Summary of PhD Candidacy Examinations

Student	Committee	Closed-book exams	Date	Take-home exam	Date	Oral Exam
[illegible]	[illegible] [illegible] [illegible] [illegible] [illegible]	[illegible] [illegible] [illegible]	[illegible]	[illegible] hours	[illegible]	Passed [illegible]
Rosanna [illegible]	B.R. [illegible] [illegible] [illegible] [illegible] [illegible]	[illegible] Part [illegible]	June [illegible]	[illegible] hours	June [illegible]	Passed June [illegible]
Derek [illegible]	B.R. [illegible] [illegible] [illegible] M. [illegible] N. [illegible] (Geog)	2 hours Part 1 2 hours Part 2	June [illegible]	72 hours	June [illegible]	Passed July [illegible]
[illegible] Liu	K.L. Chowdhury (ME) [illegible] B.R. [illegible] [illegible] (Phil) M. [illegible]	2 hours Part 1 2 hours Part 2	August 26, 28	72 hours	Oct 8	Passed Oct 2[illegible]
[illegible]	[illegible] [illegible] [illegible] [illegible] [illegible] [illegible]	3 hours [illegible] 2 hours [illegible]	Oct [illegible]	72 hours	Oct [illegible]	Passed Nov [illegible]
Dave [illegible]	B.R. [illegible] [illegible] [illegible] [illegible] MacDonald P. Prusinkiewicz	1 hour [illegible] 1 hour [illegible] [illegible]	Oct [illegible]	72 hours	Oct [illegible]	Passed Nov [illegible]

1.1 John Aldwinckle: Reading list

FOUNDATIONS

Milner, R. (1990) *Communication and Concurrency.* Prentice Hall International Series in Computer Science

Plotkin, G. *A Structural Approach to Operational Semantics.* Unpublished Lecture Notes

Chellas, B. (1980) *Modal Logic: An Introduction.* Cambridge University Press

BACKGROUND

Pressman, R.S. (1987) *Software Engineering: A Practitioner's Approach.* McGraw-Hill, (2nd edition)

Stallings, W. (1990) *Handbook of Computer Communication Standards: The OSI Model and OSI Related Standards, Volume 1.* Howard W. Sams & Company.

Gehani, N. and McGetrick, A.D. (1986) *Software Specification Techniques.* Addison Wesley.

TECHNIQUES AND APPLICATIONS

Hailpern, B. (1982) *Verifying Concurrent Processes Using Temporal Logic.* Springer-Verlag Lecture Notes in Computer Science, Volume 129.

Reisig, W. (1985) *Petri Nets: An Introduction.* EATCS Monographs, Volume 4.

IEEE Society (1990) "Special Issue on Formal Methods," *IEEE Software* 7(5): 6–67; September.

1.2 **John Aldwinckle: Exam 1** (2 hours)

FOUNDATIONS

Answer 2 questions.

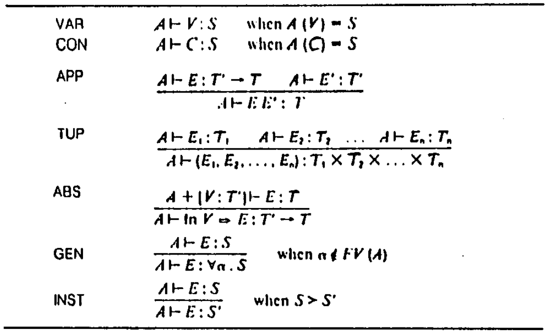
1. An informal description of the send process in the alternating bit protocol is given below:
    1. after accepting a message, send bit b along the transmission line l and set the timer.
    2. if time-out received, send the message again with b
    3. if ack b received on the ack line, prepare to accept another message to be sent with bit 1–b.
    4. if ack (1–b) received, ignore it.

   Give an informal specification of the complementary receive process.

   Give CCS-like specifications of send, receive, and timer processes.

2. Give an informal interpretation of each of the operational rules for typing minimal programs:

A polymorphic type inference system.



VAR	$A \vdash V : S$	when $A(V) = S$
CON	$A \vdash C : S$	when $A(C) = S$
APP	$\dfrac{A \vdash E : T' \rightarrow T \quad A \vdash E' : T'}{A \vdash E\,E' : T}$	
TUP	$\dfrac{A \vdash E_1 : T_1 \quad A \vdash E_2 : T_2 \quad \ldots \quad A \vdash E_n : T_n}{A \vdash (E_1, E_2, \ldots, E_n) : T_1 \times T_2 \times \ldots \times T_n}$	
ABS	$\dfrac{A + [V : T'] \vdash E : T}{A \vdash \mathrm{fn}\ V \Rightarrow E : T' \rightarrow T}$	
GEN	$\dfrac{A \vdash E : S}{A \vdash E : \forall\alpha . S}$	when $\alpha \notin FV(A)$
INST	$\dfrac{A \vdash E : S}{A \vdash E : S'}$	when $S > S'$

   Provide additional operational rules to type the following constructs:

if expr then expr then expr	(if expression)
let x = expr in expr	(let)
let rec x = expr in expr	(recursion)
(expr, expr)	(pair)

   and use them to derive the types (if possible) of

   let f =\x . x in (f 3, f true)
   \x . x x
   Y f = f (Y f) where = : * –> * –> bool

3. How does a modal logic differ from standard first order predicate calculus? (20%)

   Give three examples of types of reasoning that are appropriately modeled by different modal logics. (30%)

   What is a *possible worlds* model of a modal logic? (15%)

   Define the relations between possible worlds that are used to characterize particular modal logics. (35%)

1.3 John Aldwinckle: Exam 2 (2 hours)

TECHNIQUES AND APPLICATIONS

Answer 2 questions.

