A
(d) B will contain a superset of the data in C

(b) A contains a superset of C. Cache C is equivalent to one "way" of A .
and (d) B contains a superset of C since both are direct mapped and B is larger.

C. In a 64K-byte four-way set-associative cache with 128-byte blocks, how large is the index field used to address the cache array? (write down the number of bits)

7

D. If the cache of question 1.C 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)

26 = 40-7 (index) – 7 (block)

E. A processors needs to support multiple processes running simultaneously with complete isolation (each process cannot access the memory of any other process). What is the minimum set of features in the architecture required to implement this complete isolation? (Circle minimal subset)

(a) Atomic branch and entry to privileged mode (in which all of memory can be addressed)
(b) A single base and length (segment) register
(c) Multiple base and length (segment) registers, one per process
(d) A translation look-aside buffer with a trap to a privileged mode handler routine on miss.

(a) and (b) suffice, as does (d) by itself (either gets full credit)

F. Replacing a single memory bank with multiple interleaved memory banks has which of the following effects? (circle all that apply)

(a) Increases latency
(b) Increases bandwidth
(c) Increases reliability
(d) Decreases latency

(b)

G. A bus-based cache coherence protocol "snoops" the directory of each processor's level-2 cache on each bus access. For this scheme to work, what relationship between the level-1 cache and level-2 cache must be maintained? (one word)

inclusion

H. A cache coherent multiprocessor has four processors, each with a 16Mbyte direct-mapped level-2 cache that is organized into 128-byte lines. Each processor makes single-word (8 byte) references (read and write in equal numbers) to widely-spaced single word locations in memory in a repeating pattern of 128K references. The patterns of the processors are disjoint – there is no sharing – and small enough to entirely fit in the cache. Rank the following three cache coherence protocols in order the amount of memory or bus traffic generated after the first iteration through the patterns?

(a) No cache – all data fetched and written is from pages marked uncached.
(c) Write-through – all cache lines have two states I and V.
Lines in state V are always in sync with memory – no dirty lines.
(d) Write-back – lines have three states, I, V, and D. The first write to a V line invalidates all other copies and make the line exclusive (and dirty), the D state.

The important thing to note here is that the working set is 128K 128Byte lines, so the working set is 16Mbytes which fits in the cache.

Thus (d) has the least traffic followed by (c) and then (a). If the working set were larger than the cache, (a) would have had the least traffic.

Problem 2: Instruction-Set Architecture [33 Points]

After graduate school, you work for JIPS Technologies, a company that designs embedded MIPS systems optimized for Java applications. The main product of JIPS is a MIPS processor and a just-in-time (JIT) compiler. The processor uses the classical 5-stage pipeline. The compiler runs along with any Java application and translates on-demand its Java code to MIPS instructions. The JIT compiler is kept very simple in order to minimize its overhead. It does implement the stack storage defined by the Java architecture, but tries to eliminate the number of push and pop operations to the stack by allocating as many temporary results as possible to the registers available in the MIPS processor.

After analyzing the typical workload (application & JIT), you come up with the following frequencies and CPIs for each type of instructions supported by the processor:

Type	Frequency	CPI
Load	30%	1.6
Store	20%	1.4
Arithmetic operations	35%	1.0
Branches & jumps	15%	1.5

Your analysis also indicates that 50% of the stores are used to push values into the stack. 33.33% of the loads are used to pop values from the stack. Assume that all pushes and pops operate on 32b integer numbers.

1) [7 points] You decide to improve performance by adding push and pop instructions to the MIPS ISA:

```
push rs   # MEM[r30+4]←rs; r30←r30+4
pop  rs   # rs←MEM[r30];   r30←r30-4
```

Register r30 stores a pointer to the top of the stack by default. What additional resources will you need in the 5-stage pipeline to support the execution the two new instructions? Explain.

Pop writes both rs and r30. Hence, we need a 2nd write port to the register file.

No adder is needed for (r30+4) or (r30-4). Push and pop can perform this operation on the integer ALU in the EX stage of the pipeline.

2) [6 points] A colleague notices that `push` and `pop` are similar to loads and stores with autoincrement and autodecrement addressing respectively. Hence, she suggests introducing autoincrement/autodecrement addressing (in which the register used to provide the address is incremented/decremented as a side effect of the load or store instruction) to the MIPS ISA instead of `push` and `pop`. Explain to her why `push` and `pop` are cheaper to implement.

General autoincrement/autodecrement loads/stores have two disadvantages

- They require additional bits in the opcode to encode the increment/decrement value (register specifier or immediate)
- They require 2 integer ALUs: one for the effective address calculation (register + offset) and one for the autoincrement/autodecrement.

3) [6 points] The JIPS design group comes to the conclusion that the introduction **push** and **pop** will increase the clock cycle time of the processor by 5% due to the additional resourced needed to implement them. Calculate the overall performance improvement if the new instructions are added to the design.

$$Speedup = \frac{ExecutionTimeOld}{ExecutionTimeNew} = \frac{ICold * CPIold * CCTold}{ICnew * CPInew * CCTnew}$$

CPIold = Sum(individual CPIs) = 0.3*1.6+0.2*1.4+0.35*1+0.15*1.5 = 1.335
CCTnew = 1.05*CCTold

Each push/pop instruction will replace one store/load instruction in the original program and the arithmetic operation used to increment/decrement the stack pointer. Hence, the introduction of push/pop eliminates a number of arithmetic operations from the original program. Therefore:
ICnew = (1-#push-#pop)*ICold = (1-0.5*0.2*0.33*.3)*ICold = 0.8*ICold

The CPI of a push/pop instruction is equal to that of a store/load instruction in the original program. However, our CPInew calculation must take into account that we eliminated some of the arithmetic instructions in the original program.
CPInew = (0.3*1.6+0.2*1.4+0.15*1+0.15*1.5)/0.8 = 1.41875

Hence, the overall speedup is 1.335/(0.8*1.41875*1.05) = 1.12 or 12%

4) [7 points] A third colleague notices that for the same clock cycle time increase (5%) you can double the associativity of the data cache used with the processor. The improved cache hit rate will reduce CPI for all loads and stores in the workload. JIPS Technologies has enough engineers to either implement the new instructions or improve the cache, but cannot do both. Calculate the minimum percentage improvement of the CPI for the loads and stores that is necessary in order to decide to go with the improved cache. Assume that the percentage of CPI improvement for loads and stores is exactly the same.

The improved cache must lead to overall performance improvement of at least 12% or a speedup of 1.12. We can use the speedup equation of part 3). In this case:
ICnew = ICold
CCTnew = 1.05*ICold
CPIold = 1.335

Speedup > 1.12 $\Leftrightarrow$ 1.335 > 1.12 *CPInew*1.05 $\Leftrightarrow$ CPInew < 1.135

Assume that the new cache design changes the CPI of loads and stores by a factor of x. I.e. the CPI for loads is 1.6*x and the CPI for stores is 1.4*x
CPInew < 1.135 $\Leftrightarrow$ 0.48*x + 0.28*x + 0.35 + 0.225 < 1.135 $\Leftrightarrow$ 0.76*x< 0.56 $\Leftrightarrow$ x < 0.73.

Hence, the new cache decrease the CPI of load and stores by 0.73 or 27%. The new CPI for loads must be 1.17 and the new CPI for stores must be 1.02.

[7 points] JIPS Technologies decides to implement the push and pop after all. You propose to use 16-bit encoding for the two instructions. This will lead to better code density which is important for embedded applications. All other instruction retain their 32-bit encoding. A senior colleague quickly points out that mixing 32-bit and 16-bit instructions requires an instruction set change that may limit the code density benefits from your proposal. What is the change and how does it hurt code density?

With 32-bit fixed-length instructions, all instructions addresses are aligned to 4-byte boundaries. In other words, the 2 least significant bits of the PC are always 0. Hence, the displacement of branch and jump instructions does not have to specify these last 2 bits.

If we interleave 16-bit and 32-bit instructions, an instruction address may be aligned to 4-byte or 2-byte boundaries. Only the very least bit of the PC is guaranteed to be 0. Compared to the original ISA, we need an extra bit of displacement with branches and jumps. However, we cannot go to 33-bit encoding for branches and jumps, hence with have to live with one less bit of displacement for these instructions.

If there is a branch in a program that uses all the bits of displacement with the original ISA (32-bit encoding only), this branch will require two instructions in the new ISA (16-bit & 32-bit instructions interleaved). If such branches are common, some of the code density benefits from encoding push and pop with 16-bits may be cancelled out.

Problem 3: Pipelining [34 Points]

A company has recently announced a processor running at 1.5GHz with the following pipeline:

IF1 First part of instruction fetch (TLB access)
IF2 Instruction fetch completes (instruction cache accessed)
RF Instruction decoded and register file read
EX Perform operation; compute memory address (base + displacement); compute branch target address; compute branch condition
MEM1 First part of memory access (TLB access)
MEM2 Memory access completes (data cache accessed)
WB Write back results into register file

As in the simple MIPS 5-stage pipeline, the results are written back into the register file in the first half of the cycle, and registers are read in the second half of the cycle.

