**CV template for the employment and promotion of teachers**

**Established by the Employment Committee 13 December 2016, minutes no. 20/2016**

In order to facilitate the processing of applications, KTH uses a CV template which specifies the data required in the process. Different points in the template hold more relevance depending upon the position applied for. The following instructions apply to the applicant:

1. The CV template should be used with the retained numbering.
2. Applications should be written in English.
3. Applications should be addressed to the President (a signature is not necessary).
4. Apply online via the KTH recruitment system.
5. CV and the attachments are to be uploaded in one file.
6. Publications are to be uploaded separately.
7. Your application must be received no later than the date stated in the advertisement.
8. KTH conducts reference checks on the candidate proposed for the position.

|  |
| --- |
| The application of employment as Professor in Data Science (VL-2016-0052). |

|  |  |
| --- | --- |
| 1. Basic information | |
| 1.1. | Vijay Anand Saraswat |
| 1.2. | 30 September 1960 |
| 1.3. | Male |
| 1.4. | 30 Scout Hill Road, Mahopac, NY 10541 |
| 1.5. | IBM TJ Watson Research Centre, 1101 Kitchawan Road, Yorktown Heights, NY 10598; 914 945 3307; vsaraswa@us.ibm.com |
| 1.6. | Chief Scientist, Deep Compliance, Cognitive Computing Research, IBM TJ Watson Research Center.  Area: Cognitive Computing / AI – Knowledge Representation and Reasoning, Natural Language Understanding. Started at IBM September 2, 2003. |
| 1.7. | * May 2016 – onwards: Chief Scientist, Deep Compliance, Cognitive Computing Research at IBM TJ Watson, focused on developing a framework for strategic research in the compliance / legal / financial spaces. * January 2015 – May 2016: Distinguished Research Staff Member, reporting to VP, Cognitive Computing Research at IBM TJ Watson. * February 2014 -- January 2015: Chief Scientist, Computation as a Service Research Division, IBM TJ Watson. * December 2012 – May 2016: Chief Scientist, IBM Continuous Insight (business group); concurrent with Research appointment. * 2009 – 2013: Adjunct Professor, Columbia University, NY. Taught a course on X10 for five years with Prof Martha Kim. * 2009 – 2010: Waseda University, Global COE Visiting Professor. * February 2008 -- February 2014: Manager, Advanced Programming Languages, IBM TJ Watson. * September 2003 -- onwards: Research Staff Member, IBM TJ Watson. * August 2002 -- Sep 2003: Professor of Computer Science and Engineering, Penn State University. Area: Programming, Languages and Systems. * June 2001 -- August 2002: Vice President, Engineering, Kirusa. Responsible for recruiting the engineering team, developing product plans, architecting the product, and delivering products. * December 2000 -- June 2001: Chief Technology Officer, Vayusphere. Responsible for recruiting engineering team in San Diego and Mountain View. Responsible for defining, architecting and shipping first set of products (Monsoon Real-time Messaging Platform), setting the technical direction for the company, evaluating vendor technology, and partnering with business development. * July 2000 -- December 2000: Director, Development, Vayusphere. Employee #6 at Vayusphere. Designed, directed and architected the implementation of Vayusphere Instant Messaging. * November 99 -- July 2000: Technology Leader, Instant Messaging and Presence, AT&T Shannon Labs. Secured commitment from AT&T Wireless to trial AT&T Instant Messaging for their Pocket Net service. Grew the Labs Instant Messaging team. At its peak, the work involved over 40 people. * November 98 -- November 99: District Manager, Network Community Platform Group, AT&T Shannon Labs. Led the design, architecture and development of AT&T Instant Messaging, based on Matrix. Formed the team from scratch. * September 96 -- November 98: Principal Member, Technical Staff, AT&T Shannon Labs. Developed and implemented the underlying ideas for Matrix, an architecture for extensible network spaces based on Java. * November 87 -- September 96: Member, Research Staff, Xerox Palo Alto Research Centre. Developed the notion of concurrent constraint programming languages. Initiated and led the thrusts at PARC in model-based computing, hybrid computing and network communities. * April 86 -- November 87: Artificial Intelligence Scientist, Advanced Product Developments Group, Carnegie Group Inc. Helped initiate the ``Carnegie Inference Language'' project, to develop the next generation of expert-system development tools. * January 85 -- March 86: Consultant, Carnegie Group Inc., Pittsburgh. Designed and implemented CRL-Prolog. |
| 1.8. | *Other.* |
|  | |
| 2. *Higher education degrees and evaluations* | |
| 2.1. | PhD Computer Science, 1989 Carnegie-Mellon University.  MS Computer Science, 1985 Carnegie-Mellon University.  B Tech Electrical Engineering, 1982, IIT Kanpur. |
| 2.2. | *Appointment as docent.* N/A (I studied in the US University system.) |
| 2.3. | *Evaluation of own scientific field (research council etc.)* |
| 2.4. | *You may attach previous expert opinions (for the last five years).*  I do not have any expert opinions to attach. However, the following colleagues have kindly agreed to write letters of reference for me, so they may be contacted for expert opinions on my work:   1. Prof Doug Lea, SUNY Oswego, ACM Fellow, [dl@cs.oswego.edu](mailto:dl@cs.oswego.edu). 2. Dr David Grove, ACM Fellow, Distinguished Research Staff Member, IBM TJ Watson, [groved@us.ibm.com](mailto:groved@us.ibm.com) 3. Prof Seif Haridi, Chief Scientist SICS and Professor KTH, [seif@sics.se](mailto:seif@sics.se) 4. Prof Radha Jagadeesan, De Paul University, [RJagadeesan@cs.depaul.edu](mailto:RJagadeesan@cs.depaul.edu) 5. Prof Prakash Panangaden, U Montreal, Fellow, Royal Society of Canada, [prakash@cs.mcgill.ca](mailto:prakash@cs.mcgill.ca) 6. Prof Catuscia Palamidessi, LiX Polytechnique, Director of Research, [catuscia@lix.polytechnique.fr](mailto:catuscia@lix.polytechnique.fr) 7. Dr Johan de Kleer, PARC, AAAI Fellow, [dekleer@parc.com](mailto:dekleer@parc.com) 8. Dr Fernando Pereira, Distinguished Scientist, Google, ACM Fellow, AAAI Fellow, [Pereira@google.com](mailto:Pereira@google.com) 9. Prof Mary Dalrymple, Professor of Syntax, Oxford University, Fellow of British Academy, mary.dalrymple@ling-phil.ox.ac.uk |
|  |  |
| 3. Research merits | |
| 3.1. | *Describe your research profile (maximum 1 page).* |
| 3.2. | *Describe your planned research activities (maximum 2 pages).*   * *Planned research activities.* * *Collaboration with the society and the business world.* |
| 3.3. | *List your publications (in a numbered list).*  **Journal articles**: *Below we include the abstract of the journal paper as a way of describing the paper.*   1. Tardieu, B Herta, D Cunningham, D Grove, P Kambadur, V Saraswat, A Shinar, M Takeuchi, M Vaziri, W Zhang “X10 and APGAS at Petascale”, *ACM Transactions on Parallel Computing*, March 2016.   Abstract: X10 is a high-performance, high-productivity programming language aimed at large-scale distributed and shared-memory parallel applications. It is based on the Asynchronous Partitioned Global Address Space (APGAS) programming model, supporting the same fine-grained concurrency mechanisms within and across shared-memory nodes. We demonstrate that X10 delivers solid performance at petascale by running (weak scaling) eight application kernels on an IBM Power 775 supercomputer utilizing up to 55,680 Power7 cores (for 1.7Pflop/s of theoretical peak performance). For the four HPC Class 2 Challenge benchmarks, X10 achieves 41% to 87% of the system’s potential at scale (as measured by IBM’s HPCC Class 1 optimized runs). We also implement K- Means, Smith-Waterman, Betweenness Centrality, and Unbalanced Tree Search (UTS) for geometric trees. Our UTS implementation is the first to scale to petaflop systems. We describe the advances in distributed termination detection, distributed load balancing, and use of high-performance interconnects that enable X10 to scale out to tens of thousands of cores. We discuss how this work is driving the evolution of the X10 language, core class libraries, and runtime systems.   1. U Grandi, A Loreggia, F Rossi and V Saraswat. “A Borda Count for Collective Sentiment Analysis” *Annals of Mathematics and Artificial Intelligence*, special issue on “Preferences and Computational Social Choice”, 2015.   Abstract: Sentiment analysis assigns a positive, negative or neutral polarity to an item or entity, extracting and aggregating individual opinions from their textual expressions by means of natural language processing tools. In this paper we observe that current sentiment analysis techniques are satisfactory in case there is a single entity under consideration, but can lead to inaccurate or wrong results when dealing with a set of multiple items. We argue in favor of importing techniques from voting theory and preference aggregation to provide a more accurate definition of the collective sentiment over a set of multiple items. We propose a notion of Borda count which combines individuals’ sentiment with comparative preference information, we show that this class of rules satisfies a number of properties which have a natural interpretation in the sentiment analysis domain, and we evaluate its behavior when faced with highly incomplete domains.   