1. Given Hailpern's axioms and inference rules, (see attached sheets) prove that:

   $\Box\Box P = \Box P$

   $\Diamond\Diamond P = \Diamond P$

   $\bigcirc(P \wedge Q) = \bigcirc P \wedge \bigcirc Q$

   $\bigcirc\Box P = \Box\bigcirc P$

2. What is a Petri net? (20%)

   Characterize systems that are appropriately modeled by Petri nets. (20%)

   Define the following terms in the context of Petri nets: (40%)

   a. safeness
   b. boundedness
   c. conservation
   d. liveness
   e. reachability
   f. coverability

3. Anthony Hall says:

   > "Formal methods are controversial. Their advocates claim they can revolutionize development. Their detractors think they are impossibly difficult. . . . . some of the beliefs about formal methods have been exaggerated and have acquired almost the status of myths."

   The seven most prevalent formal–methods myths are stated as follows:

   1. Formal methods can guarantee that software is perfect.
   2. They work by proving that programs are correct.
   3. Only highly critical systems benefit from their use.
   4. They involve complex mathematics.
   5. They increase the cost of development.
   6. They are incomprehensible to clients.
   7. Nobody uses them for real projects.

   Comment critically on each of these and hence, or otherwise, derive an alternative list which may be more believable.

1.4 John Aldwinckle: Exam 3 (2 hours)

BACKGROUND

Answer 2 questions.

1. What are the primary functions of a communications protocol? (25%)

   Analyse the notion of a layered protocol in terms of:

   a. the requirements satisfied by layering; (15%)
   b. the underlying principles; (25%)
   c. the practical significance; (15%)
   d. exemplary practice and experience. (20%)

2. a. What is a program specification? Explain program specification at each stage of software development. (20%)
   b. Discuss the advantages of both formal and informal specifications. (20%)
   c. Explain how specification relates to: (20%)
      i. implementation
      ii. verification
   d. OBJ is a formal language for writing and testing algebraic program specification. Describe, with examples, the particular features of OBJ which are helpful in creating correct specifications. (40%)

3. The following software specification principles have been developed by Balzer and Goldman:

   #1. Separate functionality from implementation.
   #2. A process-oriented systems specification language is required.
   #3. A specification must encompass the system of which the software is a component.
   #4. A specification must encompass the environment in which the system operates.
   #5. A system specification must be a cognitive model.
   #6. A specification must be operational.
   #7. The system specification must be tolerant of incompleteness and augmentable.
   #8. A specification must be localized and loosely coupled.

   Give a full explanation and critical discussion of each.

1.5 John Aldwinckle: Take-home exam (72 hours)

There is a common perception that there is a gap between user requirements and system implementation to satisfy them. Propose a research project to investigate:

a. whether this gap exists and how it can be described;

b. the role of formal methods in addressing this problem;

c. the practical industrial impact of methodologies for bridging this gap.

Prepare a detailed research proposal citing relevant work and techniques with particular attention to criteria for evaluating the success of such a project.

2.1 Rosanna Heise: Reading list

MACHINE LEARNING

Cohen, P.R. and Feigenbaum, E.A. (Editors) (1982) *The handbook of artificial intelligence Volume III*, Addison-Wesley, pg. 323–511, 1982.

Issues	325–334
Rote learning	335–344
Learning from instruction	345–359
Learning from examples	360–511

Gennari, J.H., Langley, P., and Fisher, D. (1990) "Models of incremental concept formation," in *Machine Learning: Paradigms and Methods*, edited by J. Carbonell. MIT Press, pp 11–61.

Russell, S.J. and Grosof, B.N.(1990) "Declarative bias: an overview," in *Change of Representation and Inductive Bias*, edited by D.P. Benjamin, pp 267–308.

PLANNING

Hendler, J., Tate, A. and Drummond, M. (1990) "AI planning: systems and techniques," *AI Magazine 11*(2): 61–77; Summer.

Genesereth, M.R. and Nilsson, N.J. (1987) *Logical foundations of artificial intelligence.* Morgan Kaufmann, Los Altos, CA.

Chapter 4	Resolution
Chapter 5	Resolution strategies
Chapter 11	State and change
Chapter 12	Planning
Chapter 13	Intelligent-agent architecture

Ginsberg, M.L.(1989) "Universal planning: an (almost) universally bad idea," *AI Magazine 10*(4): 40–44; Winter.

Chapman, D.(1989) "Penguins can make cake," *AI Magazine 10*(4): 45–50; Winter.

Schoppers, M.J.(1989) "In defense of reaction plans as caches," *AI Magazine 10*(4): 51–60; Winter.

Ginsberg, M.L.(1989) "Universal planning research: a good or bad idea?," *AI Magazine 10*(4): 40–44; Winter.

HUMAN TASK/SKILL LEARNING

Magill, R.A.(1989) *Motor learning: concepts and applications.* Wm. C. Brown, Dubuque, Iowa.

Chapter 1	Introduction to motor skills and motor learning research
Chapter 2	Introduction to motor learning
Chapter 3	Controlling movement
Chapter 4	Attention

Anderson, J.R. "Skill acquisition: compilation of weak method problem solutions," *Psych. Review 194*(2): 192–210.

Case, R. (1980) "The Underlying Mechanism of Intellectual Development," *Cognition, Development, and Instruction.*

Carey, S. "Reorganization of Knowledge in the course of Acquisition."

Chi, M.T.H. "Children's Lack of Access and Knowledge Reorganization: An example from the concept of Animism," in *Memory Development: Universal Changes and Individual Differences*, edited by M. Perlmutter and F.E. Weinert.