3a. (8 points)
(i) How many register read ports does this pipeline require to implement a MIPS-style instruction set? how many write ports?
2 read ports, 1 write port

(ii) How many adders does the machine require (to prevent structural hazards), and in which stages are they used?
2 adders; one in IF1 (for the PC), one in EX (for the ALU).

(iii) What is the branch delay? What is the load delay?
Branch delay is 3 cycles. Load delay is 2 cycles (MEM2->EX).

(iv) To implement complete forwarding, how many destination register numbers must the machine record? how many comparators does it require?

3 destinations registers must be saved, and 6 comparators to compare 2 source register numbers with the 3 destinations

3b. (12 points)

Many processor designers favor a simple approach to branch prediction. One possibility considered for this pipeline was to use delayed branches. The compiler written for the processor is able to fill the branch delay slots with the following frequencies:

from before the branch, fill all slots:	for 10% of branches
from before the branch, fill two slots:	30%
from before the branch, fill one slot	10%
from the target path, fill one slot:	4%
from the sequential path, fill one slot:	3%

The compiler puts NOPs in unfilled delay slots. If the performance analysis group measures that 65% of branches in typical applications are taken, what is the average CPI of a branch?

```
Branches have three delay slots.
 # usefully filled = 10%*3 + 30%*2 + 10%*1 + 4%*(65% taken)*1
                   + 4%*(35% not taken)*1 = 1.0365
 ==> Avg. # of NOP cycles = 3 - 1.0365 = 1.9635

 ==> Avg. CPI of branches = 1 + penalty cycles = 1 + 1.9635 = 2.96
```

3c. (14 points)

Comparing pipelines: some people advocate staying with the simpler 5-stage MIPS pipeline. They say they can get the clock rate as high as 1.2GHz and still have only one branch delay slot while eliminating half of the load-use stall cycles.Given the following statistics for the 7-stage processor, which pipeline will perform better (ignoring memory stalls)?

	% of total execution cycles
Branch Delay NOP cycles	5%
Load Delay stall cycles	4%

You may need to use your results and the compiler statistics from part (b)(i). If you were not able to solve part (b)(i), assume that on average 1.5 of a branch's delay slots are filled with NOPs.

```
Since Exec.Time = IC * CPI / Clock Rate = # cycles / Clock rate

For 100 cycles of execution on a 7-stage pipeline, there are 5 branch
delay NOP cycles, 4 load delay cycles, and hence 91 instruction
execution cycles.

    Exec. Time (7) = 100 cycles / 1.5 GHz = 66.7 ns

The 7-stage pipeline requires 1.9635 NOP cycles per branch.

 The 5-stage pipeline needs only
    1 - (10% + 30% + 10% + 4%*65% + 3%*35%)*1 = 0.4635 NOP
cycles/branch (since if the compiler could fill 3 and 2 slots 10% and
30% of the time, respectively, it could certainly fill 1 slot for that
same fraction).
 ==> # branch delay stall cycles required is
     (0.4365/1.9635)*5 cycles = 1.11 cycles

 Also, eliminate half of load delay stall cycles, so only need 2:

 ==> Exec. Time (5 stage) = (91 + 2 + 1.11) / 1.2 GHz = 78.4 ns
which is much worse (5-stage is 78.4/66.7=17.5% slower)
```

Computer Science Comprehensive Examination
Computer Architecture
[100 points]

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

Number: _____

Problem	Max Score	Your Score
1	33	
2	33	
3	34	
TOTAL	100	

Problem 1: Memory Systems [33 Points 4 points each + 1 free]

A. Compared to a 16K direct-mapped cache, what type of misses will a 16K fully associative cache have fewer of? Circle all that apply.

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

(b) conflict misses

B. Suppose that cache A, a 16K-byte four-way set associative cache, cache B, a 16K-byte direct-mapped cache, and cache C, a 4K-byte direct-mapped cache are all referenced with an identical address sequence. All caches use a true least-recently used (LRU) replacement policy. Which of the following statements are true: (circle all that apply)

(a) A will contain a superset of the data in B
(b) A will contain a superset of the data in C
(c) B will contain a superset of the data in A
(d) B will contain a superset of the data in C

(b) A contains a superset of C. Cache C is equivalent to one "way" of A .
and (d) B contains a superset of C since both are direct mapped and B is larger.

C. In a 64K-byte four-way set-associative cache with 128-byte blocks, how large is the index field used to address the cache array? (write down the number of bits)

__7_____

D. If the cache of question 1.C 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)

___26 = 40-7 (index) – 7 (block)_____

E. A processors needs to support multiple processes running simultaneously with complete isolation (each process cannot access the memory of any other process). What is the minimum set of features in the architecture required to implement this complete isolation? (Circle minimal subset)

(a) Atomic branch and entry to privileged mode (in which all of memory can be addressed)
(b) A single base and length (segment) register
(c) Multiple base and length (segment) registers, one per process
(d) A translation look-aside buffer with a trap to a privileged mode handler routine on miss.

(a) and (b) suffice, as does (d) by itself (either gets full credit)

F. Replacing a single memory bank with multiple interleaved memory banks has which of the following effects? (circle all that apply)

(a) Increases latency
(b) Increases bandwidth
(c) Increases reliability
(d) Decreases latency

(b)

G. A bus-based cache coherence protocol "snoops" the directory of each processor's level-2 cache on each bus access. For this scheme to work, what relationship between the level-1 cache and level-2 cache must be maintained? (one word)

inclusion

H. A cache coherent multiprocessor has four processors, each with a 16Mbyte direct-mapped level-2 cache that is organized into 128-byte lines. Each processor makes single-word (8 byte) references (read and write in equal numbers) to widely-spaced single word locations in memory in a repeating pattern of 128K references. The patterns of the processors are disjoint – there is no sharing – and small enough to entirely fit in the cache. Rank the following three cache coherence protocols in order the amount of memory or bus traffic generated after the first iteration through the patterns?

(a) No cache – all data fetched and written is from pages marked uncached.
(c) Write-through – all cache lines have two states I and V.
Lines in state V are always in sync with memory – no dirty lines.
(d) Write-back – lines have three states, I, V, and D. The first write to a V line invalidates all other copies and make the line exclusive (and dirty), the D state.

The important thing to note here is that the working set is 128K 128Byte lines, so the working set is 16Mbytes which fits in the cache.

Thus (d) has the least traffic followed by (c) and then (a). If the working set were larger than the cache, (a) would have had the least traffic.

Problem 2: Instruction-Set Architecture [33 Points]

After graduate school, you work for JIPS Technologies, a company that designs embedded MIPS systems optimized for Java applications. The main product of JIPS is a MIPS processor and a just-in-time (JIT) compiler. The processor uses the classical 5-stage pipeline. The compiler runs along with any Java application and translates on-demand its Java code to MIPS instructions. The JIT compiler is kept very simple in order to minimize its overhead. It does implement the stack storage defined by the Java architecture, but tries to eliminate the number of push and pop operations to the stack by allocating as many temporary results as possible to the registers available in the MIPS processor.

After analyzing the typical workload (application & JIT), you come up with the following frequencies and CPIs for each type of instructions supported by the processor:

Type	Frequency	CPI
Load	30%	1.6
Store	20%	1.4
Arithmetic operations	35%	1.0
Branches & jumps	15%	1.5

Your analysis also indicates that 50% of the stores are used to push values into the stack. 33.33% of the loads are used to pop values from the stack. Assume that all pushes and pops operate on 32b integer numbers.

1) [7 points] You decide to improve performance by adding push and pop instructions to the MIPS ISA:

```
push rs   # MEM[r30+4]←rs; r30←r30+4
pop  rs   # rs←MEM[r30];   r30←r30-4
```

Register r30 stores a pointer to the top of the stack by default. What additional resources will you need in the 5-stage pipeline to support the execution the two new instructions? Explain.

Pop writes both rs and r30. Hence, we need a 2nd write port to the register file.

No adder is needed for (r30+4) or (r30-4). Push and pop can perform this operation on the integer ALU in the EX stage of the pipeline.

2) [6 points] A colleague notices that **push** and **pop** are similar to loads and stores with autoincrement and autodecrement addressing respectively. Hence, she suggests introducing autoincrement/autodecrement addressing (in which the register used to provide the address is incremented/decremented as a side effect of the load or store instruction) to the MIPS ISA instead of **push** and **pop**. Explain to her why **push** and **pop** are cheaper to implement.

General autoincrement/autodecrement loads/stores have two disadvantages

- They require additional bits in the opcode to encode the increment/decrement value (register specifier or immediate)
- They require 2 integer ALUs: one for the effective address calculation (register + offset) and one for the autoincrement/autodecrement.

