HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

# Research Methodology

## Review P2:  Good/bad abstracts

Review  7. September 2007


# Objective of project

- Read a number of abstracts
- Decide what works and what does not

- Hopefully, start thinking about how own abstracts could be good rather than bad


# Abstract 1

**Scannerless NSLR(1) parsing of programming languages**

The disadvantages of traditional two-phase parsing (a scanner phase preprocessing input for a parser phase) are discussed. We present metalanguage enhancements for context-free grammars that allow the syntax of programming languages to be completely described in a single grammar. The enhancements consist of two new grammar rules, the exclusion rule, and the adjacency-restriction rule. We also present parser construction techniques for building parsers from these enhanced grammars, that eliminate the need for a scanner phase.


# Abstract 2

**Parsing expression grammars:**
**a recognition-based syntactic foundation**
For decades we have been using Chomsky's generative system of grammars, particularly context-free grammars (CFGs) and regular expressions (REs), to express the syntax of programming languages and protocols. The power of generative grammars to express ambiguity is crucial to their original purpose of modelling natural languages, but this very power makes it unnecessarily difficult both to express and to parse machine-oriented languages using CFGs. Parsing Expression Grammars (PEGs) provide an alternative, recognition-based formal foundation for describing machineoriented syntax, which solves the ambiguity problem by not introducing ambiguity in the first place. Where CFGs express nondeterministic choice between alternatives, PEGs instead use prioritized choice. PEGs address frequently felt expressiveness limitations of CFGs and REs, simplifying syntax definitions and making it unnecessary to separate their lexical and hierarchical components. A linear-time parser can be built for any PEG, avoiding both the complexity and fickleness of LR parsers and the inefficiency of generalized CFG parsing. While PEGs provide a rich set of operators for constructing grammars, they are reducible to two minimal recognition schemas developed around 1970, TS/TDPL and gTS/GTDPL, which are here proven equivalent in effective recognition power.


# Abstract 3

**UNIX on a Loosely Coupled Architecture:  The Chorus/Mix Approach**
Paper presented at the EIT Workshop on Parallel and Distributed Worstation Systems, September 26-27, 1991, Florence, Italy.


# Abstract 4

**PARMON: A Comprehensive Cluster Monitoring System**

Workstation clusters have off late become a cost-effective solution for high performance computing. C-DAC's PARAM OpenFrame is a large cluster of high performance workstations interconnected through low-latency, high bandwidth communication networks. Monitoring such huge systems is a tedious and challenging task since typical workstations are designed to work as a standalone system, rather than a part of workstation clusters. System administrators require tools to effectively monitor such huge systems. PARMON provides the solution to this challenging problem.

PARMON is a portable, flexible, interactive, scalable, location-transparent, and comprehensive environment for monitoring of large clusters. It follows client-server methodology and provides transparent access to all nodes to be monitored from a monitoring machine. PARMON allows to monitor critical system resources activities and their utilization at three different levels: entire system, node, and component level. It allows monitoring multiple instances of the same component such as CPU in SMP node.

The two major components of PARMON are ***parmon-server—system resource activities*** and utilization information provider and ***parmon-client—GUI based client responsible for*** interacting with parmon-server and users for data gathering in real-time and presenting information graphically for visualization. The PARMON Client is designed, developed, and implemented using the state-of-the-art object-oriented, client-server, and Java computing technologies. The PARMON-server is developed as a multithreaded server using POSIX/Solaris threads and C as Java does not support interfaces to access system internals. PARMON is being successfully used in monitoring PARAM OpenFrame Supercomputer, which is a cluster of 48 Ultra-4 workstations running SUN-Solaris operating system.


# Abstract 5

We describe Streamer, the query-reformulation component of a data integration system. Given a utility measure and a user query, Streamer uses abstraction-based refinement planning and exploits information on plan independence to produce, in decreasing order of utility, a set of plans that access data sources to obtain answers to the query. We then focus on plan coverage as an important utility measure. We show how to use statistic information about the domain and data sources to estimate plan coverage, and how to incorporate the plancoverage framework into Streamer. In doing so, we provide the first method for effectively integrating the use of quantitative information into the query optimizer of a data-integration system. We present preliminary experimental results suggesting that Streamer runs an order of magnitude faster than brute-force plan-ordering methods, which are the only currently available methods to compute exact plan orderings. Finally, we propose methods to make Streamer...


# Abstract 6

A central problem in multistrategy learning systems is the selection and sequencing of machine learning algorithms for particular situations. This is typically done by the system designer who analyzes the learning task and implements the appropriate algorithm or sequence of algorithms for that task. We propose a solution to this problem which enables an AI system with a library of machine learning algorithms to select and sequence appropriate algorithms autonomously. Furthermore, instead of relying on the system designer or user to provide a learning goal or target concept to the learning system, our method enables the system to determine its learning goals based on analysis of its successes and failures at the performance task. The method involves three steps: Given a performance failure, the learner examines a trace of its reasoning prior to the failure to diagnose what went wrong (blame assignment); given the resultant explanation of the reasoning failure, the learner posts explicit...