1. M Fromherz, V Saraswat and D Bobrow “Model-based computing: Developing flexible machine control software”, *Artificial Intelligence*, 114(1-2): 157-202 (1999)   Abstract: In the conventional approach to simulating, controlling, and diagnosing a real-world physical system, engineers typically analyze the interactions of the system's components and processes, and then develop new and dedicated code for that system. Instead, building on principles from model-based reasoning and constraint programming research, we propose an integrated approach to software development we call *model-based computing*. We present this approach in the context of control software for modular electro-mechanical systems. Our approach is used in commercial systems and has been shown to both simplify the development of machine control software, and make the software and the controlled systems more flexible and effective. In this paper, building on a generic control software architecture, we first develop a domain theory with corresponding modeling language. Models capture a system's capabilities from first principles and independently of the control task. We then introduce modeling technology using concurrent constraint programming, which gives our modeling approach a sound and powerful theoretical foundation. Constraint programming also brings with it a host of generic reasoning techniques such as deduction, abduction, and search, and we show how such techniques can be applied to the model-based configuration and control of our systems. We end with a review of how model-based computing can be extended to other tasks such as design and testing. We believe that together, models, task architecture, and reasoners offer a compelling framework for building software for computationally controlled systems.   1. V Gupta, R Jagadeesan and V Saraswat “Computing with Continuous Change”, *Science of Computer Programming*, 30 (1:2) 3-49, 1998.   Abstract: A central challenge in computer science and knowledge representation is the integration of conceptual frameworks for continuous and discrete change, as exemplified by the theory of differential equations and real analysis on the one hand, and the theory of programming languages on the other. We take the first steps towards such an integrated theory by presenting a recipe for the construction of continuous programming languages — languages in which state dynamics can be described by differential equations. The basic idea is to start with an untimed language and extend it uniformly over dense (real) time. We present a concrete mathematical model and language (the Hybrid concurrent constraint programming model, Hybrid cc) instantiating these ideas. The language is intended to be used for modeling and programming *hybrid systems*. The language is declarative — programs can be understood as formulas that place constraints on the (temporal) evolution of the system, with parallel composition regarded as conjunction. It is expressive — it allows the definition of continuous versions of the preemption control constructs. The language is obtained by extending the general-purpose computational formalism of (default) concurrent constraint programming (Default cc) with a single temporal construct, called **hence** — **hence** *A* is read as asserting that *A* holds *continuously* beyond the current instant. Various patterns of temporal activity can be generated from this single construct by use of the other combinators in Default cc. We provide a precise operational semantics according to which execution alternates between (i) points at which discontinuous change can occur, and (ii) open intervals in which the state of the system changes continuously. Transitions from a state of continuous evolution are triggered when some condition starts or stops holding. We show that the denotational semantics is correct for reasoning about the operational semantics, through an adequacy theorem.   1. Eric Torng, Rajeev Motwani, and V Saraswat “Online scheduling with lookahead: Multipass assembly lines”, *INFORMS Journal on Computing*, 1998.   Abstract: This article describes our use of competitive analysis and the on-line model of computation in a product development setting; specifically, we use competitive analysis to evaluate on-line scheduling strategies for controlling a new generation of networked reprographic machines (combination printer-copier-fax machines servicing a network) currently being developed by companies such as Xerox Corporation. We construct an abstract machine model, the multipass assembly line, which not only models networked reprographic machines but also models several common manufacturing environments such as a robotic assembly line or a mixed product assembly line. We consider on-line algorithms with finite lookahead because these machines typically have limited knowledge of the future. We first prove some lower bounds on the performance of any online algorithm with finite lookahead. We then show that simple greedy algorithms achieve competitive ratios that are close to these general lower bounds. In particular, we show that lookahead improves the competitive ratio of these simple greedy algorithms from approximately 2 (with no lookahead) to being arbitrarily close to 1 (for large lookahead). This implies these simple greedy algorithms are realistic candidates for field use in future reprographic products.   1. Mary Dalrymple, John Lamping, F Pereira, and V Saraswat “Quantification, Anaphora and Intensionality”, In *Journal of Logic, Language and Information* 6 (3), pp 219-273, July 1997. Abstract: The relationship between Lexical-Functional Grammar (LFG) functional structures (f-structures) for sentences and their semantic interpretations can be formalized in linear logic in a way that correctly explains the observed interactions between quantifier scope ambiguity, bound anaphora and intensionality.   Our linear-logic formalization of the compositional properties of quantifying expressions in natural language obviates the need for special mechanisms, such as Cooper storage, in representing the scoping possibilities of quantifying expressions. Instead, the semantic contribution of a quantifier is recorded as a linear-logic formula whose use in a proof will establish the scope of the quantifier. Different proofs can lead to different scopes. In each complete proof, the properties of linear logic ensure that quantifiers are properly scoped.  The interactions between quantified NPs and intensional verbs such as ’’seek‘‘ are also accounted for in this deductive setting. A single specification in linear logic of the argument requirements of intensional verbs is sufficient to derive the correct reading predictions for intensional-verb clauses both with nonquantified and with quantified direct objects. In particular, both de dicto and de re readings are derived for quantified objects. The effects of type-raising or quantifying-in rules in other frameworks just follow here as linear-logic theorems.  While our approach resembles current categorial approaches in important ways (Moortgat, 1988, 1992a; Carpenter, 1993; Morrill, 1994) it differs from them in allowing the greater compositional flexibility of categorial semantics (van Benthem, 1991) while maintaining a precise connection to syntax. As a result, we are able to provide derivations for certain readings of sentences with intensional verbs and complex direct objects whose derivation in purely categorial accounts of the syntax-semantics interface appears to require otherwise unnecessary semantic decompositions of lexical entries.   1. V Saraswat ``Compositional Computing’’, CONSTRAINTS 2(1):95-97 (1997)   Abstract: Compositionlity is a critical design idea for the design of complex computational systems. We argue that concurrent constraint programming provides powerful techniques for compositional construction, allowing modularity – hence separation and partitioning – and yet openness and interdependence, via constraint-based communication.   1. V Gupta, R Jagadeesan and V Saraswat “Truly Concurrent Constraint Programming”, Theoretical Computer Science, Volume 278, pp 223-255, 2002. Conference version in Proceedings of the International Conference on Concurrency Theory, LNCS 1119 August 1996.   Abstract: We study “causality” relationships in Concurrent Constraint Programming: what is observed is not just the conjunction of constraints deposited in the store, but also the causal dependencies between these constraints. We describe a denotational semantics for cc that is fully abstract with respect to observing this “causality” relation on constraints. This semantics preserves more fine-grained structure of computation; in particular the Interleaving Law (a→P)∥(b→Q)=(a→(P∥(b→Q))) □ (b→(Q∥(a→P))) is not verified (□ is indeterminate choice). Relationships between such a denotational approach to true concurrency and different powerdomain constructions are explored.   