REGISTRATION FORM

A movable feast: a technical retrospective on Paris Kanellakis

May 3, 1996

Name: ______________________

Title: ______________________

Company: ______________________

Division: ______________________

Address: ______________________

____

____

____

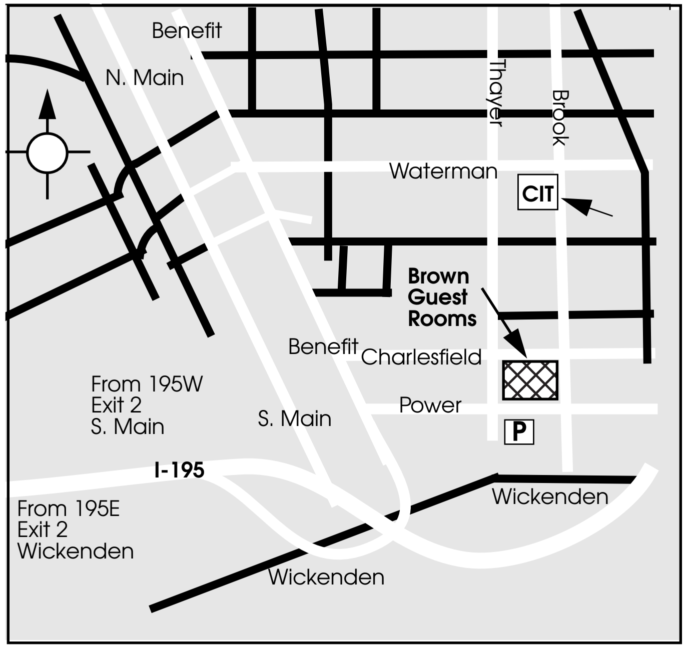
Phone: ______________________

Fax: ______________________

Email: ______________________

❋ ❋ ❋ ❋ ❋ ❋ ❋ ❋ ❋ ❋ ❋

Rooms have been reserved in the Brown Guest Facility, corner of Thayer & Charlesfield, for May 2nd only ($80/night). Please make reservations ***by April 19*** by calling Drea Harms at 401-863-7500 (fax 401-863-7300). Participants are responsible for their own lodging expenses. When calling for reservations, please refer to the Computer Science Department's Industrial Partners Program. We look forward to seeing you soon.



DIRECTIONS TO THE CIT BUILDING

- From I-95 N or S, take Exit 20 to I-195E
- From I-195E take Exit 2, Wickenden St.
- Go LEFT on Wickenden, LEFT again at the 2nd light onto Brook St.
- The red-brick CIT Building (Center for Information Technology) is on the left at the intersection of Brook and Waterman (1st light).

PARKING

For those staying at the Brown Guest Rooms parking is provided. Parking is also available on the second level of the parking garage on the corner of Thayer and Power. Please give your license plate number to our receptionist on the 4th floor upon arrival.

Please return the registration form or email same information ***by Tuesday, April 23,***
to Suzi Howe
Department of Computer Science
Brown University, Box 1910
Providence, RI 02912-1910
Phone: 401-863-7610
Fax: 401-863-7657
Email: sjh@cs.brown.edu

The 17th IPP Symposium

A movable feast: a technical retrospective on Paris Kanellakis

1953-1995

Friday, May 3, 1996
Hosts: Pascal Van Hentenryck
Stanley Zdonik

INDUSTRIAL PARTNERS PROGRAM

Department of Computer Science

CIT Building
Brown University

17th IPP Symposium
A movable feast: a technical retrospective on Paris Kanellakis (1953–1995)

A movable feast is Ernest Hemingway's memoir of life in Paris at the beginning of the century. He wrote that Paris was a 'movable feast' because, having enjoyed and been stimulated by what it has to offer, one takes it with one wherever one goes forever after.

A Formal Model for Object-Oriented Databases
Serge Abiteboul, Stanford University

Object-oriented databases combine (relational) database languages and object-oriented programming to provide more flexible systems that can dramatically improve application programmers' productivity. The field emerged in the late '80s from research venues and product development in startups. However, the new technology lacked the formal foundations of the relational model. We will survey some of Paris Kanellakis's work that contributed to bridging this gap. In particular, we discuss a formal object-based database model that he studied. The structural part generalizes the relational model and a number of extensions. The main contribution is the operational part of the data model, the query language IQL, which uses object identity for many critical purposes (e.g., representation of cycles and sharing).