3) [6 points] The JIPS design group comes to the conclusion that the introduction **push** and **pop** will increase the clock cycle time of the processor by 5% due to the additional resourced needed to implement them. Calculate the overall performance improvement if the new instructions are added to the design.

$$Speedup = \frac{ExecutionTimeOld}{ExecutionTimeNew} = \frac{ICold * CPIold * CCTold}{ICnew * CPInew * CCTnew}$$

CPIold = Sum(individual CPIs) = 0.3*1.6+0.2*1.4+0.35*1+0.15*1.5 = 1.335
CCTnew = 1.05*CCTold

Each push/pop instruction will replace one store/load instruction in the original program and the arithmetic operation used to increment/decrement the stack pointer. Hence, the introduction of push/pop eliminates a number of arithmetic operations from the original program. Therefore:
ICnew = (1-#push-#pop)*ICold = (1-0.5*0.2*0.33*.3)*ICold = 0.8*ICold

The CPI of a push/pop instruction is equal to that of a store/load instruction in the original program. However, our CPInew calculation must take into account that we eliminated some of the arithmetic instructions in the original program.
CPInew = (0.3*1.6+0.2*1.4+0.15*1+0.15*1.5)/0.8 = 1.41875

Hence, the overall speedup is 1.335/(0.8*1.41875*1.05) = 1.12 or 12%

4) [7 points] A third colleague notices that for the same clock cycle time increase (5%) you can double the associativity of the data cache used with the processor. The improved cache hit rate will reduce CPI for all loads and stores in the workload. JIPS Technologies has enough engineers to either implement the new instructions or improve the cache, but cannot do both. Calculate the minimum percentage improvement of the CPI for the loads and stores that is necessary in order to decide to go with the improved cache. Assume that the percentage of CPI improvement for loads and stores is exactly the same.

The improved cache must lead to overall performance improvement of at least 12% or a speedup of 1.12. We can use the speedup equation of part 3). In this case:
ICnew = ICold
CCTnew = 1.05*ICold
CPIold = 1.335

Speedup > 1.12 $\Leftrightarrow$ 1.335 > 1.12 *CPInew*1.05 $\Leftrightarrow$ CPInew < 1.135

Assume that the new cache design changes the CPI of loads and stores by a factor of x. I.e. the CPI for loads is 1.6*x and the CPI for stores is 1.4*x
CPInew < 1.135 $\Leftrightarrow$ 0.48*x + 0.28*x + 0.35 + 0.225 < 1.135 $\Leftrightarrow$ 0.76*x< 0.56 $\Leftrightarrow$ x < 0.73.

Hence, the new cache decrease the CPI of load and stores by 0.73 or 27%. The new CPI for loads must be 1.17 and the new CPI for stores must be 1.02.

[7 points] JIPS Technologies decides to implement the push and pop after all. You propose to use 16-bit encoding for the two instructions. This will lead to better code density which is important for embedded applications. All other instruction retain their 32-bit encoding. A senior colleague quickly points out that mixing 32-bit and 16-bit instructions requires an instruction set change that may limit the code density benefits from your proposal. What is the change and how does it hurt code density?

With 32-bit fixed-length instructions, all instructions addresses are aligned to 4-byte boundaries. In other words, the 2 least significant bits of the PC are always 0. Hence, the displacement of branch and jump instructions does not have to specify these last 2 bits.

If we interleave 16-bit and 32-bit instructions, an instruction address may be aligned to 4-byte or 2-byte boundaries. Only the very least bit of the PC is guaranteed to be 0. Compared to the original ISA, we need an extra bit of displacement with branches and jumps. However, we cannot go to 33-bit encoding for branches and jumps, hence with have to live with one less bit of displacement for these instructions.

If there is a branch in a program that uses all the bits of displacement with the original ISA (32-bit encoding only), this branch will require two instructions in the new ISA (16-bit & 32-bit instructions interleaved). If such branches are common, some of the code density benefits from encoding push and pop with 16-bits may be cancelled out.

Problem 3: Pipelining [34 Points]

A company has recently announced a processor running at 1.5GHz with the following pipeline:

IF1 First part of instruction fetch (TLB access)
IF2 Instruction fetch completes (instruction cache accessed)
RF Instruction decoded and register file read
EX Perform operation; compute memory address (base + displacement); compute branch target address; compute branch condition
MEM1 First part of memory access (TLB access)
MEM2 Memory access completes (data cache accessed)
WB Write back results into register file

As in the simple MIPS 5-stage pipeline, the results are written back into the register file in the first half of the cycle, and registers are read in the second half of the cycle.

3a. (8 points)
(i) How many register read ports does this pipeline require to implement a MIPS-style instruction set? how many write ports?

```
2 read ports, 1 write port
```

(ii) How many adders does the machine require (to prevent structural hazards), and in which stages are they used?

```
2 adders; one in IF1 (for the PC), one in EX (for the ALU).
```

(iii) What is the branch delay? What is the load delay?

```
Branch delay is 3 cycles. Load delay is 2 cycles (MEM2->EX).
```

(iv) To implement complete forwarding, how many destination register numbers must the machine record? how many comparators does it require?

```
3 destinations registers must be saved, and 6 comparators to compare
2 source register numbers with the 3 destinations
```

3b. (12 points)

Many processor designers favor a simple approach to branch prediction. One possibility considered for this pipeline was to use delayed branches. The compiler written for the processor is able to fill the branch delay slots with the following frequencies:

from before the branch, fill all slots:	for 10% of branches
from before the branch, fill two slots: 30%	
from before the branch, fill one slot	10%
from the target path, fill one slot: 4%	
from the sequential path, fill one slot: 3%	

The compiler puts NOPs in unfilled delay slots. If the performance analysis group measures that 65% of branches in typical applications are taken, what is the average CPI of a branch?

```
Branches have three delay slots.
 # usefully filled = 10%*3 + 30%*2 + 10%*1 + 4%*(65% taken)*1
                   + 4%*(35% not taken)*1 = 1.0365
 ==> Avg. # of NOP cycles = 3 - 1.0365 = 1.9635

 ==> Avg. CPI of branches = 1 + penalty cycles = 1 + 1.9635 = 2.96
```

3c. (14 points)

Comparing pipelines: some people advocate staying with the simpler 5-stage MIPS pipeline. They say they can get the clock rate as high as 1.2GHz and still have only one branch delay slot while eliminating half of the load-use stall cycles.Given the following statistics for the 7-stage processor, which pipeline will perform better (ignoring memory stalls)?

	% of total execution cycles
Branch Delay NOP cycles	5%
Load Delay stall cycles	4%

You may need to use your results and the compiler statistics from part (b)(i). If you were not able to solve part (b)(i), assume that on average 1.5 of a branch's delay slots are filled with NOPs.

```
Since Exec.Time = IC * CPI / Clock Rate = # cycles / Clock rate

For 100 cycles of execution on a 7-stage pipeline, there are 5 branch
delay NOP cycles, 4 load delay cycles, and hence 91 instruction
execution cycles.

    Exec. Time (7) = 100 cycles / 1.5 GHz = 66.7 ns

The 7-stage pipeline requires 1.9635 NOP cycles per branch.

 The 5-stage pipeline needs only
    1 - (10% + 30% + 10% + 4%*65% + 3%*35%)*1 = 0.4635 NOP
cycles/branch (since if the compiler could fill 3 and 2 slots 10% and
30% of the time, respectively, it could certainly fill 1 slot for that
same fraction).
==> # branch delay stall cycles required is
     (0.4365/1.9635)*5 cycles = 1.11 cycles

 Also, eliminate half of load delay stall cycles, so only need 2:

 ==> Exec. Time (5 stage) = (91 + 2 + 1.11) / 1.2 GHz = 78.4 ns
which is much worse (5-stage is 78.4/66.7=17.5% slower)
```

Stanford University Computer Science Department
2003 Comprehensive Exam in Databases
SAMPLE SOLUTION

1. **(14 pts.)** Relation $R(A, B, C, D, E)$ has functional dependencies $AB \rightarrow C$, $BC \rightarrow D$, $CD \rightarrow E$, and $DE \rightarrow A$.

   (a) **(6 pts.)** What are all the keys of R?

   Answer: AB, BC, and BDE. Note that B must be in any key, since it doesn't appear on the right of any FD. That fact makes the search for keys fairly easy.

   (b) **(2 pts.)** Which of the given functional dependencies are Boyce-Codd Normal Form (BCNF) violations? Which are Third Normal Form (3NF) violations?

   Answer: $CD \rightarrow E$ and $DE \rightarrow A$ violate BCNF. None violate 3NF, because all attributes are prime.

   (c) **(6 pts.)** Find a lossless-join decomposition (a decomposition from which the original relation R can be recovered) of R into BCNF relations. Show your steps and your reasoning—which violating functional dependencies cause which decompositions in which order.

   Answer: Suppose we use $CD \rightarrow E$ to decompose. Since $CD^+ = ACDE$, one of the schemes is $R1(A, C, D, E)$ and the other is $R2(B, C, D)$. The latter is in BCNF, since BC is the only key, and $BC \rightarrow D$ the only projected FD. However, $R1$ is not in BCNF. For example, $DE \rightarrow A$ is a projected FD, but $DE^+ = ADE$, so DE is not a superkey for $R1$. Thus, we decompose $R1$ into $R3(A, D, E)$ and $R4(C, D, E)$. The constituents of the decomposition are $R2$, $R3$, and $R4$.