1. V Saraswat, R Jagadeesan and V Gupta “Timed Default Concurrent Constraint Programming”, In *Journal of Symbolic Computation* 22 (5,6) 475--520, 1996. Extended abstract published in the Proceedings of the ACM Symposium on Principles of Programming Languages, San Francisco, 1995.   Abstract. Synchronous programming (Berry, 1989) is a powerful approach to programming reactive systems. Following the idea that “processes are relations extended over time” (Abramsky, 1993), we propose a simple but powerful model for timed, determinate computation, extending the closure-operator model for untimed concurrent constraint programming (CCP). In Saraswat *et al*. (1994a) we had proposed a model for this called tcc—here we extend the model of tcc to express strong time-outs: if an event *A* does not happen through time *t*, cause event *B* to happen at time *t*. Such constructs arise naturally in practice (e.g. in modeling transistors) and are supported in synchronous programming languages. The fundamental conceptual difficulty posed by these operations is that they are non-monotonic. We provide compositional semantics to the non-monotonic version of concurrent constraint programming (Default cc) obtained by changing the underlying logic from intuitionistic logic to Reiter's default logic. This allows us to use the same construction (uniform extension through time) to develop Default cc as we had used to develop tcc from cc. Indeed the smooth embedding of cc processes into Default cc processes lifts to a smooth embedding of tcc processes into Default cc processes. We identify a basic set of combinators (that constitute the Default cc programming framework), and provide constructive operational semantics (implemented by us as an interpreter) for which the model is fully abstract. We show that the model is expressive by defining combinators from the synchronous languages. We show that Default cc is compositional and supports the properties of multiform time, orthogonal pre-emption and executable specifications. In addition, Default cc programs can be read as logical formulae (in an intuitionistic temporal logic)—we show that this logic is sound and complete for reasoning about (in)equivalence of Default cc programs. Like the synchronous languages, Default cc programs can be compiled into finite state automata. In addition, the translation can be specified compositionally. This enables separate compilation of Default cc programs and run-time tradeoffs between partial compilation and interpretation. A preliminary version of this paper was published as Saraswat *et al*. (1995). Here we present a complete treatment of hiding, along with a detailed treatment of the model.   1. M Dalrymple, John Lamping, F Pereira, and V Saraswat “Intensional Verbs Without Type-Raising or Lexical Ambiguity”, In *Logic, Language and Computation*, volume 1, ed. Jerry Seligman and Dag Westerstaahl. Stanford, California: Center for the Study of Language and Information. 1996. Also in Proceedings of the Conference on Information-Oriented Approaches to Logic, Language and Computation/Fourth Conference on Situation Theory and its Applications, Saint Mary's College of California, Moraga, California. June 1994.   Abstract. We present an analysis of the semantic interpretation of intensional verbs such as seek that allows them to take direct objects of either individual or quantifier type, producing both *de dicto* and *de re* readings in the quantifier case, all without needing to stipulate type-raising or quantifying-in rules. This simple account follows directly from our use of logical deduction in linear logic to ex- press the relationship between syntactic structures and meanings. While our analysis resembles current categorial approaches in important ways ((Moortgat , 1988; Moortgat , 1992a; Morrill , 1993; Carpenter , 1993)), it differs from them in allowing the greater type flexibility of categorial semantics ((van Benthem , 1991)) while maintaining a precise connection to syntax. As a result, we are able to provide derivations for certain readings of sentences with intensional verbs and complex direct objects that are not derivable in current purely categorial accounts of the syntax-semantics interface. The analysis forms a part of our ongoing work on semantic interpretation within the framework of Lexical-Functional Grammar.   1. Pascal van Hentenryck, Yves Deville, V Saraswat “Design, implementation and evaluation of the constraint language cc(FD)”, *Journal Of Logic Programming* 37(1-3):139-164 (1998). Conference paper in LNCS 910, pp 293-316 (1994).   Abstract: This paper describes the design, implementation, and applications of the constraint logic language cc(FD). cc(FD) is a declarative nondeterministic constraint logic language over finite domains based on the cc framework [33], an extension of the Constraint Logic Programming (CLP) scheme [21]. Its constraint solver includes (nonlinear) arithmetic constraints over natural numbers which are approximated using domain and interval consistency. The main novelty of cc(FD) is the inclusion of a number of general-purpose combinators, in particular cardinality, constructive disjunction, and blocking implication, in conjunction with new constraint operations such as constraint entailment and generalization. These combinators significantly improve the operational expressiveness, extensibility, and flexibility of CLP languages and allow issues such as the definition of nonprimitive constraints and disjunctions to be tackled at the language level. The implementation of cc(FD) (about 40,000 lines of C) includes a WAM-based engine [44], optimal arc-consistency algorithms based on AC-5 [40], and incremental implementation of the combinators. Results on numerous problems, including scheduling, resource allocation, sequencing, packing, and hamiltonian paths are reported and indicate that cc(FD) comes close to procedural languages on a number of combinatorial problems. In addition, a small cc(FD) program was able to find the optimal solution and prove optimality to a famous 10/10 disjunctive scheduling problem [29], which was left open for more than 20 years and finally solved in 1986.  **List of conference and workshop papers, technical reports**  **I AI, Logic, Knowledge Representation and Reasoning**   1. C Cornelio, V Saraswat “Expressing Probabilistic Graphical Models in RCC”, Proceedings of the **National Conference of American Association of Artificial Intelligence AAAI-17 Workshop** on Symbolic Inference and Optimization, 2017 2. A Loreggia, H Samulowitz, Y Malitsky, V Saraswat “Deep Learning for Algorithm Portfolios”, Proceedings of the **National Conference of American Association of Artificial Intelligence, AAAI-16, 2016**. 3. V Saraswat and J Milthorpe “The Continuous Allreduce algorithm for asynchronous stochastic gradient descent”, **NIPS 2015 Workshop** on Non-Convex Optimization for Machine Learning: Theory and Practice. 4. C Cornelio, A Loreggia, V Saraswat “Logical Conditional Preference Theories”, MPREF **Workshop**, **IJCAI 2015**. 5. U Grandi, A Loreggia, F Rossi and V Saraswat. “From Sentiment Analysis to Preference Aggregation”. In Proceedings of the 2014 International Symposium on Artificial Intelligence and Mathematics, **ISAIM-2014**, 2014. 6. R Jagadeesan and G Nadathur and V Saraswat “Testing concurrent systems: An interpretation of intuitionistic logic”, Proceedings of **FST&TCS 05,** 2005. 7. V Gupta, R Jagadeesan and V Saraswat “Probabilistic Concurrent Constraint Programming”, Proceedings of the International Conference on Concurrency Theory, LNCS, **CONCUR 97**, 243-257, 1997. 8. M Fromherz, V Gupta and V Saraswat, “CC --- A generic framework for domain specific languages”, Workshop on Domain-oriented specification languages, **POPL 97, 1997**. 9. M Fromherz and V Saraswat “Model-based computing: using concurrent constraint programming for modelling and model compilation”, U. Montanari and F. Rossi (ed.) Principles and Practices of Constraint Programming, **CP 95**, Springer-Verlag, LNCS 976, Sep 1995, pp. 629--635. 10. Y Iwasaki, A Farquhar, V Saraswat, D Bobrow and V Gupta “Modelling time in hybrid systems: How fast is `instantaneous’’?’’, Proceedings of the **International Joint Conference on Artificial Intelligence**, Montreal, **IJCAI-95**, **1995**. 11. H Wong and M Fromherz and V Gupta and V Saraswat. “Control-based programming of electro-mechanical controllers.’’ Proceedings of the **International Joint Conference on Artificial Intelligence Workshop** on Executable Temporal Logics, Montreal, August **1995**. 12. V Gupta and V Saraswat and P Struss, “A model of a photocopier paper path’’, Proceedings of the 2nd **IJCAI Workshop** on Engineering Problems for Qualitative Reasoning, August **1995**. 13. M Fromherz and V Saraswat “Model-based computing: constructing constraint-based software for electro-mechanical systems’’, **Practical Applications of Constraint Technology**, Paris, France, April **1995**, pp. 63-66. 