Neumann, O.(1987) "Beyond capacity: a functional view of attention," Chapter 14 of *Perspectives on Perception and Action*, edited by H. Heurer and A.F. Sanders; 361–394. Lawrence Erlbaum Associates.

2.2 Rosanna Heise: Exam 1 (3 hours)

PART ONE

Answer both questions.

1. Describe and distinguish the following types of planning, state the advantages and disadvantages of each, and give the details of initial knowledge, the method of planning and the resulting plan for the three-disk towers of Hanoi problem.

   Green's method, goal regression, universal planning, reactive planning.

2. a Describe and distinguish the following kinds of agent: tropistic, hysteretic, knowledge-level, and deliberate.

   b Consider an agent as a cart in a 3 by 3 Maze World with the ability to observe its own location exactly and the relative location of the gold (in the cart, in the same cell, or elsewhere). The goal is to find the gold, put it in the cart, move to the rightmost bottom square, and take the gold out. Design an initial internal state, an action function, and an internal-state update function for a hysteretic agent that allows it to solve the Maze World problem when started in any external state.

PART TWO

Answer 2 questions.

1. That control is passed back and forth from conscious, effortful processes to automatic ones is a primary assertion of cognitive theory. Describe John Anderson's model of how this occurs and identify way sin which artificial systems might use this principle.

2. Compare and contrast the notions of mental restructuring postulated by Sue Carey and Michelene Chi. Describe the benefits that might result from the application of each position in the area of machine learning. Discuss the relative merits of each.

3. Summarize the central postulates of Case's theory of cognitive development and discuss the applicability of each to machine learning. Outline briefly and benefits that might result from such application.

2.3 Rosanna Heise: Take-home exam (56 hours)

Describe the use of focusing mechanisms to fruitfully bias the search in concept learning systems. Be sure to

1. explain the need for dynamic bias
2. relate your comments to human learning whenever possible
3. explain the bias in each system

Include the following systems where relevant: EPAM, UNIMEM, COBWEB, CLASSIT, Samuel's checkers players, Mostow's operationalizer, Control and Pattern recognition systems, candidate elimination, BACON, ID3, INDUCE, SPARC, AQ11, Meta-dendral, AM, Waterman's poker player, HACKER, LEX, grammatical inference systems. You should largely restrict yourself to the Handbook of AI pages 325–511, the paper by Russell and Grosof on page 267 of Benjamin's "Change of Representation and Inductive Bias," and Gennari et al on page 11 of Carbonell's "Machine Learning: Paradigms and Methods" (these are from your reading list and you won't have time to go and read more details elsewhere), plus any material that is relevant from Utgoff's STABB or your own ETAR system.

3.1 Debbie Leishman: Reading list

DESIGN

Broadbent, G. (1973) Design in Architecture. Wiley.

Chapter 1	The architect as designer
Chapter 2	The architect at work
Chapter 5	Models
Chapter 7	Human science techniques
Chapter 8	Basic needs
Chapter 9	Social needs
Chapter 10	New problem–solving techniques
Chapter 12	New maths
Chapter 13	Development of design methods
Chapter 14	New design processes
Chapter 16	The design spectrum
Chapter 17	Creative techniques
Chapter 19	An environmental design process
Chapter 20	The derivation of architectural form
Chapter 21	Prospect

Coyne, R.D., Rosenman, M.A., Radford, A.D., Balachandran, M. and Gero, J.S. (1990) *Knowledge Based Design Systems* Addison–Wesley.

Chapter 1	Introduction to Knowledge Based Design
Chapter 2	A Knowledge Based Model of Design
Chapter 3	Representation and Reasoning
Chapter 4	Reasoning in Design
Chapter 5	The Interpretation of Designs
Chapter 6	Producing Designs
Chapter 7	Design Processes
Section 8.4	Analogical Reasoning in Design Systems

Mostow, J. (1989) "Design by Derivational Analogy: Issues in the Automated Replay of Design Plans," *Artificial Intelligence 40*: 119–184.

ANALOGY

Hall, R. (1989) "Computational Approaches to Analogical Reasoning: A Comparative Analysis," *Artificial Intelligence 39*: 39–120

Prieditis, A. (Editor) (1988) *Analogica*. Morgan Kaufmann.

Chapter 3	Analogical Inference and Analogical Access
Chapter 4	Toward a Computational Model of Purpose
Chapter 6	Constrained Semantic Transference

Helman, D.H. (Editor) (1988) *Analogical Reasoning*. Kluwer, Boston.

Part 1	Categories and Analogies
Part 2	Dimensions of Analogy
Part 3	Determination, Uniformity, and Relevance
Part 3	Analogy by Similarity

KNOWLEDGE REPRESENTATION AND REASONING

Davis, E. (1990) *Representations of Commonsense Knowledge*. Morgan Kaufmann, San Mateo, California.

Chapter 1	Automating Common Sense
Chapter 2	Logic
Chapter 3	Plausible Reasoning (except sections 3.34,3.35,3.36)
Chapter 5	Time
Chapter 6	Space
Chapter 9	Plans and Goals

Sowa, J.F. (1984) *Conceptual Structures: Information Processing in Mind and Machine*. Addison–Wesley, London.

Chapter 1	Philosophical Basis
Chapter 2	Psychological Evidence
Chapter 3	Conceptual Graphs
Chapter 4	Reasoning and Computation
Chapter 6	Knowledge Engineering

Ellman, T. (1989) "Explanation–Based Learning: A Survey of Programs and Perspectives," *Computing Surveys 21*(2): 163–221; June.

3.2 Debbie Leishman: Exam 1 (2 hours)

Answer 2 questions.