2. **(16 pts.)** Consider the following three entity-relationship (E/R) diagrams that represent students, courses, and the quarters (e.g., Autumn, 2003) that the student took the course.

(a)

(b)

(c)

(a) **(6 pts.)** Explain the differences among the three diagrams (a), (b), and (c). What constraints does each imply? Based on constraints, do any two of them capture exactly the same real-world data? Explain the differences in terms of students, courses, and quarters (e.g., "a student can take only one course in a given quarter").

Answer: (a) is an unconstrained relationship among students, courses, and quarters. In particular, students can take a course several times. (b) adds the constraint that a student can take a given course in only one quarter; i.e., no repeating courses is allowed. (c) is the same as (b). In fact, (c) is really a shorthand for the diagram (b). Many people did not realize that the "no repeating courses" constraint is implied by (c), but you can reason as follows. Since the meaning of *Takes* is a relationship set, a student-course pair can appear in this set only once. Whatever season and year are attributes of this pair are the one and only one quarter in which that student took that course.

(b) **(2 pts.)** If in E/R diagram (b) we added an arrow on the line from *Takes* to *Students*, what additional constraint would be implied by that diagram?

Answer: Only one student can take any given course in any given quarter.

(c) **(8 pts.)** Convert the structure of E/R diagram (b), *without the addition mentioned in part (b) of this problem*, to an ODL (Object Definition Language) schema. You

need not specify extents, but you should specify keys when appropriate. Try to keep it simple, not introducing classes that you don't really need. You may use the following abbreviations: `att` for "attribute," `rel` for "relationship," `inv` for "inverse."

Answer: In general, we need a connecting class to represent student-course-quarter triples. However, since quarters are really just data, we are better off making its data part of the connecting class, which we'll call *Takes*.

```
class Student (key name) {
    attribute string name;
    attribute string addr;
    relationship Set<Takes> courses
        inverse Takes::theStudent;
}

class Course (key number) {
    attribute integer number;
    attribute string title;
    relationship Set<Takes> students
        inverse Takes::theCourse;
}

class Takes (key (theStudent, theCourse)) {
    attribute string season;
    attribute integer year;
    relationship Student theStudent
        inverse Student::courses;
    relationship Course theCourse
        inverse Course::students;
}
```

3. **(14 pts.)** Consider a relation $R(A, B)$ that contains $r > 0$ tuples, and a relation $S(B, C)$ that contains $s > 0$ tuples. You may assume that neither R nor S contains duplicate tuples.

   (a) **(10 pts.)** For each of the expressions in (set-based) relational algebra in the following table, state the minimum and maximum number of tuples that could possibly be in the result of the expression. (The actual number will depend on the actual data.) Your statements should be based on r and s to the extent possible.

   Answer:

Expression	Minimum #tuples	Maximum #tuples
$\sigma_P(R)$ P a predicate	0	r
$\Pi_A(R)$ A an attribute list	1	r
$R \bowtie S$	0	$r \cdot s$
$R \cup \rho_{S(A,B)}(S)$	$\max(r, s)$	$r + s$
$\Pi_B(R) - (\Pi_B(R) - \Pi_B(S))$	0	$\min(r, s)$
$R \overset{\circ}{\bowtie}_L S$ (left outerjoin)	r	$r \cdot s$
$R \overset{\circ}{\bowtie} S$ (full outerjoin)	$\max(r, s)$	$r \cdot s$

(b) **(4 pts.)** Continue to assume neither R nor S contains duplicate tuples, but now use bag (multiset) instead of set-based relational algebra. Do any of your entries in the table above change? If so, state which ones and give the new entries.

Answer: The minimum for $\Pi_A(R)$ changes to r. The minimum for $R \cup \rho_{S(A,B)}(S)$ changes to $r + s$.

4. **(10 pts.)** Consider a table `CompScore(name, score)` containing scores on the database comp. For simplicity assume that both names and scores are unique in the table, and further assume the table has an odd number of rows. Use SQL to find the name of the student with the median score. Assume you do *not* have a built-in `median` operation.

Answer:

```
Select name
From   CompScore C1
Where  (Select Count(*) From CompScore C2
        Where C2.score < C1.score) =
       (Select Count(*) From CompScore C2
        Where C2.score > C1.score)
```

5. **(6 pts.)** Consider a database with three tables containing one tuple each:

   $R(A, B)$ contains tuple $\langle 1, 2 \rangle$
   $S(C, D)$ contains tuple $\langle 2, 3 \rangle$
   $T(E, F)$ contains tuple $\langle 2, 4 \rangle$

   Specify a minimal set of SQL referential-integrity constraints over the three-table schema such that when the single tuple in relation R is deleted, all three tables are made empty automatically.

   Answer:

```
S.C References R.B On Delete Cascade
T.E References R.B On Delete Cascade
```

Computer Graphics Comprehensive Exam

Computer Science Department
Stanford University
Fall 2003

NAME:

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

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

You have 60 minutes to complete the exam.

1. [20 points] Computer graphics definitions.

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

1A [5 points] Alpha channel.

1B [5 points] Phong reflection model.

1C [5 points] Spline.

1D [5 points] Gouraud shading.

2. [20 points] Transformations.

Computer graphics relies heavily on transformations. Most common are linear transformations such as rotations and translations. 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(360) = Rx(720) T(1,0,0)?

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

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

Rx(45) Rx(30) = Rx(15) Rx(60)?

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

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)?

$[T(1,0,0)T(0,2,0)]^{-1}$ = T(-1,0,0)T(0,-2,0)?

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

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

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 description of a surface; the output is the point of intersection.

Assume a 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.

The simplest surface is a plane. The following equation defines a plane:

$$Ax + By + Cz + D = 0.$$

A plane also defines a half-space: $Ax + By + Cz + d < 0$.

A set of planes may be used to define convex polyhedra. Each face i of the convex polyhedra is associated with a plane (Ai, Bi, Ci, Di). A convex polyhedra is defined to be the intersection of the half-spaces created by the planes making up its faces.

Work out a procedure for computing the point of intersection of a ray with a convex polyhedra defined as the intersection of n half-spaces. In general, a ray intersects a convex shape in two points. Your procedure should return the closest point of intersection in the direction of the ray.

4. [20 points] Hidden-surface elimination.

Hidden-surface elimination is one the classic algorithms in computer graphics. The goal of hidden-surface elimination is to draw a picture where each point in the image shows the surface visible at that point; hidden points and surfaces are not shown.

4A [5 points] Describe succinctly the z-buffer algorithm for hidden surface elimination.

4B [5 points] What is the complexity of the z-buffer algorithm?

4C [5 points] Is the z-buffer algorithm optimal? What would be the complexity of an optimal algorithm?

4D [5 points] Suppose you are given a scene consisting of a set of polygons. You can draw polygons as lines (i.e. as an outlined polygon) or as a filled polygon. How would you create a line drawing of the scene with hidden lines removed? Hint: Consider enhancing the basic z-buffer drawing mode slightly.

5 [20 points] Intensity and Gamma.

Cathode-Ray Tubes (CRTs), or monitors, convert voltage to light. However, the relationship between the amount of light energy produced and the input voltage is nonlinear:

$$L = V^{\gamma}$$

In this equation, L is the amount of light output and V is the input voltage. Gamma γ characterizes this non-linear relationship. A typical value for γ is 2.2. This means, that if the input voltage is doubled, the output light increases by more than a factor of four.

The framebuffer stores pixel values. These pixel values can be converted to voltages in various ways.

5A [5 points] Suppose the voltage is set to be proportional to the framebuffer values. Call this V-space. What are some advantages of working in this V-space?

5B [5 points] Suppose that it were possible to store values in the framebuffer in such a way that the amount of light energy produced were directly proportional to the value stored in the framebuffer. Call this L-space. What are some advantages of working in L-space?

5C [5 points] What hardware would be required to convert framebuffer values to voltages if the framebuffer values are stored V-space?

5D [5 points] What additional hardware would be required to convert framebuffer values to voltages if the values are stored L-space?

Stanford University
Computer Science Department

Fall 2003 Comprehensive Exam in Logic

1. **Open Book & Notes / No Laptops. Write only in the space provided on the question paper.**
2. **The exam is timed for 60 minutes.**
3. **Write your Magic Number on this sheet.**

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*----------**

Comprehensive Examination in Logic

Stanford University

November 2003

Instruction

- You have **60 minutes** to complete the exam.
- The exam consists of **20 questions** for a total of **60 points**. You get **3 points** for each correct answer and **-1 point** for each incorrect one.
- The exam is **open book**, but no laptops or electronics accessories are allowed.

Do not turn this page until instructed to do so.

Proposition Logic (9 points)

Question 1. The sentence $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$ is:

(A) Valid and satisfiable.

(B) Invalid but satisfiable.

(C) Valid but unsatisfiable.

(D) Invalid and unsatisfiable.