Global Optimization Problems for Database Logic Programs
Moshe Vardi, Rice University

Datalog is the language of logic programs without function symbols. It is used as a database query language and can be viewed as an abstraction of general logic programs. The additional expressive power of Datalog, which is due to its ability to express recursion, comes, however, at the price of increased computational complexity. Thus, the most fundamental optimization problem for Datalog programs is that of eliminating recursion. If it is possible to eliminate recursion from a Datalog program P, then P is said to be bounded. The boundedness problem is to decide whether or not a given Datalog program is bounded. This is a typical global optimization problem in which one tries to optimize the program globally rather than locally. I will provide an overview of a decade of research into the boundedness problem in which Paris Kanellakis played a significant role. On one hand, I will show that boundedness is undecidable, and use this result to show that 'almost' all global optimization problems are undecidable. On the other, I will describe some interesting special cases in which boundedness is decidable. Underlying the positive results are new tools for the optimization of Datalog programs based on automata theory and logic.

Efficient Parallel Computation in the Presence of Failures and Delays
Alex Shvartsman, MIT

Our goal is to combine the speed-up potential of parallel computation with the reliability potential that comes with replicated processors in distributed computation, an area in which Paris Kanellakis made significant contributions. Combining efficiency and fault-tolerance involves an inherent tradeoff: efficiency is achieved by eliminating redundancy, while fault-tolerance is achieved by introducing redundancy. We present algorithmic techniques for reconciling this tradeoff in the presence of processor failures and delays. We show that for certain models of computation it is possible to translate automatically programs that assume fault-free synchronous parallel machine to programs that can be executed correctly and efficiently (or even optimally) on failure-prone or asynchronous multiprocessors. For the archetypal 'Write-All' problem—P processors initializing N memory cells—we overview the state of the art in algorithmics, load-balancing techniques and lower bounds.

Complexity-Theoretic Aspects of Programming Language Design
Harry Mairson, Brandeis University

We survey three of Paris Kanellakis's contributions to the complexity-theoretic analysis of constructs in the design of programming languages. These are (1) his result that first-order unification is *complete for polynomial time*; (2) his contributions to the complexity analysis of type inference for polymorphically-typed functional programming languages, which was proven to be *complete for exponential time*; and (3) his work on *expressibility of simply typed lambda calculus* when used as a functional database programming language.

These research investigations are interrelated, emphasizing common themes and difficulties in the design of programming languages with concrete implications that can be appreciated by language designers. First-order unification is a ubiquitous building block in implementations of sophisticated programming languages. It is, for example, the workhorse of logic programming engines and the essential component of compile-time type analysis. The research on unification provided a secure foundation for understanding the complexity of automatic polymorphic type inference in functional languages such as ML and Haskell.

Modern functional languages are based on the typed lambda calculus, so it is then natural to consider its computational expressiveness. We describe how the degree of higher-order functionality in simply typed terms can be related directly to well known complexity classes.

Constraint Databases
Gabriel Kuper, INRIA, Rocquencourt

Constraint databases were first introduced in a paper by Kanellakis, Kuper and Revesz in 1990 as a framework for combining concepts from constraint programming and databases, and have since become a very active area of database research. The basic idea is that the notion of a tuple in a relational database is replaced by a conjunction of constraints from an appropriate language; for example, order constraints or linear arithmetic constraints. Such a tuple can can be seen as representing a large, possibly even infinite, set of points in a compact way (e.g., for spatial databases and GIS).

We describe the general framework for constructing constraint databases and query languages for them. We describe query languages and complexity results for several specific constraint classes. Besides briefly surveying other developments in the field (aggregation, applications in GIS, etc.), we discuss Paris's more recent work, including work on indexing, constraint query algebras and expressive power.

SCHEDULE

8:30 Registration and Breakfast
4th Floor, CIT Building

9:15 Introductions and Overview
Eugene Charniak, Chairman
Pascal Van Hentenryck, Co-Host
Stanley Zdonik, Co-Host

9:45 ***A Formal Model for Object-Oriented Databases***
Serge Abiteboul, Stanford University

10:45 **BREAK**

11:00 ***Global Optimization Problems for Database Logic Programs***
Moshe Vardi, Rice University

12:00 **BUFFET LUNCH**

1:30 ***Efficient Parallel Computation in the Presence of Failures and Delays***
Alex Shvartsman, MIT

2:30 **BREAK**

2:45 ***Complexity-Theoretic Aspects of Programming Language Design***
Harry Mairson, Brandeis University

3:45 ***Constraint Databases***
Gabriel Kuper, INRIA, Rocquencourt

4:45 **PANEL DISCUSSION**

5:15 **RECEPTION (5th Floor)**