14. M Fromherz, D Bell, D Bobrow, B Falkenhainer, V Saraswat and M Shirley “Rapper: The Copier Modelling Project’’, Working Papers of the Eight International **Workshop on Qualitative Reasoning** about physical systems'', pages 1-12, June **1994**. 15. O Raiman and J de Kleer and V Saraswat “Critical reasoning”, Proceedings of the **International Joint Conference on Artificial Intelligence, 1993**. 16. P Codognet and V Saraswat, “Abduction in Concurrent Constraint Languages”, Proceedings of the First Compulog Network meeting on Logic Programming and Artificial Intelligence, London, U.K., 1992. 17. O Raiman and J de Kleer and V Saraswat and Mark Shirley “Characterizing non-intermittent faults”, Proceedings of the National Conference on **Artificial Intelligence**, **AAAI-91,** June **1991**. 18. V Saraswat, J de Kleer and O Raiman “Contributions to the theory of diagnosis”, International **Workshop on Principles of Diagnosis**, Stanford University, July **1990**. 19. V Saraswat “CP as a general-purpose constraint-language”, Proceedings of the National Conference of **American Association of Artificial Intelligence, AAAI-87, 1987**.   **Natural Language Understanding**   1. M Dalrymple, F Pereira, John Lamping, V Saraswat “Introduction’’, In *Semantics And Syntax in Lexical Functional Grammar: The Resource Logic Approach,* edited by M Dalrymple. The MIT Press, 1999. 2. V Saraswat “LFG qua Concurrent Constraint Programming”, In *Semantics And Syntax in Lexical Functional Grammar: The Resource Logic Approach*, edited by M Dalrymple. The MIT Press, 1999. 3. Mary Dalrymple, V Gupta, John Lamping, and V Saraswat “Relating Resource-based semantics to categorial semantics”, In *Semantics And Syntax in Lexical Functional Grammar: The Resource Logic Approach*, edited by M Dalrymple. The MIT Press, 1999. 4. Mary Dalrymple, John Lamping, F Pereira, and V Saraswat “A Deductive Account of Quantification in LFG”, In *Quantifiers, Deduction, and Context*, ed. Makoto Kanazawa, Christopher J. Pinon, and Henriette de Swart. Stanford, California: Center for the Study of Language and Information. 1996. 5. Andrew Kehler, M Dalrymple, John Lamping, and V Saraswat “The Semantics of Resource Sharing in Lexical-Functional Grammar”, Proceedings of the 1995 Meeting of the European Chapter of the Association for Computational Linguistics, **EACL 95**, Dublin, Ireland. March 1995. 6. M Dalrymple, J Lamping, and V Saraswat. “LFG semantics via constraints”, In Proceedings of the Sixth Meeting of the European Association for Computing Linguistics, **EACL 93**, University of Utrecht, April 1993. 7. M Dalrymple, A Hinrichs, J Lamping, and V Saraswat “The resource logic of complex predicate interpretation”, In Proceedings of the 1993 Republic of China Computational Linguistics Conference (**ROCLING**), Hsitou National Park, Taiwan, September. Computational Linguistics Society of R.O.C.   **Parallel Constraint Solvers**   1. D Munera, D Diaz, S Abreu, F Rossi, V Saraswat, P Codognet “Solving Hard Stable Matching Problems via Local Search and Cooperative Parallelization”, 29th National Conference on Artificial Intelligence, **AAAI-15**, 2015. 2. D Bergman, A Cire, A Sabharwal, H Samulowitz, V Saraswat, W Jan van Hoeve “Parallel Combinatorial Optimization with Decision Diagrams”, **CPAIOR 2014**. 3. B Bloom, D Grove, B Herta, A Sabharwal, H Samulowitz, V Saraswat “SatX10: A Scalable Plug&Play Parallel SAT Framework”, in Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing, **SAT 2012**.     **II Programming Languages and Systems**  **The X10 Programming Language**  I started the X10 project in January 2004, after joining IBM TJ Watson, at the behest of Vivek Sarkar. I am the main designer of X10, though a team of outstanding contributors has contributed significantly to the design and implementation. The project is documented at x10-lang.org. It has a fairly large university community (over 35 universities using X10), with over 160 papers published. Courses have been taught based on X10 at Columbia University, UCLA, Australian National University, University of Kassel, [Georg-Simon-Ohm Hochschule Nürnberg](http://www.ohm-hochschule.de/), U Chicago, and Tianjin University.   1. S Crafa, D Cunningham, V Saraswat, Avraham Shinnar, O Tardieu “Semantics of (Resilient) X10”, Proceedings of European Conference on Object-oriented Programming, **ECOOP 2014**. 2. D Cunningham, D Grove, B Herta, Arun Iyengar, Kiyokuni Kawachiya, Hiroki Murata, V Saraswat, Mikio Takeuchi and O Tardieu “Resilient X10: efficient failure-aware programming”, Proceedings of Principles and Practice of Parallel Programming, **PPoPP 2014**. 3. Tardieu, B Herta, D Cunningham, D Grove, Prabhanjan Kambadur, V Saraswat, Avraham Shinnar, Mikio Takeuchi, Mandana Vaziri “APGAS at Peta-scale”, Proceedings of Principles and Practice of Parallel Programming, **PPoPP 2014**. 4. Wei Zhang, O Tardieu, D Grove, B Herta, T Kamada, V Saraswat, M Takeuchi “GLB: Life-line based Global Load Balancing library in X10”, Workhop on Parallel Programming for Analytic Applications, Proceedings of Principles and Practice of Parallel Programming, **PPoPP 2014**. 5. T Yuki, P Feautrier, S Rajopadhye, V Saraswat “Array dataflow analysis for polyhedral X10 programs”, Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming, **PPoPP 2013**. 6. M Takeuchi, D Cunningham, D Grove, V Saraswat “Java interoperability in Managed X10”, Proceedings of Third ACM SIGPLAN **X10 Workshop**, pp 39—46, 2013. 7. Tardieu, N Nystrom, I Peshansky and V Saraswat “Constrained Kinds”, **OOPSLA 2012**. 8. Y Zibin, D Cunningham, I Peshansky, V Saraswat “Object initialization in X10”, **ECOOP 2012**. 9. D Cunningham, R Bordawekar, V Saraswat “GPU programming in a High-level language: compiling X10 to CUDA”, Proceedings of the ACM SIGPLAN **X10 workshop**, **2011**. 10. D Grove, O Tardieu, D Cunningham, B Herta, I Peshansky, V Saraswat “A Performance Model for X10 Applications”, Proceedings of the ACM SIGPLAN **X10 workshop**, **2011**. 11. V Saraswat, George Almasi, Ganesh Bikshandi, Calin Cascaval, D Cunningham, D Grove, Sreedhar Kodali, Igor Peshansky, O Tardieu “The Asynchronous Partitioned Global Address Space Model”, AMP'10: Proceedings of the First **Workshop on Advanced in Message Passing**, **2010**. 12. Ganesh Bikshandi, Jose Castanos, Sreedhar Kodali, Krishna Nandivada, Igor Peshansky, V Saraswat, Sayantan Sur, Pradeep Varma, Tong Wen “Efficient, Portable Implementation of Asynchronous Multi-place Programs”, Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming, **PPoPP 2009**. 13. Maged Michael, Martin Vechev and V Saraswat “Idempotent Work stealing”, Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming, **PPoPP 2009**. 14. Nathaniel Nystrom, V Saraswat, Jens Palsberg and Christian Grothoff ``Constrained types for OO Languages'', **OOPSLA 2008**. 15. Satish Chandra, V Saraswat, Vivek Sarkar and Ratislav Bodik, “Type Inference for Locality Analysis of Distributed Data Structures”, Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming, **PPoPP 2008**. 16. Philippe Charles, Christian Grothoff, Kemal Ebcioglu, Allan Kielstra, Christoph von Praun, V Saraswat and Vivek Sarkar “X10: An Object-oriented Approach to Non-Uniform Cluster Computing”, Onwards! Track of the Proceedings of **OOPSLA 2005**. 17. V Saraswat and R Jagadeesan “Concurrent Clustered Programming”, Proceedings of **CONCUR**, **2005**. 18. V Saraswat “Report on the Experimental Language X10”, July 2005. Multiple versions published, over many years, with additional authors.   **Semantic Foundations for Concurrent Programming**   1. V Saraswat, R Jagadeesan, Maged Michael and Christoph von Praun, “A Theory of Memory Models”, Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming, **PPoPP 2007**. 2. V Saraswat “Constraint-Based Memory Machines: A framework for Java Memory Models”, **ASIAN 2004**, pp 494-508. 3. V Saraswat, V Gupta, R Jagadeesan “TCC, With History”, Book chapter *Horizons of the Mind*, 2014, pp 458-475. 4. Catuscia Palamidessi, V Saraswat, Frank Valencia and Bjorn Victor “On the expressiveness of linearity and persistence in the pi-calculus”, **LICS 2006**. 5. V Saraswat and R Jagadeesan ``Static support for capabilities in Java'', Second **APPSEM workshop**, U of Nottingham, April **2003**. 6. V Gupta, R Jagadeesan and V Saraswat “Models of concurrent constraint programming”, Proceedings of the International Conference on Concurrency Theory, LNCS 1119, **Concur 1996**. 7. Ugo Montanari, F Rossi, and V Saraswat “Event structure semantics for concurrent constraint programming”, 1994. 8. Ugo Montanari, F Rossi, and V Saraswat “CC programs with both in- and non-determinism”, Principles and Practice of Constraint Programming, PPCP 1995, Springer Verlag Lecture Notes in Computer Science, **LNCS 874**, pp 162-172. 9. Prakash Panangaden, V Saraswat, Phillip J. Scott and Robert Seely, “A Hyperdoctrinal view of concurrent constraint programming”, Proceedings of the REX Workshop on Semantics: Foundations and Applications, eds. J. W. deBakker, W.