Question 2. Let P stand for *"I will serve you tea"* and let Q stand for *"you order coffee"*. The sentence *"I will serve you tea if you do not order coffee"* is best represented by which of the following?

(A) $\neg P \to Q$.

(B) $P \to \neg Q$.

(C) $P \to Q$.

(D) $\neg P \to \neg Q$.

Question 3. Which of the following forms a complete set of propositional connectives?

I. $\wedge, \neg$

II. $\wedge, \vee$

III. $\to$, *false*

(A) I only.

(B) II only.

(C) I and III only.

(D) I, II, and III.

Predicate Logic (9 points)

Question 4. Consider the sentences:

I. $(\exists y)(\forall x)P(x,y) \rightarrow (\forall x)(\exists y)P(x,y)$

II. $(\forall x)(\exists y)P(x,y) \rightarrow (\exists y)(\forall x)P(x,y)$

III. $(\forall x)Q(x) \rightarrow (\exists x)Q(x)$

Which of them is valid?

(A) None.

(B) I and III only.

(C) II and III only.

(D) III only.

Question 5. Let $\mathcal{F}$ and $\mathcal{G}$ be arbitrary formulae, and define S_1 and S_2 as follows:

$$S_1: \quad (\forall x)\mathcal{F} \rightarrow (\forall x)\mathcal{G}$$
$$S_2: \quad (\forall x)(\mathcal{F} \rightarrow \mathcal{G})$$

Which of the following are true?

I. S_1 and S_2 are equivalent.

II. Assuming that S_1 and S_2 are closed, if S_1 is valid then S_2 is valid.

III. Assuming that S_1 and S_2 are closed, if S_2 is valid then S_1 is valid.

(A) I only.

(B) II only.

(C) III only.

(D) I and III only.

Question 6. Let $\mathcal{F}$ and $\mathcal{G}$ be arbitrary formulae. If $(\exists x)(\mathcal{F} \wedge \neg\mathcal{G})$ is unsatisfiable then:

(A) $(\exists x)(\neg\mathcal{F} \rightarrow \mathcal{G})$ is valid.

(B) $(\forall x)(\mathcal{F} \wedge \mathcal{G})$ is valid.

(C) $(\forall x)(\mathcal{F} \rightarrow \mathcal{G})$ is valid.

(D) $(\forall x)(\mathcal{G} \rightarrow \mathcal{F})$ is valid.

Unification (6 points)

Question 7. Which of the following is a most general unifier of $h(x, y)$ and $h(g(y), f(x))$?

(A) $\{x \leftarrow g(y)\}$

(B) $\{y \leftarrow f(x)\}$

(C) $\{x \leftarrow g(y), y \leftarrow f(x)\}$

(D) the terms are not unifiable.

Question 8. Which of the following are unifiable?

$$t_1 : h(f(x), x)$$
$$t_2 : h(y, z)$$
$$t_3 : h(z, g(y))$$

(A) (t_1, t_2) only.

(B) (t_1, t_2) and (t_1, t_3) only.

(C) (t_1, t_2) and (t_2, t_3) only.

(D) (t_2, t_3) and (t_1, t_3) only.

Skolemization (6 points)

Question 9. Consider the sentence

$$(\forall x)(\exists y)[(\exists z)(\forall w)P(x,y,z,w) \rightarrow (\exists w)Q(w)]\,.$$

Which of the following skolemizations preserves *validity*?

(A) $(\forall x)[(\exists z)P(x, f(x), z, g(x,z)) \rightarrow Q(h(x))]$

(B) $(\exists y)[(\forall w)P(a, y, f(y), w) \rightarrow (\exists w)Q(w)]$

(C) $(\forall x)[(\exists z)P(x, f(x), z, g(x,z)) \rightarrow Q(a)]$

(D) $(\forall x)[(\forall w)P(x, f(x), g(x,w), w) \rightarrow Q(h(x))]$

Question 10. Consider the sentence

$$(\forall x)(\exists y)[(\exists z)(\forall w)P(x,y,z,w) \rightarrow (\exists w)Q(w)]\,.$$

Which of the following skolemizations preserves *satisfiability*?

(A) $(\forall x)[(\exists z)P(x, f(x), z, g(x,z)) \rightarrow Q(h(x))]$

(B) $(\exists y)[(\forall w)P(a, y, f(y), w) \rightarrow (\exists w)Q(w)]$

(C) $(\forall x)[(\exists z)P(x, f(x), z, g(x,z)) \rightarrow Q(a)]$

(D) $(\forall x)[(\forall w)P(x, f(x), g(x,w), w) \rightarrow Q(h(x))]$

Deductive tableaux (6 points)

Question 11. Consider the following deductive tableau where x, y, and z are variables, and a is a constant.

	assertions	goals
A1	$P(x,x) \vee Q(y)$	
G2		$R(y,z) \wedge P(a,z)$

What is the result of an AG-resolution of A1 and G2?

(A)

G3		$\neg Q(y) \wedge R(y,z)$

(B)

G3		$\neg Q(y) \vee R(w,a)$

(C)

G3		$Q(y) \wedge R(w,z)$

(D)

G3		$\neg Q(y) \wedge R(w,a)$

Question 12. Consider the connective $\downarrow$ (*nor*) with the following semantics: the truth value of $\mathcal{F} \downarrow \mathcal{G}$ is true if both $\mathcal{F}$ and $\mathcal{G}$ are false, and false otherwise. Which of the following nor-splitting deduction rules preserves equivalence of a deductive tableau?

I.

assertions	goals
$A_1 \downarrow A_2$	
	A_1
	A_2

II.

assertions	goals
$A_1 \downarrow A_2$	
A_1	
A_2	

III.

assertions	goals
	$A_1 \downarrow A_2$
A_1	
A_2	

(A) None.

(B) I only.

(C) II only.

(D) I and III only.

Polarity (6 points)

Question 13. Let X be a formula whose occurrences in $\mathcal{F}[X]$ have all negative polarity. Assume that the formula $X \to Y$ is true under interpretation I. Then:

(A) $\mathcal{F}[X]$ is true under I.

(B) $\mathcal{F}[Y]$ is true under I.

(C) $\mathcal{F}[X] \to \mathcal{F}[Y]$ is true under I.

(D) $\mathcal{F}[Y] \to \mathcal{F}[X]$ is true under I.

Question 14. Which of the following sentences are instances of the polarity proposition?

I. $(P \to Q) \to ((R \to P) \to (R \to Q))$

II. $(Q \to P) \to ((R \to P) \to (R \to Q))$

III. $(P \to Q) \to (Q \to P)$

(A) None.

(B) I only.

(C) II only.

(D) III only.

First-order theories (9 points)

Question 15. Let T be a first-order theory such that

- all finite interpretations with an even number of members are models of T;
- all finite interpretations with an odd number of members are not models of T.

Which of the following hold?

I. T does not exists

II. T has an enumerable model

III. There exists a theory S such that $T \cup S \models (\forall x, y)(x = y)$

(A) None.

(B) I only.

(C) II only.

(D) II and III only.

Question 16. Let T be a finitely axiomatizable theory. Which of the following necessarily holds?

I. T has a finite model

II. T is decidable

III. If T is complete, then T is decidable

(A) None.

(B) I and II only.

(C) I and III only.

(D) III only.

Question 17. Which of the following theories is undecidable?

(A) The theory of natural numbers under zero and successor.

(B) The theory of natural numbers under zero, successor, and addition.

(C) The theory of natural numbers under zero, successor, addition, and multiplication.

(D) None of the above.

Well-founded relations (9 points)

Question 18. Let $<$ be a well-founded relation over A. Which of the following necessarily holds?

(A) $<$ is irreflexive.

(B) $<$ is transitive.

(C) A is finite.

(D) A is infinite.

Question 19. Which of the following is true?

I. There exists a well-founded relation which is symmetric

II. There exists a well-founded relation which is transitive

(A) None.

(B) I only.

(C) II only.

(D) I and II.

Question 20. Which of the following relations $\prec$ over the set of natural numbers is well-founded?

I. $x \prec y$ iff $x = y + 1$

II. $x \prec y$ iff $x \leq y$

III. $x \prec y$ iff there exists a z such that $yz = x$.

(A) I only.

(B) I and II only.

(C) I and III only.

(D) None of the above.

Stanford University
Computer Science Department

Fall 2003 Comprehensive Exam in Networks

1. **Closed Book/ No Laptops & Notes Write only in the Blue Book.**
2. **The exam is timed for 60 minutes.**
3. **Write your Magic Number on the Blue Book.**

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

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

1. *that they will not give or receive aid in examinations; that they will not give or receive 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----------

Comprehensive Exam: Networks (60 points) **Autumn 2003**

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

   (a) (8 points) Define the so-called "end-to-end argument" in networking, and describe the key way this principle has affected the design of the Internet.

   (b) (7 points) Describe briefly how an Internet ebanking would be structured to be truly "end-to-end", focusing on the basic debit-credit transactions.

2. (15 points total) *TCP transport protocol* TCP was originally described and implemented as a go-back-N protocol with respect to its error control.