1. a. Discuss the connotations of the term "design", defining the term as concisely as possible, and relating your discussion to that in the literature.
   b. Use your definition to analyze the design process in architecture, either overall or in some major component, classifying the types of activity involved at each stage.
   c. Use your classification to discuss the role of computational systems in supporting the designer and design process.

2. a. Briefly analyze four major areas of commonsense reasoning and the issues involved in their support through computational knowledge representation and reasoning.
   b. Discuss how conceptual graphs support knowledge representation and reasoning in each of these four areas.

3. a. Give a concise definition of analogical reasoning.
   b. Use your definition to analyze two distinct examples of common human reasoning by analogy.
   c. Use your definition to develop a computational framework for reasoning by analogy.
   d. Use this framework to outline briefly the state of the art in the main components required to implement analogical reasoning.
   e. Discuss the limitations of analogical reasoning based on similarity.

3.3. **Debbie Leishman: Exam 2** (2 hours)

Answer 2 questions.

4. a. Analyze the knowledge representation and reasoning processes involved in design, maintaining a broad a perspective on design as possible but specifying particular design domains to the extent you find this necessary.
   b. Using your analysis, discuss the state of the art in knowledge representation and reasoning in artificial intelligence products and research in terms of its potential support of design processes.

5. a. Outline briefly the major features of explanation-based learning.
   b. Outline briefly the major features of design by derivational analogy.
   c. Discuss how the two techniques might be combined in the support of the design process.

6. Critically discuss the contributions of the following researchers to the development of analogical reasoning processes in the computer, highlighting the objectives of the research, its unique contributions, and the limitations of the approach.
   a. Gentner's structure mapping and systematicity principle.
   b. Russell's determinations.
   c. Kedar-Cabelli's purpose-directed analogy.
   d. Indurkhya's constrained semantic transference.
   e. Thadgard's dimensions of analogy.

3.4 Debbie Leishman: Take-home exam (72 hours)

Write an essay of about 5,000 words as specified below.

A research group is undertaking a project on the support of human design activities. One objective is the support of building design but it is hoped to make the support system as widely applicable as possible. Another objective is not to duplicate commercial tools concerned with "computer-aided design" but to identify major areas of design not currently supported.

Write an initial briefing report for the team on the issues involved, background in previous analyses of the nature and support of the design process, the relevant computational technologies, the research issues, and some realistic objectives for the project.

Structure the report to be suitable for circulation to management as well as researchers with a concise executive summary, clear separation of issues, and concise, clear and well-supported conclusions.

4.1 Mengchi Liu: Reading list

Ullman, J.D. (1988) *Principles of Database and Knowledge-Base Systems*. Computer Science Press.

Chapter 1	Databases, Object Bases, and Knowledge Bases
Chapter 2	Data Models for Database Systems
Chapter 3	Logic as a Data Model
Chapter 4	Relational Query Languages
Chapter 5	Object-Oriented Databases Languages

Gupta, R. and Horowitz, E. (Editors) (1991) Object-Oriented Databases with Applications to Case, Networks and VLSI CAD. Prentice Hall.

Chapter 1	A Guide To The OODB Landscape
Chapter 2	A Perspective on Object-Oriented and Semantic Database Models and Systems
Chapter 3	Algorithmic and Computational Aspects of Object-Oriented Schema Design
Chapter 7	An Overview of Existing Object-Oriented Database Systems

Atkinson, M.P., Buneman, P. and Morrison, R. (1988) *Data Types and Persistence*. Springer-Verlag.

Chapter 7	Data Types for Database Programming
Chapter 8	The Type System of Galileo
Chapter 9	Integrating Data Type Inheritance into Logic Programming
Chapter 10	Class Hierarchies in Information Systems: Sets, Types, or Prototypes

Ceri, S. and Tanca, G.T. (1990) *Logic Programming and Databases*. Springer-Verlag.

Chapter 1	Logic Programming and Databases: An Overview
Chapter 2	A Review of Relational Databases and Prolog
Chapter 6	Syntax and Semantics of Datalog
Chapter 7	Proof Theory and Evaluation Paradigms of Datalog
Chapter 11	Extensions of Pure Datalog

Lloyd, J.W. (1987) *Foundation of Logic Programming*. Springer-Verlag, second edition.

Chapter 1	Preliminaries
Chapter 2	Definite Programs
Chapter 3	Norman Programs
Chapter 5	Deductive Databases

Minker, J. (1988) *Foundation of Deductive Databases and Logic Programming*. Morgan Kaufmann.

Chapter 1	Negation in Logic Programming
Chapter 2	Towards a Theory of Declarative Knowledge
Chapter 4	On the Declarative Semantics of Logic Programs
Chapter 5	On the Declarative Semantics of Deductive Database and Logic Programs

Hatcher, W. S. (1982) *The Logical Foundations of Mathematics*. Pergamon Press.

Chapter 1	First-Order Logic
Chapter 3	Frege's System and The Paradoxes
Chapter 5	Zermelo-Fraenkel Set Theory

Jackson, P., Reichgelt, H. and van Harmelen, F. (1989) *Logic-Based Knowledge Representation*.MIT Press.

Chapter 1	Introduction
Chapter 2	A Classification of Meta-Level Architectures
Chapter 3	The Architecture of Socrates
Chapter 4	Applications of Socrates

4.2 Mengchi Liu: Exam 1 (2 hours)

Answer 2 questions.