-P. de Rover and G. Rozenberg, **LNCS 666**, pp 457-476, 1993. 10. V Saraswat “The category of constraint systems is Cartesian-closed”, Proceedings of the Symposium on Logic in Computer Science, Santa Cruz, June 1992, **LICS 92**. 11. R Jagadeesan, V Saraswat and Vasant Shanbhogue “Angelic non-determinism in concurrent constraint programming”, Technical Report, Xerox PARC, 1992. 12. V Saraswat and Rinard M. and Panagaden, P. “The semantic foundations of concurrent constraint programming”, Proceedings of the ACM Symposium on Principles of Programming Languages, Orlando, January 1991, **POPL 91**. 13. Patrick Lincoln and V Saraswat “Proofs as concurrent processes: A logical interpretation for concurrent constraint programming”, Technical Report, Systems Sciences Laboratory, Xerox PARC, November 1991. (Revised report, Penn State University, April 2003.) 14. V Saraswat and Rinard, M. “Concurrent constraint programming”, Proceedings of the ACM Symposium on Principles of Programming Languages 1990, San Francisco, January 1990, **POPL 90**. 15. V Saraswat “A somewhat logical formulation of CLP synchronization primitives”, Proceedings of **Symposium** on **Logic Programming**, August **1988**, MIT Press.   **Real-time and Hybrid Systems**   1. V Gupta, R Jagadeesan and V Saraswat “Hybrid CC, Hybrid Automata, and Prgram Verification”, Hybrid Systems Workshop, DIMACS, Rutgers, October 1995. Appeared in Hybrid Systems III Verification and Control ed. R. Alur, T.A. Henzinger, E.D. Sontag (**LNCS 1066**), Springer-Verlag, Berlin, 1996. 2. V Gupta, R Jagadeesan, V Saraswat, and Daniel Bobrow “Programming in Hybrid Constraint Languages”, Hybrid Systems Workshop, Cornell, October 1994. Hybrid Systems II, ed. P. Antsaklis, W. Kohn, A. Nerode, S. Sastry (**LNCS 999**), Springer-Verlag, Berling, 1995. 3. V Saraswat, R Jagadeesan and V Gupta “Foundations of Timed Concurrent Constraint Programming’’, Proceedings of the Symposium on Logic in Computer Science, Paris, July **LICS 1994**. 4. V Saraswat, R Jagadeesan and V Gupta “Programming in Timed Concurrent Constraint Programming’’, Chapter in Constraint Programming, ed. B. Mayoh and E. Tyugu, **NATO ASI Workshop**, April **1994**.   **Concurrent programming languages and paradigms**   1. V Saraswat, R Jagadeesan and V Gupta “jcc: Integrating timed default concurrent constraint programming into Java’’, Proceedings of the Eleventh Portugese Conference on Artificial Intelligence (**EPIA '03**), Springer Verlag LNCS, December 2003. 2. V Saraswat “Java is not type-safe”, Web-note http://www.research.att.com/ vj/bug.html. 3. V Saraswat and Patrick Lincoln “Higher-order, linear concurrent constraint programming’’, Xerox PARC Technical report, August 1992. 4. V Saraswat and Kenneth Kahn and Jacob Levy “Janus: A step towards distributed constraint programming”, Proceedings of the North American Conference on Logic Programming, Austin, Texas, October 1990, **NACLP 1990**. 5. Kenneth Kahn and V Saraswat “Actors as a special case of concurrent constraint (logic) programming’’, Proceedings of the **ECOOP/OOPSLA** conference, **1990**. 6. V Saraswat “The language CP: Definition and Operational semantics”, in Proceedings of the ACM SIGACT-SIGPLAN Conference on Principles of Programming Languages, Munich, January 1987, **POPL 87**. 7. V Saraswat “CP as a general-purpose constraint-language”, Proceedings of the National Conference on Artificial Intelligence, (AAAI), Seattle, July 1987, **AAAI-87**. 8. V Saraswat “The language GHC: operational semantics and comparison with CP(!,|)”, Proceedings of the Fourth IEEE Symposium on Logic Programming, San Francisco, September, **SLP 1987**. 9. V Saraswat “Partial Correctness semantics for CP(!,|,&)”, Proceedings of the Conference on Foundations of Software Technology and Theoretical Computer Science, Springer Verlag LNCS 206, pp. 347-368, **FST&TCS 1985**. 10. V Saraswat “Problems with Concurrent Prolog”, Technical Report, CMU-CS-86-100, January 1986.   **Constraint programming in Software Engineering**   1. R Jagadeesan, Will Marrero, Corin Pitcher and V Saraswat “Timed Constraint Programming: A Declarative Approach to Usage Control’’, Proceedings of Principles and Practices of Declarative Programming, **PPDP 2005**. 2. F Rossi and V Saraswat “Constraint Programming”, in Encyclopedia of Computer Science and Technology (entry: Constraint Programming), A. Kent and J.G. Williams eds, Marcel Drekker Inc, 1994.   **Program Sketching**   1. Armando Solar-Lezama, Gilad Arnold, Liviu Tancau, Ratislav Bodik, V Saraswat and Sanjit Seshia “Sketching Stencils”, in ACM SIGPLAN Conference on Programming Language Design and Implementation, **PLDI '07**. 2. Armando Solar-Lezama, Liviu Tancau, Ratislav Bodik, V Saraswat “Combinatorial Sketching for Finite Programs”, in **ASPLOS 2006**.   **III. Applications**  **Concurrent programming: techniques, algorithms**   1. S Hamouda, J Milthorpe, P Strazdins, V Saraswat “A Resilient Framework for Iterative Linear Algebra Applications in X10”, 16th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing, **PDSEC 2015**. 2. A Shinnar, D Cunningham, V Saraswat, B Herta “M3R: increased performance for in-memory Hadoop jobs”, Proceedings of **VLDB 2012**. 3. V Saraswat, P Kambadur, S Kodali, D Grove, S Krishnamoorthy “Lifeline-based global load balancing”, Proceedings of ACM Conference on Principles and Practice of Parallel Programming **PPoPP 2011**. 4. Guojing Cong, Sreedhar Kodali, Sriram Krishnamoorthy, Doug Lee, V Saraswat and Tong Wen ``Solving large, irregular graph problems using adaptive work-stealing'', **ICPP 2008**. 5. Rajeev Motwani, Suresh Venkatsubramaniam, Rina Panigrahy, V Saraswat “On the decidability of accessibility problems”, ACM Symposium on the Theory of Computing, **STOC 2000**. 6. Saraswat, V.A. et al. “Detecting stable properties of networks in concurrent logic programming languages”, in Proceedings of the ACM Conference on Principles of Distributed Computing, Toronto, **PoDC 1988**. 7. V Saraswat “Merging many streams efficiently: the importance of atomic commitment”, chapter in “Concurrent Prolog: Collected Papers”, ed. E. Shapiro, MIT Press, December 1987.   **Visual Programming**   1. Kenneth M. Kahn and V A. Saraswat “Complete visualization of concurrent programs and their execution”, Proceedings of the IEEE **Workshop on Visual Programming**, October **1990**.   *This work led to a rich body of work on Visual Programming, cf. Pictorial Janus systems, and also to the company Animated Programs founded by Ken Kahn. The company has introduced a revolutionary product for school-children ``ToonTalk'', in the tradition of Logo. See* [*www.toontalk.com*](http://www.toontalk.com)*.*  **Multi-modal systems**   1. Stephane Maes and V Saraswat “Multi-Modal Requirements”, W3C Note, January 2003.   **Network Communities**   1. V Saraswat and F Pereira “Interaction media: Some thoughts on models for cyberspace”, Proceedings of the **Virtual Worlds in Simulation Conference**, San Francisco, January **1999**. 2. V Saraswat “Design requirements for network spaces”, Proceedings of the **Virtual Worlds in Simulation Conference**, San Francisco, January **1998**. 3. Jay Carlson, Roger Crew, Ken Fox, Richard Goddard, Dave Kormann, Erik Ostrom, John Ramsdell, V Saraswat, Andrew Wilson “The MUD Client Protocol, Version 2.1”, <http://www.moo.mud.org/mcp2>. 1997 4. V Saraswat “The dog, the catcher, the fish and the frying pan: Melding work, play and theater in network community”, **Virtual Communities 97**, February 1997, Sydney, Australia. 5. Vicki O'Day, Daniel Bobrow, Billie Hughes, Kimberly Bobrow, V Saraswat, JoAnne Talazus, Jim Walters, Cynde Welbes “Community Designers”, Participatory Design Conference, **PDC 1996**. 6. Daniel Bobrow, Vicki O'Day, V Saraswat, Billie Hughes and Jim Walters “Learning through computationally-mediated conversations: Interaction and Construction in virtual spaces”, Presented at the Annual Meeting of the American Anthropological Association, Washington D.C., **AAA 1995**. |