   (a) (8 points) Describe what this means, illustrating with a time-based diagram of packet exchanges including a single data packet loss. (You can assume the connection is already set up.)

   (b) (7 Points) Provide a formula for determining the TCP maximum throughput for a given packet drop rate D, introducing whatever additional parameters/variables you need, e.g. max link data rate R, MTU M, etc. and assuming that only data packets are dropped, there is at most one drop occurring per go-back-N event, and the transmission rate in the absence of drop is only limited by the link rate.

3. (15 points total) *Virtual Circuits vs. Datagrams* The UN has decided to jump in to mediate the never-ending dispute between the "virtual circuit" (VC) nerds and the "datagram" nerds of networking. You are called upon to provide quantifiable evaluation of the two approaches.

   (a) (5 points) Regarding bandwidth consumption, describe how you would quantify the bandwidth cost differential between VCs and datagrams, defining any parameters (e.g. difference in header size D) that you need, and assuming separate network layer packets are used to setup and teardown VCs and there is some reasonable limit on number of VCs per host.

   (b) (5 Points) Considering memory consumption in the routers and the endpoints, describe how you would quantify this cost between VCs and datagrams. (You can give formulae or just provide a convincing description that you know the key parameters and how they affect.)

   (c) (5 points) What cost factors would you consider most significant between VCs and datagrams as the Internet scales (you can consider others, such as processor cycles), why, which of VCs vs datagrams would you favor, and why?

4. (15 points total) *Ethernet*

   (a) (6 points) Describe the key extension that the Ethernet media access control (MAC) protocol, CSMA-CD, makes over Aloha network protocol and explain what effect this has on its performance characteristics.

   (b) (5 points) The ATM boys used to claim that Ethernet could not be scaled to higher speeds than 10 MBPS yet now we see 10 GBPS Ethernet emerging (after 100MBPS and 1 GBPS). What the basis of their argument (or any argument) that you could not scale Ethernet to higher speeds and how has industry gotten around this?

please turn over

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

The End

Computer Science Department
Stanford University
Comprehensive Examination in Numerical Analysis
Fall 2003

1. **Vector and Matrix Norms [8 pts]**

The following definitions hold for the norm and condition number of a *rectangular* $m \times n$ matrix A with respect to a specific matrix norm

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \qquad cond(A) = \left(\max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \right) \cdot \left(\min_{x \neq 0} \frac{\|Ax\|}{\|x\|} \right)^{-1}$$

Given the singular value decomposition $A = U \Sigma V^T$ of the matrix A (where U and V are orthogonal and Σ is the $m \times n$ diagonal matrix containing the singular values of A)

i. [3 pts] Prove that $\|A\|_2 = \|\Sigma\|_2$ using the definition above

ii. [5 pts] Prove that

$$\|A\|_2 = \sigma_{max} \qquad \text{and} \qquad cond_2(A) = \sigma_{max} / \sigma_{min}$$

where σ_{max} is the largest singular value of A and σ_{min} the smallest one.

Solutions

i. For any $x \in R^n \setminus \{\vec{0}\}$ we have

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} = \frac{(Ax)^T Ax}{x^T x} = \frac{x^T A^T A x}{x^T x} = \frac{x^T \left(U \Sigma V^T\right)^T U \Sigma V^T x}{x^T x} = \frac{x^T V \Sigma^T U^T U \Sigma V^T x}{x^T x} = \frac{x^T V \Sigma^T \Sigma V^T x}{x^T V V^T x} =$$

$$= \frac{\left(\Sigma V^T x\right)^T \Sigma V^T x}{\left(V^T x\right)^T V^T x} = \frac{\|\Sigma V^T x\|_2^2}{\|V^T x\|_2^2} \Rightarrow \frac{\|Ax\|_2}{\|x\|_2} = \frac{\|\Sigma V^T x\|_2}{\|V^T x\|_2}$$

Since the mapping $x \mapsto V^T x$ is an isomorphism on $R^n \setminus \{\vec{0}\}$ (its inverse is simply $x \mapsto Vx$) we have

$$\max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max_{x \neq 0} \frac{\|\Sigma V^T x\|_2}{\|V^T x\|_2} = \max_{V^T x \neq 0} \frac{\|\Sigma (V^T x)\|_2}{\|V^T x\|_2} = \max_{y \neq 0} \frac{\|\Sigma y\|_2}{\|y\|_2}$$

thus $\|A\|_2 = \|\Sigma\|_2$

ii. Let σ_i, $i=1,\ldots,n$ be the singular values of A, forming the diagonal of the matrix Σ. Let $\sigma_k=\sigma_{\max}$ be the largest and $\sigma_l=\sigma_{\min}$ the smallest among them. Then

$$\left.\begin{aligned}\max_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2}=\max_{x\neq 0}\frac{\|\Sigma x\|_2}{\|x\|_2}=\max_{x\neq 0}\sqrt{\frac{\sum_i \sigma_i^2 x_i^2}{\sum_i x_i^2}}\le \max_{x\neq 0}\sqrt{\frac{\sum_i \sigma_{\max}^2 x_i^2}{\sum_i x_i^2}}=\sigma_{\max}\\ \max_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2}=\max_{x\neq 0}\frac{\|\Sigma x\|_2}{\|x\|_2}\ge\frac{\|\Sigma e_k\|_2}{\|e_k\|_2}=\frac{\|\sigma_k e_k\|_2}{1}=\sigma_k=\sigma_{\max}\end{aligned}\right\}\Rightarrow \max_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2}=\sigma_{\max}$$

and

$$\left.\begin{aligned}\min_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2}=\min_{x\neq 0}\frac{\|\Sigma x\|_2}{\|x\|_2}=\min_{x\neq 0}\sqrt{\frac{\sum_i \sigma_i^2 x_i^2}{\sum_i x_i^2}}\ge \min_{x\neq 0}\sqrt{\frac{\sum_i \sigma_{\min}^2 x_i^2}{\sum_i x_i^2}}=\sigma_{\min}\\ \min_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2}=\min_{x\neq 0}\frac{\|\Sigma x\|_2}{\|x\|_2}\le\frac{\|\Sigma e_l\|_2}{\|e_l\|_2}=\frac{\|\sigma_l e_l\|_2}{1}=\sigma_l=\sigma_{\min}\end{aligned}\right\}\Rightarrow \min_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2}=\sigma_{\min}$$

Therefore, using the definition we have $\|A\|_2=\sigma_{\max}$ and $cond_2(A)=\sigma_{\max}/\sigma_{\min}$

2. **Differential Equations [10 pts]**

i. [4 pts] Given a square matrix A whose eigenvalues have negative real parts, show that the matrix $I-A$ is invertible and the eigenvalues of $B=(I-A)^{-1}(I+A)$ are given by the formula $\lambda_i^B=\frac{1+\lambda_i^A}{1-\lambda_i^A}$, where λ_i^A are the eigenvalues of A

ii. [6 pts] Consider the vector ordinary differential equation $\vec{y}'=f(x,\vec{y})$ and the implicit trapezoidal method for solving it:

$$\vec{y}_{k+1}=\vec{y}_k+h\frac{f(x_k,\vec{y}_k)+f(x_{k+1},\vec{y}_{k+1})}{2}$$

Prove that this method is unconditionally stable when applied to the model vector ODE $\vec{y}'=A\vec{y}$ for a matrix A whose eigenvalues have negative real parts (that is, show that $\|\vec{y}_k\|\to 0$ as $k\to\infty$, regardless of the value of the step size h)

Solutions

i. The matrix $I-A$ is singular if and only if $\det(I-A)=0$. But this implies that $\lambda=1$ is an eigenvalue of A, which is not the case since all the eigenvalues of A have negative real parts. Furthermore, we have

$$\det\left[(I-A)^{-1}(I+A)-\lambda I\right]=0 \Leftrightarrow \det\left[(I-A)^{-1}(I+A)-\lambda(I-A)^{-1}(I-A)\right]=0 \Leftrightarrow$$
$$\Leftrightarrow \det\left[(I-A)^{-1}\right]\det\left[(I+A)-\lambda(I-A)\right]=0 \Leftrightarrow \det\left[(I+A)-\lambda(I-A)\right]=0 \Leftrightarrow$$
$$\Leftrightarrow \det\left[(\lambda+1)A-(\lambda-1)I\right]=0 \Leftrightarrow \det\left[A-\frac{\lambda-1}{\lambda+1}I\right]=0$$

Therefore, the eigenvalues of A and $B=(I-A)^{-1}(I+A)$ are associated as follows

$$\lambda_i^A=\frac{\lambda_i^B-1}{\lambda_i^B+1} \Leftrightarrow \lambda_i^B=\frac{1+\lambda_i^A}{1-\lambda_i^A}$$

ii. Application of the trapezoidal rule in the model equation yields

$$\vec{y}_{k+1}=\vec{y}_k+\frac{h}{2}(A\vec{y}_k+A\vec{y}_{k+1}) \Rightarrow$$
$$\Rightarrow\left(I-\frac{h}{2}A\right)\vec{y}_{k+1}=\left(I+\frac{h}{2}A\right)\vec{y}_k$$
$$\Rightarrow \vec{y}_{k+1}=\left(I-\frac{h}{2}A\right)^{-1}\left(I+\frac{h}{2}A\right)\vec{y}_k$$
$$\Rightarrow \vec{y}_k=\left[\left(I-\frac{h}{2}A\right)^{-1}\left(I+\frac{h}{2}A\right)\right]^k\vec{y}_0$$

The method is stable if and only if the spectral radius of the matrix $\left(I-\frac{h}{2}A\right)^{-1}\left(I+\frac{h}{2}A\right)$ is less than 1. Using the result of (i) the eigenvalues of the matrix above are given as $\hat{\lambda}_i=\frac{1+\frac{h}{2}\lambda_i^A}{1-\frac{h}{2}\lambda_i^A}$ which all have magnitude less than 1 for $\mathrm{Re}\{\lambda_i^A\}<0$ (since each λ_i^A lies closer to -1 than to 1 on the complex plane) regardless of the value of h. Therefore, the trapezoidal rule is unconditionally stable for this model equation.