# Abstract 7

**Abstract**

This empirical research was undertaken as part of a multi-method programme of research to investigate unsupported claims made of object-oriented technology. A series of subject-based laboratory experiments, including an internal replication, tested the effect of inheritance depth on the maintainability of object-oriented software. Subjects were timed performing identical maintenance tasks on object-oriented software with a hierarchy of three levels of inheritance depth and equivalent object-based software with no inheritance. This was then replicated with more experienced subjects. In a second experiment of similar design, subjects were timed performing identical maintenance tasks on object-oriented software with a hierarchy of five levels of inheritance depth and the equivalent object-based software.

The collected data showed that subjects maintaining object-oriented software with three levels of inheritance depth performed the maintenance tasks significantly quicker than those maintaining equivalent object-based software with no inheritance. In contrast, subjects maintaining the object-oriented software with five levels of inheritance depth took longer, on average, than the subjects maintaining the equivalent object-based software (although statistical significance was not obtained). Subjects' source code solutions and debriefing questionnaires provided some evidence suggesting subjects began to experience difficulties with the deeper inheritance hierarchy

It is not at all obvious that object-oriented software is going to be more maintainable in the long run. These findings are sufficiently important that attempts to verify the results should be made by independent researchers.


# Abstract 8

This paper introduces EasyAccept, a tool to create and run client-readable acceptance tests easily, and describes how it can be used to allow a simple but powerful acceptancetest driven software development (ATDD) approach. EasyAccept takes acceptance tests enclosing business rules and a Façade to access the software under development, and checks if the outputs of the software's execution match expected results from the tests. Driven by EasyAccept runs, software can be constructed with focus, control and correctness, since the acceptance tests also serve as automated regression tests. Results of experiments with undergraduate students demonstrate the benefits of the ATDD approach using EasyAccept and show that this tool can also help to teach and train good testing and development practices


# Abstract 9

## ABSTRACT

We propose a novel approach to performing efficient similarity search and classification in high dimensional data. In this framework, the database elements are vectors in a Euclidean space. Given a query vector in the same space, the goal is to find elements of the database that are similar to the query. In our approach, a small number of independent "voters" rank the database elements based on similarity to the query. These rankings are then combined by a highly efficient aggregation algorithm. Our methodology leads both to techniques for computing approximate nearest neighbors and to a conceptually rich alternative to nearest neighbors.

One instantiation of our methodology is as follows. Each voter projects all the vectors (database elements and the query) on a random line (different for each voter), and ranks the database elements based on the proximity of the projections to the projection of the query. The aggregation rule picks the database element that has the best median rank. This combination has several appealing features. On the theoretical side, we prove that with high probability, it produces a result that is a $(1+\epsilon)$-factor approximation to the Euclidean nearest neighbor. On the practical side, it turns out to be extremely efficient, often exploring no more than 5% of the data to obtain very high-quality results. This method is also database-friendly, in that it accesses data primarily in a pre-defined order without random accesses, and, unlike other methods for approximate nearest neighbors, requires almost no extra storage. Also, we extend our approach to deal with the $k$ nearest neighbors.

# Abstract 10

**Abstract**

The R-tree, one of the most popular access methods for rectangles, is based on the heuristic optimization of the area of the enclosing rectangle in each inner node By running numerous experiments in a standardized testbed under highly varying data, queries and operations, we were able to design the R*-tree which incorporates a combined optimization of area, margin and overlap of each enclosing rectangle in the directory Using our standardized testbed in an exhaustive performance comparison, it turned out that the R*-tree clearly outperforms the existing R-tree variants Guttman's linear and quadratic R-tree and Greene's variant of the R-tree This superiority of the R*-tree holds for different types of queries and operations, such as map overlay, for both rectangles and multidimensional points in all experiments From a practical point of view the R*-tree is very attractive because of the following two reasons 1 it efficiently supports point and spatial data at the same time and 2 its implementation cost is only slightly higher than that of other R-trees


# Abstract 11

**Automatically Tuned**
**Linear Algebra Software.**

This paper describes an approach for the autmatic generation and optimization of numerical software for processors with deep memory hierarchies and pipelined functional units. The production of such software for machines ranging from desktop workstations to embedded processors can be a tedious and time consuming process. The work described here can help in automating much of this process. We will concentrate our efforts on the widely used linear algebra kernels called the Basic Lienar Algebra Subroutines (BLAS).
In particular, the work presented here is for general matrix multiply, DGEMMM. However much of the technology and approach developed here can be applied to the other Level 3 BLAS and the general strategy can have an impact on basic linear algebra operations in general and amay be extended to other important kernel operations1.


# Abstract 12

We evaluate pathchar, a tool that infers the charasteristics of links along an Internet path (latency, bandwidth, queue delays). Looking at two example paths, we identify cirumstances where pathchar is likely to succeed, and develop techniques to improve the accuracy of pathcar's estimates and reduce the time it takes to generate them. The most successful of thesse techniques is a form of adaptive data collection that reduces the number of measurments pathchar nedds by more than 90% for some links