| 3.4. | *Other publications including books and patents.*  Books:   1. Vijay Saraswat “Concurrent Constraint Programming’’ MIT Press Logic Programming and Doctoral Dissertation Award Series, 1991.   Books edited:   1. Proceedings of ASIAN 03, ed. Vijay Saraswat, Springer Verlag LNCS 2896, 2003. 2. Constraint Programming: The Newport Papers, ed. Vijay Saraswat and Pascal van Hentenryck, MIT Press, 1995. 3. Proceedings of the 1991 International Symposium on Logic Programming, ed. Vijay Saraswat and Kazunori Ueda, MIT Press, 1991.   Special issue of Journals:   1. Vijay Saraswat, Pascal van Hentenryck Special issue of CONSTRAINTS on Strategic Directions in Constraint Programming, February 1997. 2. Pascal van Hentenryck, Vijay Saraswat, ed. ``Constraint Programming'', in Special Issue of ACM Computing Surveys on Strategic Directions in Computer Science, February 1997.   Patents:   1. US Patent 9,547,821 ``Deep learning for algorithm portfolios'', with Andrea Loreggia, Yuri Malitsky and Horst Samulowitz, dated 1/17 2. US Patent 9,147,373 ``Transparent efficiency for in-memory execution of map reduce job sequences'', with David Cunningham, Ben Herta, and Avi Shinnar, dated 9/15 3. US Patent 9,135,083 ``Methods for single-owner multi-consumer work queues for repeatable tasks'', with Maged Michael and Martin Vechev, dated 9/15 4. US Patent 8,924,946 ``Systems and methods for automatically optimizing high performance computing programming languages'', with Ganesh Bikshandi, Krishna Nandivada Venkata, Igor Peshansky, dated 12/14 5. US Patent 8,869,155 on ``Increasing parallel program performance for irregular memory access problems with virtual data partitioning and hierarchical collectives'', with George Almasi, Guojing Cong, David Klapecki, dated 10/14 6. US Patent 8,726,238 on ``Interactive, iterative program parallelization based on dynamic feedback'', with Robert Fuhrer and Evelyn Duesterwald, dated 5/14. 7. US Patent 8,266,394 on ``Methods for single-owner multi-consumer work queues for repeatable tasks''. with Maged Michael and Martin Vechev, dated 9/12. 8. US Patent 7,275,217, 2007 on ``System and method for multi-modal browsing with integrated update feature'', with Vijaybalaji Prasanna, Rohitashva Mathur and Shirish Vaidya, dated 9/07. 9. US Patent 5,831,853 ``Automatic construction of digital controllers/device drivers for electro-mechanical systems using component models'', dated 11/98. 10. US Patent 5,701,557 ``Machine graphs and capabilities to represent document output terminals composed of arbitrary configurations'', dated 12/97. 11. US Patent 5,696,893 ``System for generically describing and scheduling operations of modular printing machine'', dated 12/97. 12. US Patent 5,631,740 ``Transducers with constraints model for print scheduling'', dated 5/97. 13. US Patent 5,617,214 ``Commitment groups to generalize the scheduling of interdependent document output terminal capabilities'', dated 4/97. 14. US Patent 5,504,568 ``Print sequence scheduling system for duplex printing apparatus'', dated 4/96 |
| 3.5. | *Account for any funding you have received.*  For almost my entire career I have worked at industrial research labs (PARC, AT&T Research, IBM TJ Watson). Most of my work during this time was internally funded. However, I did have opportunities to get external funding. I was the Principal Investigator (or co-PI) on all of them, typically writing the main proposal.   1. Principal Investigator on “Resilient X10”, 2 year, $500K grant from US Air Force Research Lab, 2013-2014. (Award to IBM Watson Research Lab) 2. Co-Principal Investigator (with Dave Grove), sub-contracting to MIT (Prof Saman Amarasinghe) US Department of Energy X-Stack award “CAP3: A Computer-Aided Performance Programming Platform”, 3 years, $1500K, 2012-2014. (Award to IBM TJ Watson Research). DoE extended the award by one year, 2015. 3. Co-Principal Investigator (with Danny Bobrow), “Testing of hybrid and reactive systems”, 1 year, $150K NASA, 1994-1995. (Award to Xerox PARC) 4. Principal Investigator, “Timed Concurrent Constraint Programming”, 2 years, $120K, ONR, 1994--1996. (Award to Xerox PARC) 5. Co-Principal Investigator (with Danny Bobrow), “Articulate Spaces: Model-based authentic environments for Collaborative Learning”, 2 years, $2000K ARPA, Department of Defense, 1995-1997. 6. Co-Principal Investigator (with Danny Bobrow, Billie Hughes, Jim Walters), “Collaborative learning spaces”, US National Science Foundation, 1 year, 1996-97.   In addition, in 1991 Prof Seif Haridi (KTH, and Director, Swedish Institute of Computer Science, Stockholm) and I conceived of the ACCLAIM project (Advanced Concurrent Constraint Languages --- Applications, Implementation and Methodology) based in part on my thesis work on Concurrent Constraint Programming. (I was at Xerox PARC at that time, and unable to receive funding.) My role was adviser to Prof Haridi and I contributed ideas and text to the proposal.  The project was funded for several million ECUs by ESPRIT. It involved the Max-Planck Institut and DEC Paris Research Lab, INRIA, DFKI, SICS, RISC Linz, Universidad Politecnica de Madrid, Universita di Pisa, Marseille Luminy and Katholieke Universiteit Leuven. It led to the development of AKL (Andorra Kernel Language) and the Oz system at DFKI. The list of deliverables from the project at  http://www.sics.se/ps/acclaim/deliverables/perpartner.html  enumerates approximately a hundred papers on concurrent constraint programming. |
| 3.6. | *Describe active participation in national and international conferences over the past five years. State activities, such as plenary lectures, invited lectures, articles or items, chairmanship, session organisation etc.*  **Program Committee Membership past five years:**   1. Program Committee, AAAI 2016, ECOOP 2016, PPoPP 2016. 2. Extended Committee, PLDI 2015. 3. Program Committee, PPoPP 2014, ParSearchOpt 2014, EuroPar 2014. 4. Program Committee, PPoPP 2013, SuperComputing 2013. 5. Program Committee, OOPSLA 2010, PPoPP 2010; External Program Committee, PLDI 2010. 6. Steering Committee, P2S2 (Parallel Programming Models and Systems Software for High-end Computing), 2008-2016.   **Invited Talks past five years:**   1. Invited talk at ILP'16 (Inductive Logic Programming), London, Sep 2016. 2. Invited talk at DISCO'16, Crete, Greece, June 2016. 3. Invited talk on X10 at U Pennsylvania Computer Science Department, Feb 2015. 4. Invited talk on “Writing Robust Applications in Resilient X10” at ACSI 2015, Tsukuba, Jan 2015. 5. Invited talk at Exascale Runtime workshop on “Resilient X10” at TU Munich, December 2014. 6. Invited talk on “Resilient X10” at ETH Software Correctness and Reliability Worshop, Zurich, October 2014. 7. Participant in Dagstuhl meeting on “Constraints, Data and Optimization”, Dagstuhl, October 2014. 8. Invited talk at CP'14, Lyon, September 2014 (“The Concurrent Constraint Programming Research Programmes – Redux”). 9. Keynote on “Computing in the post-cloud era” InForum, Porto, Portugal, September 2014. 10. Invited talk on “Programming in X10”, Department of Engineering, Padua University, June 2014. 11. Invited talk on “C10: Probabilistic Concurrent Constraint Programming” at Software Day, Tsinghua University, April 2014. 12. Invited talk on “Constraints Solvers in X10” at CPAIOR 2013. 13. Invited talk on “X10” at AICS International Symposium, Kobe, February 2013. 14. Invited talk at Parallel Constraint Solver workshop at Shonan, Tokyo, May 2012. 15. Invited talk at Prof Aki Yonezawa Festschrifft, Kobe, May 2012. 16. Invited talk at PMCS Workshop, International Conference on Constraint Programming (CP'11), Perugia, Sep 2011. 17. Keynote at Ohio LinuxFest 2011 on “Programming in the Concurrency Era: The X10 Programming Language”, Columbus, Sep 2011. 18. Keynote at LCPC on “Constrained types: What they are, and what they can do for you”, Fort Collins, Sep 2011. |
| 3.7. | *National and international awards.*   1. ACM SIGPLAN OOPSLA Ten-year “Most Influential Paper” award (for 2005 OOPSLA X10 paper), 2015. 2. IBM Outstanding Scientific Accomplishment for X10, 2013. 3. IBM Outstanding Technical Achievement, for contribution to PERCS, 2013. 4. HPC Challenge Award for “Best Performance” for X10, at SuperComputing, 2012. 5. HPC Challenge Award for “Best Performance” for X10 and UPC, at SuperComputing, 2009. 6. HPC Challenge Award for “Most productive research implementation” for X10 and UPC, at SuperComputing, 2008. 7. HPC Challenge Award for “Most productive research implementation” for X10 (on behalf of X10 team), at SuperComputing, 2007. 8. “Most influential paper in 20 years in Concurrent Constraint Programming” award from the Association of Logic Programming for POPL90 paper (w/ Martin Rinard), 2004. 9. Excellence in Research Award, Xerox PARC, 1994. 10. Excellence in Support of Research Award, Xerox PARC, 1992. 11. ACM Doctoral Dissertation Award for the best Computer Science PhD Thesis in 1989. 12. Ratan Swaroop Gold Medal for all-round excellence in B Tech graduating class, Indian Institute of Technology, Kanpur, 1982. |
| 3.8. | *Membership in academies/committees, etc.*   * Member of the W3C Working Group on Multi-Modal Interaction (2002). |