3. **Numerical Quadrature [12 pts]**

Consider a real function f that is differentiable on an interval $[a,b]$

i. [3 pts] Find a *quadratic* polynomial $g(x)$ that approximates $f(x)$ on $[a,b]$ in that $f'(a)=g'(a)$, $f'(b)=g'(b)$ and $f\left(\frac{a+b}{2}\right)=g\left(\frac{a+b}{2}\right)$ [Hint : Consider expressing $g(x)$ as a quadratic polynomial of $\left(x-\frac{a+b}{2}\right)$]

ii. [2 pts] Define a numerical quadrature rule for $\int_a^b f(x)dx$ by integrating the interpolant $g(x)$ on $[a,b]$

iii. [3 pts] Prove that this integration scheme has degree of accuracy equal to 3.

iv. [2 pts] Define the corresponding composite quadrature rule for $\int_a^b f(x)dx$ we obtain by subdividing $[a,b]$ into the n sub-intervals $\left[a+k\frac{b-a}{n}, a+(k+1)\frac{b-a}{n}\right]$, $k=0,1,\ldots,n-1$ and applying the basic integration rule on each of them

v. [2 pts] Consider the composite rule of (iv), the composite midpoint rule and the composite Simpson's rule. Under which circumstances would you prefer to use each one of them?

Solutions

i. Let $g(x)=c_2\left(x-\frac{a+b}{2}\right)^2+c_1\left(x-\frac{a+b}{2}\right)+c_0$. Using the given constraints we have

$$\left\{\begin{array}{c} g'(a)=f'(a) \\ g'(b)=f'(b) \\ g\left(\frac{a+b}{2}\right)=f\left(\frac{a+b}{2}\right)\end{array}\right\}\Rightarrow\left\{\begin{array}{c} c_2(a-b)+c_1=f'(a) \\ c_2(b-a)+c_1=f'(b) \\ c_0=f\left(\frac{a+b}{2}\right)\end{array}\right\}\Rightarrow\left\{\begin{array}{c} c_2=\frac{f'(b)-f'(a)}{2(b-a)} \\ c_1=\frac{f'(a)+f'(b)}{2} \\ c_0=f\left(\frac{a+b}{2}\right)\end{array}\right\}$$

Therefore

$$g(x)=\frac{f'(b)-f'(a)}{2(b-a)}\left(x-\frac{a+b}{2}\right)^2+\frac{f'(a)+f'(b)}{2}\left(x-\frac{a+b}{2}\right)+f\left(\frac{a+b}{2}\right)$$

ii. We have

$$\int_a^b f(x)dx\approx\int_a^b g(x)dx=\int_a^b\left[c_2\left(x-\frac{a+b}{2}\right)^2+c_1\left(x-\frac{a+b}{2}\right)+c_0\right]dx=c_2\frac{(b-a)^3}{12}+c_0(b-a)$$

$$\Rightarrow\int_a^b f(x)dx\approx(b-a)f\left(\frac{a+b}{2}\right)+\frac{(b-a)^2}{24}[f'(b)-f'(a)]$$

iii. The interpolant used approximates exactly polynomials of degree up to 2, thus the degree of accuracy is at least 2. We also have

$$\int_a^b x^3 dx = (b-a)\left(\frac{a+b}{2}\right)^3 + \frac{(b-a)^2}{24}\left[3b^2 - 3a^2\right] = \frac{(b-a)(a+b)^3}{8} + \frac{(b-a)^3(a+b)}{8} =$$

$$= \frac{(b-a)(a+b)}{8}\left[(a+b)^2 + (a-b)^2\right] = \frac{(b^2-a^2)}{4}\left(b^2+a^2\right) = \frac{b^4-a^4}{4}$$

which is the exact result. To show that the degree of accuracy is <u>exactly</u> 3, we give the counterexample $f(x) = x^4$ on the interval $[-a, a]$

$$\int_{-a}^{a} x^4 dx = 2a \cdot 0^4 + \frac{(2a)^2}{24}\left[4a^3 + 4a^3\right] = \frac{4}{3}a^5$$

which is <u>not</u> the exact result ($\frac{2}{5}a^5$). Therefore the method is third order accurate

iv. The composite rule is

$$\int_a^b f(x)dx = \sum_{k=0}^{n-1} \int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} f(x) \approx$$

$$\approx \sum_{k=0}^{n-1}\left\{\frac{b-a}{n} f\left(a + (2k+1)\frac{b-a}{2n}\right) + \frac{(b-a)^2}{24n^2}\left[f'\left(a + (k+1)\frac{b-a}{n}\right) - f'\left(a + k\frac{b-a}{n}\right)\right]\right\}$$

$$= \left\{\frac{b-a}{n}\sum_{k=0}^{n-1} f\left(a + (2k+1)\frac{b-a}{2n}\right)\right\} + \frac{(b-a)^2}{24n^2}\left[f'(b) - f'(a)\right]$$

v. If we know the *exact* value of $f'(a)$ and $f'(b)$, the rule we proved in (iv) is third order accurate while only slightly more complex than the midpoint rule and should be preferred. Note that this wouldn't work if we tried to approximate $f'(a)$ and $f'(b)$ from nearby values of f, since this approximation would have an $O(h)$ error leading to an $O(h^3)$ error in the integration formula (same as the midpoint rule).

If we don't know $f'(a)$ and $f'(b)$ and third order accuracy is desired, Simpson's rule is the only option. Nevertheless, if first order accuracy is sufficient (for example if f is very smooth or if the discretization step h is already very small) the midpoint rule is simpler and requires much fewer floating point operations.

Comprehensive Exam: Programming Languages Autumn 2003

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

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

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

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

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

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

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

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

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

_____

(Number)

Prob	# 1	# 2	# 3	# 4	Total
Score					
Max	6	6	8	10	30

1. (6 points) *Garbage Collection.*

   This question is about the design and use of a garbage collector as part of the runtime system.

   (a) (2 points) Assume you have a language like ML, in which all pointers are statically typed. Describe the general operation of a garbage collector that may be run while the application program is stopped. Your collector should only mark objects as garbage and delete them if really are garbage. Does your collector find and delete all inaccessible objects?

   (b) (2 points) Garbage collection for languages like C and C++ is more complicated because casting and unions make pointers difficult to recognize. What changes are needed in the garbage collector you proposed in (a) to support languages like C and C++?

   (c) (2 points) Some languages, like Java, allow simultaneous execution of multiple threads. In a multi-threaded run-time system, it is advantageous to allow the garbage collector to run concurrently with one or more program threads. What changes are needed for your garbage collector to make it concurrent?

2. (6 points) *Subtyping.*

   (a) (3 points) Describe the standard subtyping rules for function types.

   (b) (3 points) Why is a type of mutable cells (ML reference cells, or records with a single assignable field) not subtypable?

3. (8 points) *Method Lookup.*

   In a multiple inheritance language such as C++, one needs to put "deltas" in the virtual function table along with pointers to functions. Explain why the "deltas" are needed in the virtual function table. Illustrate with an example.

4. (10 points) *Implementing Exceptions*

   One way of implementing exceptions is to make a table mapping exception names to code for handlers, for each scope, and store this table on the run-time stack. This has little performance impact at run-time, unless an exception is raised, since the tables can be determined at compile time. However, when an exception is raised, there is some cost. Specifically, if the current activation record contains a handler, control is transferred to this handler. If not, then the exception will have to be "re-raised" in another scope.

   (a) If an exception is raised and there is no handler in the current scope, which pointer in the activation record should be used to find the next scope?

   (b) Can the compiler determine the number of pointers to follow, for a given exception and scope, at compile time? Explain why or why not.

   (c) Optimizing compilers often change the order of instructions for various reasons. Why do languages with exceptions make this kind of optimization more difficult?

   (d) What information would a compiler like to know, at compile time, about a given expression such as a function call, in order to reorder instructions?