1. a. For both the following programs give: their Herbrand base and Herbrand Universe, their minimal model and show the first few steps of constructing the minimal model using the T_P mapping.

```
%program 1
p(0).
p(f(X)) <- p(X).

r(X,Y) <- p(X), p(Y).
s(X) <- r(X,X).

%program 2
p(X) <- q(a).

q(a) <- r(X).

r(0).
r(f(X)) <- r(X).
```

   b. Construct a normal program which has more than one minimal model and which can be stratified in more than one way. Give all the minimal models for the program and all the possible stratifications. Show which of the minimal models is computed when a fix-point is constructed using the stratifications. Show the steps in constructing the fix-point.

   c. Construct a normal program which is not stratified but which is locally stratified and has more than one minimal model. State which of the minimal models are perfect and justify your answer.

2. a. Outline the important features of Codd's relational database model.
   b. Define the 5 primary operations of relational algebra, illustrating each with a database query example.
   c. Outline the important features of object-oriented database models.
   d. Summarize the similarities and differences between relational and object-oriented database models.

3. a. Outline an algebraic formulation of data types applicable to databases in terms of inheritance and partial functions.
   b. Describe a system for integrating inheritance into logic programming.
   c. Discuss how the notions of class and object may be implemented in terms of data types and inheritance.

4.3 Mengchi Liu: Exam 2 (2 hours)

Answer 2 questions.

4. a. Describe what is meant by an SLD-tree and an SLD-refutation procedure. State explicitly in what ways SLD-refutation is restricted in the deduction steps it can make compared with general clausal resolution.
   b. Define what is meant by a fair SLD-refutation procedure. Give an example of a definite Prolog program (and an associated goal) which has a finite SLD-tree using a fair computation rule and an infinite one using an unfair rule (exhibit both trees and describe the computation rules used).
   c. Find a definite program and a definite goal such that each SLD-tree for them has two success branches, but no depth-first search will ever find both success branches no matter what the computation rule.

5. a. Describe the background to the development of ZF set theory, particularly in terms of what problems with previous formulations it was intended to overcome.
   b. Give a synopsis of the main definitions and axioms of ZF set theory.
   c. Discuss briefly the limitations of ZF set theory as a foundation for the set data type in databases.

6. a. Describe the problems of incorporating logical negation in a deductive database system.
   b. Describe "negation as failure" and analyze its semantics.
   c. Describe stratification and discuss its role in implementing negation in deductive databases.

4.4 Mengchi Liu: Take-home exam (72 hours)

Write an essay of about 5,000 words as specified below.

A number of papers on deductive databases research have used the term "higher-order' to describe features concerned with sets and schema, but have not necessarily shown the need for a higher-order logic to support these features, or shown how they cope with the problems of going beyond first-order logic.

Write an expository survey paper about higher-order logics aimed at research groups developing deductive database systems providing features for representing complex objects and sets, and dealing with queries about such objects, their schema, and sets.

The paper should provide a clear exposition of first-order logic, higher-order logic, the differences between them, the problems of higher-order logic, and developments in higher-order logic.

The requirements of deductive databases for representing complex objects and sets that lead to the necessity of logical constructs going beyond first-order should be clearly defined. In particular, the representations and operations that can be achieved using only first-order logic should be clearly stated.

The paper should discuss the tractability of the deductive processes associated with the higher-order features.

Structure the paper with a concise summary, clear separation of issues, and concise, clear and well-supported conclusions.

5.1 Thong Phan: Reading list

FOUNDATIONS OF ARTIFICIAL INTELLIGENCE

Genesereth, M.R. and Nilsson, N.J. (1987) *Logical foundations of artificial intelligence.* Morgan Kaufmann, Los Altos, CA.

Chapter 1	Introduction
Chapter 2	Declarative knowledge
Chapter 3	Inference
Chapter 4	Resolution
Chapter 5	Resolution strategies
Chapter 7	Induction

Stickel, M.E. (1988) "Resolution theorem proving," in *Annual Review of Computer Science 3*: 285–316.

Korf, R.E. (1988) "Search: A survey of recent results," in *Exploring Artificial Intelligence*, edited by H.E. Shrobe, Morgan Kaufmann, Los Altos, pp. 197–237.

PRACTICAL MACHINE LEARNING

Cohen, P.R. and Feigenbaum, E.A. (Editors) (1982) *The handbook of artificial intelligence Volume III*, HeurisTech Press, Stanford, CA, Chapter 14.

Issues	360–372
Learning single concepts	383–419
Learning multiple concepts	20–451

THEORETICAL FOUNDATIONS OF MACHINE LEARNING

Gold, E.M. (1967) "Language identification in the limit," *Information and Control 10*: 447–474.

Angluin, D. and Smith, C.H. (1983) "Inductive inference: theory and methods," *Computing Surveys 15*(3): 237–269; September.

Valiant, L.G. (1984) "A theory of the learnable," *Communications of the ACM 27*(11): 1134–1142; November.

Angluin, D. (1988) "Queries and concept learning," *Machine Learning 2*(4): 319–342; April.

Littlestone, N. (1988) "Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm," *Machine Learning 2*(4): 285–318; April.

Angluin, D. and Laird, P. (1988) "Learning from noisy examples," *Machine Learning 2*(4): 343–370; April.

FINITE FIELDS

Lidl, R. and Niederreiter, H. (1986) *Introduction to finite fields and their applications.* Cambridge University Press, Cambridge, England.

Chapter 1	Algebraic foundations
Chapter 2	Structure of finite fields
Chapter 3	Polynomials over finite fields

ALGEBRAIC COMPLEXITY

Valiant, L.G. (1979) "Completeness classes in algebra," *Proceedings 11th Annual ACM Symposium Theory of Computing*, 249–261.

von zur Gathen, J. (1987) "Feasible arithmetic calculations: Valiant's hypothesis," *J Symbolic Computation 4*: 137–172.

von zur Gathen, J. (1988) "Algebraic complexity theory," in *Annual Review of Computer Science 3*: 317–347.