| 3.9. | *Assignments as reviewer /independent expert*   1. Referred for Journal of Logic Programming, Artificial Intelligence Journal, Journal of Association for Computing Machinery, Theoretical Computer Science; IEEE Computer; IEEE Software; Science of Computer Programming; IEEE Transactions on Computers. (Reviewing load: sporadic) 2. Past Editorial Board member for Journal of Logic Programming. 3. On numerous expert panels for US National Science Foundation, US Department of Energy ASCR. 4. External Expert Reviewer NSERC Canada – 2017. 5. Independent Reviewer, Royal Society (UK) – 2017. 6. Invited Expert for OECD Expert Meeting on “Testing computers at human scale”, May 2016. 7. Invited Expert Reviewer for INRIA Research Theme: “Embedded and Real-time Systems”, 2016. 8. Invited Expert, NSF Workshop on Cognitive Assistants, Almaden, May 2016.   *Thesis committee member: (Assignment as opponent)*   1. Stefan Muller “Latency-hiding Work-stealing” – tentative title (PhD exp. 2017, CMU Computer Science, Advisor: Prof Umut Acar) 2. Arvind Neelkanthan “Knowledge Representation and Reasoning with Deep Neural Networks”, (PhD exp. 2017, U Mass, Amherts, Advisor: Prof Andrew McCullum) 3. Laura Tittolo “An Abstract Interpretation Framework for Diagnosis and Verification of Timed Concurrent Constraint Languages” (PhD 2014, U. Udine, Advisor: Prof Marco Comini) 4. Sophia Knight “The Epistemic View of Concurrency Theory”(PhD 2013, LiX Polytechnique, Advisor: Prof Frank Valencia, Prof Catuscia Palamidessi) 5. Carlos Olarte “Universal Temporal Concurrent Constraint Programming” (PhD 2009, LiX Polytechnique, Advisor: Prof Frank Valencia) 6. Venkatesh Mysore “Algorithmic Algebraic Model Checking: Hybrid Automata and Systems Biology” (PhD 2005, New York University, Advisor: Prof Bud Mishra) 7. Paul Ruet “Logique non-commutative et programmation concurrente par constraints” (PhD 1996, University of Paris, Advisor: Prof Francois Fages) 8. Eric Torng “Non-omniscient scheduling” (PhD 1994, Stanford University, Advisor: Prof Rajeev Motwani) 9. Francesca Rossi “Constraints and Concurrency” (PhD 1993, University of Pisa, Advisor: Prof Ugo Montanari) |
| 3.10. | *Other scientific work*   * Helped set up European ACCLAIM research network on concurrent constraint programming (1991). * Co-Chair of the IETF Working Group on Presence and Instant Messaging (PRIM), 2001. * Co-Chair of the IETF Working Group on Instant Messaging and Presence Protocols (IMPP) 1998-2000. |
| 4. Pedagogical merits  To support the reporting and assessment of pedagogical merits, KTH uses a pedagogical portfolio. The following items and titles are based on that portfolio. For an in-depth description, see the basic document “Pedagogical portfolio at KTH Teacher Support Web”: <https://www.kth.se/en/om/work-at-kth/cv-mall-for-anstallning-av-larare-1.471907>  Expected extent of descriptive and reflective sections in paragraph 4.1, 4.3, 4.4 and 4.6: 3-5 pages. | |
| 4.1. | Give a brief summary of your profile as a teacher in higher education (maximum half a page).  My core interests are in logic, concurrency, constraints – as they apply in programming languages and systems and AI. As such, I am interested in teaching courses on concurrent programming models and languages (particularly for data-centric, scale-out computing), on underlying theory (logic, type systems, lambda calculus), and AI (knowledge representation and reasoning, natural language understanding, deep learning). Concretely, I have taught senior/master’s level courses on Principles and Practice of Parallel Programming at Columbia (based on the X10 programming language), junior/senior course on Programming Languages, and graduate level courses on Big Data Analysis and Big Data Applications, on Model-based Programming and on the X10 programming language (details below). |
| 4.2. | *List your experiences of teaching in undergraduate, advanced/master level and doctoral level (first, second and third cycle), as well as further education. You can leave comments under point 4.5.*   1. *Teaching.* 2. *Production and development of teaching and learning material.* 3. *Educational administration and formal leadership roles.* 4. *Collaboration within study programme.* 5. *Teaching of general skills.* 6. *Supervision at bachelor and advanced/master level (first and second cycle level).* 7. *Supervision at doctoral level (third cycle level). State the student's name, university, year of admittance, funding, type of degree, when it was issued or is expected to be issued. Describe your role and the extent of your involvement as supervisor. Documents supporting your role as main supervisor should be attached to the application. If you have been de facto main supervisor, this should be supported by certificates.* 8. *Pedagogical activity outside the higher education institution e.g. via collaborations with the society and the business world.* 9. *General public presentations.* 10. *Development and use of e-learning/blended learning as teaching method.* 11. *Other pedagogical merits, for example pedagogical work related to internationalisation, diversity, sustainability or prizes and awards.*   My teaching experience:   1. June 2014: Graduate course on “Big Data Applications in X10”, U. Padova (invitation of Prof Francesca Rossi).   Based on material developed in the Columbia University course on X10 (see below), with additional material building on X10 projects, such as M3R (Main-Memory Map Reduce).   1. April 2013: Graduate course on “Big Data Analysis in X10”, U. Padova (invitation of Prof Francesca Rossi).   Based on material developed in the Columbia University course on X10 (see below).   1. Designed and taught a senior-level course on “Principles and Practice of Parallel Programming” at Columbia University, New York with a new professor, Prof Martha Kim. Courses taught in Fall 2009, 2010, 2011, 2012, 2013.   Course was organized around X10 and typically had an enrolment between 20-40 students. The course had a keystone project in which students in teams of 2-3 worked collaboratively to complete a medium-sized programming assignment. They were tested for design, implementation, performance and documentation.  In designing this course I collaborated closely with Prof Kim (her background is in hardware and systems). We developed lecture material, course assignments and projects from scratch. About hundred pages of lecture notes were developed for the course, on all aspects of parallel programming.   1. 2008: Lectures on X10, Waseda University, Tokyo (invitation of Prof Kazunori Ueda). 2. 2007: Summer course on X10, University of Pisa (invitation of Prof Ugo Montanari). 3. August 2004: 2-week summer course on “Hybrid Constraint Programming”, (with Prof R Jagadeesan) at ESSLLI, Nancy, France.   This course was based on our work on HCC, the Hybrid Concurrent Constraint Programming language developed at PARC by us, in collaboration with Vineet Gupta and Bjorn Carlson.   1. Spring 2003: CS 598f “Concurrent Constraint Programming as a Foundation for Model-based Programming”, Graduate course, Penn State University. 2. Fall 2002: CS 428 “Programming Languages”, Penn State University.   Over the years I have been a member of the PhD Examination committee for several students (see Section 3.9). Additionally, in 1992 I was the supervisor at PARC (with Prof Bill Dally at MIT) of the MIT VI-A student Clifford Tse. He produced a Master’s thesis under my supervision on “Linear Janus”.  In addition I organized two workshops on “Curricula for Concurrency and Parallelism” at OOPSLA 2009 and 2010.  In 2009 the workshop was organized with Guy Steele, Jr. and had a Program Committee consisting of Guy Blelloch (Carnegie-Mellon University), Kim Bruce (Pomona College), Shriram Krishna- murthy (Brown University), Tim Mattson (Intel Corporation), Michael L. Scott (University of Rochester), and Kathy Yelick (University of California, Berkeley and NSERC). It addressed the following questions:   * What are the “fundamental ideas” of concurrency and parallelism that every Computer Science graduate should know? That every college graduate should know? * Should sequential programming be taught as a “special case” of concurrent and parallel programming? * Should concurrency and parallelism issues be addressed in introductory computer science courses? * Should concurrency and parallelism topics be “sprinkled” in existing courses (e.g. in architecture, systems, programming languages, algorithms) – if so which topics in these courses should be taken out to make room? Should these topics be taught in their own separate streams.   The 2010 workshop had a similar theme. The presentations at this workshop (by Dan Grossmann, U Washington; Doug Lea, SUNY Oswego; Steven Bogaerts; Vijay Pai, UIUC; Jan Pedersen; Tom van Cutsem; Caitlin Sadowski; Robert Cartwrigth, Rice U; Richard Brown) are available at <http://www.cs.pomona.edu/~kim/CCP2010.html>. |