   (e) Does Java provide this information for method calls? If so, for all exceptions or just some exceptions?

2003 Programming Languages Comp. Solutions

written by 2006 Ph.D. First-Years

1. *Garbage Collection:*

   (a) You could do mark-and-sweep. Follow pointers because you know what their types are at compile-time. No, it cannot find and delete all inaccessible objects; any garbage collector can at most hope to delete all unreachable objects, but not all inaccessible objects.

   (b) You need to implement conservative garbage collection whereby you treat every value that could possibly point to a region in the heap as a pointer.

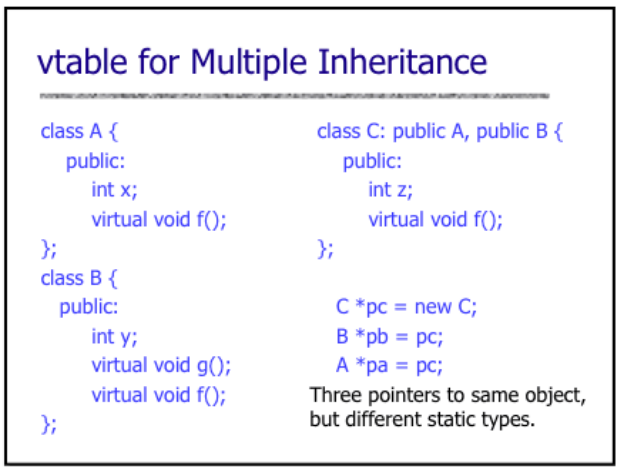
   (c) You need to lock thread-local heaps.

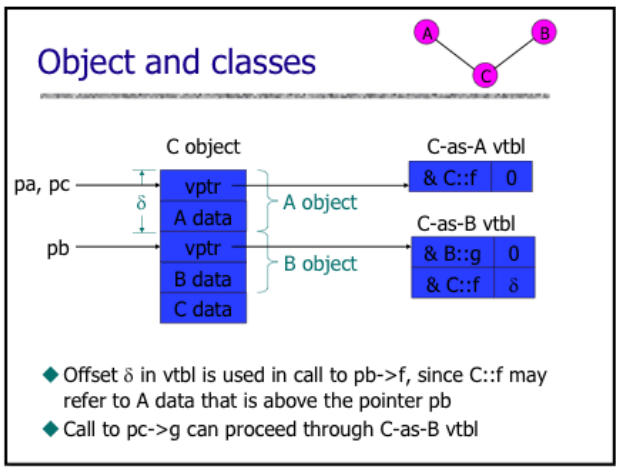
2. *Subtyping:*

   (a) Return types are covariant; function arguments are contravariant. e.g., `circle <: shape`. A function returning a `circle` is a subtype of a function returning a `shape` because if a caller expects a `shape` to be returned, then it's perfectly okay to return a `circle` (covariance). A function taking in a `shape` as an argument is a subtype of a function taking in a `circle` as an argument, because if the caller expects to pass in a `circle`, it's perfectly okay to treat it as simply a `shape` within the function and only use the features that are available in `shape` and ignore the `circle`-specific features (contravariance).

   (b) `ref circle` is not a subtype of `ref shape`, because you can put a `square` in a `ref shape`, but not in a `ref circle`.

3. *Method Lookup:* Deltas are needed in the vtable because the compiler needs to put in a fixed offset in the compiled code when compiling virtual function calls, but when a class inherits from multiple classes, the offset of the function in the subclass will probably be different than the offset in one of the superclasses. The delta is used to go backwards to the beginning of the object so that all fields can be accessed. See below (taken from Prof. John Mitchell's CS242 lecture notes) for an illustrated example.





4. *Implementing Exceptions:*

   (a) The pointer that refers to the base pointer of the next activation record up in the stack (the one for the current method's caller). This is called the *control link* in CS242 lecture notes.

   (b) No, because when an exception is raised at run-time, the activation records on the stack can be different depending on what series of method calls led up to that exception. There is no way for a compiler to know at compile-time what records are going to be on the stack when an exception is raised, and thus how many pointers to follow.

   (c) It's difficult to optimize to re-arrange instructions around side effects in general, and an exception is a particularly nasty side-effect.

   (d) Whether that function is allowed to throw exceptions.

   (e) Yes, only for checked exceptions, though. Unchecked exceptions do not need to be declared in the method declaration.

Stanford University Computer Science Department

Fall 2003 Comprehensive Exam in Software Systems

SOLUTIONS

1. **CLOSED BOOK**: no notes, textbook, computer, PDA, Internet access, etc.
2. **WRITE ONLY IN BLUE BOOKS: No credit** for answers written on these exam pages.
3. **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 short.
5. **All questions are worth the same.**
6. If you need to make assumptions to answer a question, *state them clearly*.

---

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

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

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

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

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

   Serialization or marshalling refers to packing the call and its arguments into a portable format for transmission to the RPC server. It is necessary when you are not sure if the machine architecture of the server is the same as that of the client (and therefore whether types like integer or floating-point numbers have the same representation on both). Marshalling is not needed if the two machines have identical data representations and if the arguments in question do not contain pointers into the client's address space (which would not be safely dereferenceable on the server).

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

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

5) 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, each process usually has its own page table (or page table entries are annotated by PIDs), so the same virtual-to-physical mapping can have different access attributes in the different page tables.

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

Memory-mapped: reads and writes to a designated range of memory cause input or output to a device. A typical use is a video framebuffer since the framebuffer size is static and we're modifying entries in place. Programmed I/O: special instructions like "In" and "Out" cause data to be placed on special I/O lines that are decoded by devices. Uses include accessing device registers and short data transfers for character-mode devices since we are actually generating a data stream.

7) Least-frequently-used (LRU) replacement is often done for filesystem caches, but only a crude approximation of LRU is usually done for page tables. Why the difference?

It's too expensive in time and space to update the LRU information on every memory access (you could do this in the TLB, but you have to worry about flushing the info out on TLB evictions, which are still on much shorter timescales than filesystem caches). For filesystems, the cost of doing this update is usually small compared to the cost of doing the actual file operations.

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

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

9) *Priority inheritance* is when a lower-priority thread or process temporarily acquires the higher priority of another thread that is currently blocked. What problem does priority inheritance solve, and how does it solve it?

Consider threads 1, 2, 3 with decreasing priorities. 3 is running but has a resource needed by 1 (so 1 is blocked and 3 is about to release the resource so 1 can run). But 2 is on the ready queue, so 2 is scheduled. Now 1 cannot proceed because the resource it needs is still held by a lower-priority but not-running thread (3). **Note:** you need 3 threads to illustrate this.
I/O bound processes frequently block till I/O completes, so when they become ready to run again, they will not have been running recently. CPU-bound processes have plenty of time to run during the I/O waits of the I/O bound processes.

10) You compile the following code in C using a typical Unix or Windows C compiler. Suppose you run it and call *func* with x set to –1 (negative one). What happens and why?

```
void func(int x)
{
   int a[10],b[10],c[10];
    ...code to initialize all elements of a[] to 1,b[] to 2,c[] to 3...
   printf("%d\n", b[x]);
}
```

Depending on the implementation of the compiler, this will print either "1" or (more likely) "3". No implementation will throw an exception or give a memory error, unless it stores automatic variables non-contiguously on the stack.

11) You translate the above code to Java, compile and run it, and again call *func(–1)*. Does the behavior differ from the previous case, and if so, how? What is the reason for the difference in behavior (or lack thereof)?

The Java virtual machine specification requires it to do runtime bounds checking, so you'd get an array bounds exception being thrown.

12) Some OSs provided a system call RENAME to give a file a new name. Is there any difference between using this call to rename a file, vs. just copying the file to a new file with the new name and then deleting the old one? If so, describe the difference. If not, roughly sketch why there is no difference.

It's not the same. Rename modifies the file's metadata but does not move the file contents to a different set of disk blocks (unless the file is being moved across filesystems). (Rename is usually faster, but you had to explain *why* it's faster.)

13) With respect to a virtual memory system, give one argument in favor of large page sizes and one argument in favor of small page sizes. Give an example of an appropriate use of each type of page.

Small page sizes are useful when you want to avoid internal fragmentation (wasted space within a page, leading to wasted space overall). Most user data pages fit this description. Large pages are appropriate when you want to minimize page swapping because there is a large set of data that is infrequently swapped or is used all together; kernel image is a typical use for large pages.

14) Describe the structure of an inverted page table and how it differs from a conventional page table. When would you want to use an inverted page table?

Inverted page tables hash from virtual to physical address, allowing the page table to be the size of physical memory rather than the size of the virtual address space. If the virtual address space is very large and/or is sparsely used, inverted tables may be desirable.

15) When programming in a language that uses garbage collection such as Java, is it possible to have memory leaks? If not, why not? If so, give an example of a scenario that would cause a leak to occur.

Yes, circular data structures (such as self-referential linked lists) or references that go out of scope and are never used again cannot be garbage collected (because there is no way for the GC to prove that those data structures are no longer accessible).

THE END