5.2 Thong Phan: Exam 1 (3 hours)

FOUNDATIONS OF ARTIFICIAL INTELLIGENCE

Answer Question 1 or Question 2.

1. Describe the A* and the Iterative-Deepening A* (IDA*) algorithms for heuristic search.
   a. Under what conditions will A* find an optimal solution?
   b. What is the advantage of IDA* over A*?
   c. How does the running time of the two methods compare as the size of the search space increases?

2. a. What is the "resolution rule" for first-order predicate calculus?
   b. Describe the "ordered resolution" search strategy for resolution.
   c. Describe briefly the execution strategy of the standard Prolog interpreter.
   d. Show how Prolog interpretation can be viewed as a form of ordered resolution.
   e. Ordered resolution is an "incomplete" resolution strategy. What does this mean? How does it affect the Prolog interpreter's ability to prove general theorems in first-order predicate calculus?

PRACTICAL MACHINE LEARNING

Answer Question 3 or Question 4.

3. Describe Mitchell's candidate elimination algorithm for learning from examples.
   a. Explain how it would learn the concept of any circle in a version space that represents objects by their size (small or large) and shape (square, circle or triangle) given the positive example "small circle," the negative example "large triangle," and the positive example "large circle."
   b. Consider the problem of inducing an expression (for example, sqrt(*x*) + *y*) over a known list of functions and a known list of constants, given a list of input-output pairs. Discuss the difficulties of using candidate elimination to solve this problem.

4. Describe the algorithm and rule-space operators used by BACON-1 for learning functional expressions (particularly scientific laws) from examples.
   a. Explain how BACON-1 would go about inducing the law $I3^3/I2^2 = k$ from the training instances below.

	Features		
Instance #	I1	I2	I3
1	M	1	1
2	V	8	4
3	E	27	9

   b. Discuss the advantages and disadvantages of BACON.

THEORETICAL FOUNDATIONS OF MACHINE LEARNING

Answer Question 5 or Question 6.

5. State precisely Valiant's (1984) definition of a class X of programs being "learnable" (i.e., "pac-learnable") with respect to a given learning protocol.

   Define the following classes of programs and state the results obtained by Valiant (1984) about their learnability under appropriate learning protocols:

   a. general CNF (conjunctive normal form) expressions;
   b. k-CNF expressions;
   c. monotone DNF (disjunctive normal form) expressions;
   d. μ-expressions.

6. List and describe the six different types of query that a learning system might use, that were identified by Angluin (1988).

   Summarize the results that Angluin (1988) obtained on the learning of k-CNF and k-DNF formulas using different query types, in terms of

   a. the minimal set of query types that has been shown to suffice for efficient exact identification;
   b. the maximal set of query types for which an exponential lower bound on exact identification has been obtained.

5.3 Thong Phan: Exam 2 (2 hours)

FINITE FIELDS

Answer Question 7 or Question 8.

You may use the following results without proof for these questions.

> For every prime p and every positive integer n there exists a finite field with p^n elements. Any finite field with $q=p^n$ elements is isomorphic to the splitting field $\mathbb{F}_q$ of $x^q - x$ over $\mathbb{F}_p$.
>
> For every finite field $\mathbb{F}_q$ the multiplicative group $\mathbb{F}_q^*$ of nonzero elements of $\mathbb{F}_q$ is cyclic.

7. a. Give the definition of a field.
   b. Show that if f is an irreducible polynomial of degree n over a finite field $\mathbb{F}_q$ then $\mathbb{F}_q[x]/(f)$ is a finite field with q^n elements.
   c. Show that the rings $\mathbb{F}_{11}[x]/(x^2 + 1)$ and $\mathbb{F}_{11}[x]/(x^2 + x + 4)$ are both finite fields, with 121 elements, and that they are isomorphic.
   d. Prove or disprove: "Every element of $\mathbb{F}_{11}$ has at least one square root in $\mathbb{F}_{11}[x]/(x^2 + 1)$."

8. a. Give the definition of the order of a polynomial $f \in \mathbb{F}_q[x]$ over a finite field $\mathbb{F}_q$.
   b. Show that every nonzero polynomial $f \in \mathbb{F}_q[x]$ has a *finite* order over $\mathbb{F}_q$.
   c. Compute the order of the polynomial $x^3 + 2x + 2 \in \mathbb{F}_3[x]$ over $\mathbb{F}_3$.

ALGEBRAIC COMPLEXITY

Answer Question 9 or Question 10.

9. a. Define each of the following classes: p-expressible, p-computable, p-definable, qp-expressible, qp-computable, and qp-definable.
   b. Let $f = (f_n)_{n \in \mathbb{N}}$ be a family of polynomials $f_n \in F[x_1, \ldots, x_{v(n)}]$ such that $v(n)$ and the degree of f_n are bounded by a polynomial in n. Prove that f is qp-computable if and only if f is qp-expressible.
   c. State Valiant's hypothesis.

10. a. Show that the following decision problem is NP-complete.

    Instance: Reasonable encodings of 2 boolean circuits A and B.
    Question: Do A and B compute different functions?

    b. Recall:
       1. The *size* of a boolean circuit is the number of gates in the circuit.
       2. An equivalence query E_f ("Is g equivalent to f?") receives as input a reasonable encoding of a circuit that computes a boolean function g, and compares g to a fixed boolean function f. If $f = g$ then E_f returns yes, otherwise E_f returns a point x such that $f(x) \neq g(x)$.

    Show that the assumption: "For every boolean function f there is a Turing Machine M_f that answers the equivalence query E_f using time polynomial in the size of its input," implies P = NP.

5.4 Thong Phan: Take-home exam (72 hours)

Answer *all* of the following questions.

11. This questions has several parts that taken together explore the thesis of your research proposal and some possible extensions.
    a. Show that the number of circuits of size n with k inputs where gates are in $\{\wedge, \vee, \neg\}$ is bounded by $2^{3n\log(n+k)}$ (for sufficiently large n).

       Hint: Assume gates are numbered 1 through n and that gate i can access variables, constants 0 and 1, and values produced by gates numbered 1 through $i-1$.
    b. Expand on a technique described in an earlier draft of your research proposal to prove the thesis of your current research proposal. Specifically, describe an algorithm that learns an unknown boolean function using at most $f(n,k)$ membership and equivalence queries, where n is the size of the smallest circuit that computes the function, k is the arity of the function, and $f(n,k)$ is a polynomial in n and k.
    c. Assume $\wedge$ and $\vee$ gates can have unbounded fan-in (so that, for example, the formula $x_1 \vee x_2 \vee x_3$ has size 1). Does the result of part (b) still hold? Of course you must defend your answer.
    d. Examine the results of part (b) when equivalence queries are constrained to be polynomial size functions.
    e. Examine the results of part (b) when the charging mechanism is:
       i. $\log s$ to ask an equivalence query consisting of a function of size s.
       ii. linear in s to ask an equivalence query consisting of a function of size s.

12. In this question you will explore hypothesis 1 of your proposal.

    For any subset S of $\{1, 2, ..., k\}$ let

    $$f_S(x_1, x_2, ..., x_k) = \wedge_{i \in S} x_i.$$

    Let

    $$\mathbb{S}_k = \{f_S: S \text{ is a subset of } \{1, 2, ..., k\}\}.$$

    You will consider the cost of learning functions in $\mathbb{S}_k$. In particular, you will count the number of equivalence and membership queries used by an algorithm to learn a function from this class in the worst case. Your result will apply even if the algorithm "knows" that the function being learned belongs to $\mathbb{S}_k$, and no constraints are placed on the equivalence queries that can be used.
    a. Show that $\mathbb{S}_k$ contains 2^k functions.
    b. Show that each function in $\mathbb{S}_k$ has size at most $k-1$.
    c. Prove that at least k equivalence and/or membership queries are needed in the worst case by any algorithm to learn a function in $\mathbb{S}_k$. One way to do this is to develop an "adversarial lower bound" for the problem by considering the following "game."
       - The algorithm is actually communicating with an adversary who is responsible for giving plausible answers to queries, and who wishes to make the algorithm use as many queries as possible.

- The adversary doesn't start with a particular function in mind when answering queries. Instead, the adversary keeps in mind a set of functions that are consistent with all the information given away so far. Whenever the algorithm makes a query, the adversary examines the remaining set of candidate functions and chooses a response that doesn't contradict previous answers. After making a response, the adversary must reexamine the set of candidate functions and eliminate any functions that disagree with the response.
- The adversary always tries to choose its response so that as many candidates are left over after each response as possible. In particular, the adversary will try to follow a strategy that ensures that two or more candidates remain after $k-1$ queries. Then, the adversary can truthfully say "No" to any equivalence query (and give a counterexample), forcing the game to continue and forcing the algorithm to use more queries before a function has been learned.
- Finally, note that this will imply a lower bound on the number of queries needed in the worst case. The algorithm will not be able to distinguish between an adversary of the type given above, and an oracle for the final candidate — which could give exactly the same responses as the adversary's.

Hints

- It may be helpful to think of the set of remaining candidate functions at any stage as being represented by a vector of length k, with each entry having one of the values "In", "Out", or "Unknown". If the ith entry is "In" then it has been established that $i \in S$. Therefore for $(\alpha_1, \alpha_2, ..., \alpha_k) \in \{0, 1\}^k$, $f_S(\alpha_1, \alpha_2, ..., \alpha_k) = 0$ if $\alpha_i = 0$. If the ith entry is "Out" then it has been established that $i \notin S$. Therefore, there will exist at least one input $(\alpha_1, \alpha_2, ..., \alpha_k)$ such that $\alpha_i = 0$ and $f_S(\alpha_1, \alpha_2, ..., \alpha_k) = 1$. (Consider, in particular, the case that $\alpha_i = 0$ and $\alpha_j = 1$ for all $j \neq i$.) If the ith entry is "Unknown" then it hasn't been determined yet whether i is in S.
- Consider a case study of the adversary's responses to each type of query.

d. Now consider a slightly different situation. The kind of learning algorithm that can be used and the class of functions $\mathbb{S}_k$ are exactly the same as before. The only change is that the oracle is now a benevolent teacher, who knows what algorithm the learner is using and wants to teach a function with as few queries as possible, rather than a hostile adversary. Of course, the teacher must only communicate with the learning algorithm through correct responses to the learner's queries.

Describe a learning algorithm, and a corresponding teaching strategy for the oracle, with the following property (and prove that this property holds).

Working with your teaching strategy, your algorithm can learn any function in $\mathbb{S}_k$ using at most two equivalence queries in the worst case. No membership queries are required.

13. Let $H_{n,m}$ be the set of boolean functions $f(x_1, ..., x_n)$, each with size m, and all belonging to some class of functions (e.g. monotone DNF). A set S_f of examples is called *representative* for a function $f \in H_{n,m}$ if no function $g \in H_{n,m}$ satisfies S_f except f.

a. For two different function classes $H_{n,m}$, give examples of a function f and a set of representative examples for it.

b. In general, there will be many different sets of representative examples for a given function. Discuss, for example functions f, the number of such sets and the size of maximal and minimal sets.

c. Evaluate the applicability of the halving strategy in a version space (Nilsson & Genesereth, 1987, §7.3) to the problem of determining a set of representative examples.

d. Discuss, in as much detail as you can, other strategies for determining a set of representative examples.

e. How does the restriction to boolean functions affect the definition of representative sets and the ease of finding them?

6.1 Dave Maulsby: Reading list

MACHINE LEARNING

Carbonell, J. (Editor) (1990) *Machine learning: paradigms and methods*. MIT Press.

Introduction	
Anderson	A theory of the origins of human knowledge
Gennari *et al.*	Models of incremental concept formation
Langley and Zytkow	Data-driven approaches to empirical discovery
Minton *et al.*	Explanation-based learning: a problem solving perspective
Mostow	Design by derivational analogy

Kodratoff, Y. and Michalski, R. (Editors) (1990) *Machine learning, vol. 3*. Morgan Kaufmann.

Bergadano, and Giordana	Guiding induction with domain theories
Michalski and Kodratoff	Research in machine learning: recent progress, classification of methods, and future directions
Rivest and Schapire	A new approach to unsupervised learning in deterministic environments

J. Shavlik, T.G. and Dieterrich, T. (Editors) (1990) *Readings in machine learning*. Morgan Kaufmann.

Chapter 1	Introduction
Chapter 2	Introduction
Chapter 5	Introduction
Gentner	The mechanisms of analogical learning
Mitchell	The need for biases in learning generalizations
Mitchell	Generalization as search
Porter *et al.*	Concept learning and heuristic classification in weak-theory domains
Simon and Lea	Problem solving and rule induction: a unified view
Valiant	A theory of the learnable

CONVERSATION ANALYSIS, CONTENT ANALYSIS, RESEARCH METHODS

Allen, D.E. and Guy, R.F. (1974) *Conversation analysis: the sociology of talk.* Mouton Press.

Chapter 6	Selected properties of the verbal exchange

Carney, T.F. (1972) *Content analysis: a technique for systematic inference from communications.* University of Manitoba Press.

Chapter 2	Content analysis: how to make documents answer your questions reliably
Chapter 3	Content analysis and general semantics
Chapter 4	Content analysis and the "new look" in psychology: selective perception and models

Cohen, L. and Manion, L. (1989) *Research methods in education, 3rd ed.* Croom Helm.

Chapter 8	Experiments, quasi-experiments and single-case research
Chapter 10	Accounts
Chapter 14	Personal constructs

USER INTERFACE DESIGN

Sullivan, J.W. and Tyler, S.W. (Editors) (1991) *Intelligent user interfaces.* ACM Press.

Wahlster	User and discourse models for multimodal communication

Thimbleby, H. (1990) *User interface design.* ACM Press / Addison-Wesley.

Chapter 9	Science
Chapter 10	Principles for principles
Chapter 13	A formal model for interactive systems

6.2 Dave Maulsby: Exam 1 (2 hours)

MACHINE LEARNING

Answer two of the following three questions.

1. With reference to Rivest and Schapire's new approach to unsupervised learning in deterministic environments, explain what is meant by
    a. a "test";
    b. "equivalence" of tests;
    c. a "canonical" test;
    d. a "square" test.

    What is the minimum information, in terms of tests, that is needed to simulate an environment?

    Discuss the role that this algorithm might play in eliciting procedures from untrained users of a computer system.

2. Define in the form of pseudo-code the COBWEB algorithm for incremental concept formation. Include a brief description of the form of the input and the result.

    Identify opportunities in a constraint-oriented procedural programming-by-example system for the application of such an incremental concept formation mechanism.

3. What is "design by derivational analogy"? Outline briefly its main features. Explain the following dimensions for the evaluation of replay mechanisms:
    - scope
    - evolvability
    - quality
    - efficiency
    - autonomy.

    Relate these to the evaluation of systems for end-user programming by example.

6.3 Dave Maulsby: Exam 2 (2 hours)

CONVERSATION ANALYSIS, CONTENT ANALYSIS, RESEARCH METHODS

Answer Question 4 or Question 5.

4. How do repertory grid studies differ from classical experimental designs?

   How could the repertory grid procedure be used to study users' attitudes toward a particular man-machine interface (say the CIMA system of your thesis proposal)?

5. How could content analysis be used to test the hypothesis that users respond differently to computers with male and female voices? Discuss how your test of this hypothesis would be organized. Include, where possible, information about subjects, task, independent and dependent variables, and research design.

USER INTERFACE DESIGN

Answer Question 6 or Question 7.

6. What are Popper's criteria for a "good scientific idea"?

   If possible, choose one "good idea" from your proposed research and discuss the extent to which it fulfils each of these criteria. If this is not possible, explain why.

7. Define, with examples, Thimbleby's notion of "gueps"—generative user engineering principles.

   Invent, and discuss, two new gueps that apply to instructable computer systems.

6.4 Dave Maulsby: Take-home exam (72 hours)

This question asks you to explore the process of generalizing data descriptions that is central to your proposed research. According to your proposal, the process involves

a associating selected objects with one another as a class;

b composing descriptions from relevant relations that are instantiated in or inferred from the instances of the class;

c finding a logical form that optimizes coverage of positive versus negative instances.

1. Define a data description language that is suitable for illustrating this process (you are encouraged to adopt, or if necessary adapt, the language used in an existing system such as METAMOUSE).
2. For each of steps (a) and (b) choose one existing technique, already described in the literature, that can be brought to bear on the problem. Explain the technique and how it can be used. Be sure to ground your discussion in specific examples of generalization problems using the data description language you have defined.
3. Consider in detail the application to step (c) of particular techniques, already described in the literature, of *both* empirical learning *and* case-based reasoning, again grounding your discussion in specific examples. Identify the strengths and weaknesses of each approach.