| 4.3. | *Theoretical knowledge (maximum 2 pages)*   * *Describe your insights into theory and research on teaching and learning relevant to higher education.* * *List your higher education courses in teaching and learning that you have completed. For courses taken outside of KTH, state year, university, course designation, number of credits and the course goals and aims for each course. Attach course certificate with passing grade for each course. Knowledge/competencies regarding teaching and learning that correspond to course fulfilment acquired in other ways are to be described in a similar way. Explain in what way your knowledge/competencies correspond to the course requirements and provide certificates or equivalent documents.*   (Not taken any courses in teaching and learning.)  Back in 1995, while at Xerox PARC, I started on a journey with the Pueblo project to re-imagine learning for school children in the networked age. I founded the Pueblo online community, using the (object-oriented, networked, shared, persistent, programmable) MOO system, with Jim Walters and Billie Hughes of Phoenix College. The system supported authenticated networked access to a persistent virtual world, using text-based clients from a variety of systems. Users connected to the community could connect with students from a number of schools (such as Longview Elementary in inner-city Phoenix), work and play with them. Within 18 months, the community had over 1000 characters, including over 300 students from Phoenix and New York, with over 20,000 objects. Our work led to a 2-year $2m grant from DARPA to build such networked communities for the US Department of Defense (DoD) schools in Germany and Japan, networked with each other, and supporting a constructive, virtual, online laboratory for simulation and experimentation with physical systems. We wrote a paper in the Participatory Design Conference (“Community Designers”), presented at the American Anthropological Association (see papers on “Network communities”), and received a 1 year grant from the US National Science Foundation to explore “collaborative learning spaces”.  Today, twenty years later, I believe that learning is on the verge of an even more sweeping change – because of AI and personal assistants.  Time was when an engineering student was able to use a computing device – a slide rule – to perform calculations faster than s/he could with the unaided use of the brain. Today, we can add data to computation: it is common place that a professional human’s laptop often “knows” more than the human does – it has access to vast amounts of email, documents, presentations, calendars, memos, todo lists etc on disk, and on the network, much more than the human can retain in his/her unaided memory.  Now imagine a future in which this computational “buddy” has become intelligent. It is able to participate in your everyday experience – going to class with you, listening to the lectures you listen to, to the questions being asked and answers being given, monitoring every key-stroke you type and document you read. Naturally, it is able to monitor the code you write and the interactions you have with your team members on (open source) collaborative projects. It learns with you, but it also has access (through the network) to other buddies of human learners facing similar pedagogical tasks. It can also access teacher bots to help its human. Your buddy is able to work with you to solve pedagogical tasks, e.g. help you with homework, in your exams, with programming assignments etc. Indeed, you go through life as a learner with this buddy, a constant companion, able to help you in everything you do. You are never alone. The buddy is an extension of your self. You get hired with your buddy, you perform tasks at work with your buddy, you get paid for what you and your buddy are able to accomplish together.  I believe the future belongs to such mixed human/computational “centaurs” and that pedagogical theory will need to be completely revamped for such a setting. We can see our way to the design of such buddies based on deep learning, reinforcement learning and continuous learning – though enormous technical challenges must still be overcome, even if we set aside privacy concerns. Not the least of which is “explanation” … to be effective, the buddy must be able to explain why it is making certain suggestions to its human partner in a way that makes sense to him, given his current state of understanding. Fortunately, unlike a human teacher, the buddy will have at its disposal a vast and intimate storehouse of knowledge about how its human partner understands. In principle it can tailor explanations for an “audience of one”.  Once such centaurs come to pass, the usual frame of pedagogy – the unaided human learner – will be gone forever. No longer can we afford to consider just the affordances that unaided humans have – enormously capable brains isolated within one cubic meter of space, with a discrete body boundary between sharing that which is instantaneous and holistic, and that which is slow and limited. We must re-design our pedagogy, and learner experiences for the centaur. The implications are staggering. The very meaning of “insight”, “intuition”, “understanding” and “evaluation” will change. Give that human students will continue to come from very diverse backgrounds – because of their ethnicity, culture, DNA, attitudes and mores – can their buddies provide complementary knowledge, skills, attitude so that the centaur **together** is able to accomplish what any other centaur can (e.g. get advanced degrees in any discipline they choose)? |
| 4.4. | *Approach.**Describe your underpinning view on teaching and learning and your practice as a teacher and supervisor in higher education.*  My fundamental goal as a teacher is to get the student excited about the subject matter. I love all the areas described above, and teach with passion, in the hope that my passion will pass on to at least some my students. I focus on fundamentals – it is critical the student understands the foundation of any discipline and has the tools to solve problems on their own. I strongly believe that students learn by doing, by problem solving, by writing code, by working together in teams, in open-source projects. I generally give open book exams, testing for understanding and problem solving rather than memorization. To challenge the more advanced students I tend to have a set of hard problems identified at the outset of a course, so if the student wants to move ahead at an accelerated pace they have a target to attack.  I believe we should evolve our current practice of pedagogy keeping a vision of the future (such as described in Section 4.3) in mind. Students are welcome to organize their thoughts developing whatever computational tools they want (even writing their own, if they so desire), and use these tools at exam time. For instance, they are allowed to bring to the exam all the code that has been discussed in class, or they know of, or they have worked with. They can browse the code using whatever code development environment (e.g. Eclipse) they are comfortable with, using whatever code assistance is provided. If they have built an “AI” to help them take their exams, they can bring it to the exam!  For software projects, I require students to work in teams, following well-established open-source development methodologies, e.g. Apache Software Foundation methodologies. I encourage the team to use code repositories such as git (designed for collaboration). This helps them develop skills that will stand them in good stead in the workplace. Moreover, such tools surface electronic trails that can be used to understand the contributions of each team member (what code/model/data artefacts/documentation did they commit when, building on whose contributions) and how well they played in the team, in a way that is natural and similar to the way they would be evaluated in the work place. |
| 4.5. | *Proficiency as a teacher.**Describe the proficiency you have attained as a teacher on all levels. Relate to the areas you have mentioned under points 4.3 and 4.4. Write sub-headings corresponding to those in point 4.2. Verify your work with course analyses and other documents.*  Prof Kim at Columbia was able to retrieve student evaluations for the course we taught jointly, for 2009, 2011, 2013. They are attached. They establish that we received higher than average scores for “Amount Learned”, “Overall Quality”. Instructor evaluations were high for “Classroom Delivery”, “Approachability”, and “Overall Quality” (2 out of 3 rated the instructors “Very good” or “Excellent”). Here is one comment from a student:  “*I believe this course takes an interesting approach to parallel programming. The progression of concepts is logical and easy to follow. The coverage of the course ranges from the classical, as in the Dining Philosophers problem, to the modern, as in the CUDA programming section. More than anything, the course benefits a great deal from the cooperation with IBM. Vijay is an excellent lecturer, and the guest lecturers invited to speak have been excellent.*” |
| 4.6. | *Educational development work/projects.**Describe your development as a teacher and how you want to continue developing teaching-learning in higher education.*  My focus on developing would be to engage with researchers in Education and Learning (in academia and in corporate research at companies like IBM that are investing in education and learning), particularly those working to advance the vision of centaur learning outlined above. |
|  | |
| 5. Management and collaboration | |
| 5.1. | *Describe your leadership profile*   * *Account for your personal view of academic leadership. Highlight challenges and tools for gender equality, diversity and equal treatment work at an institute of technology (1-2 pages).*   I cannot speak to challenges in education institutions on these topics having worked only for a year at a university. However, I have been privileged to work nearly 25 years at Xerox PARC, AT&T Research and IBM Research, bastions of corporate research in the United States. All three of these institutions have a stellar record in supporting equal opportunity and diversity, without regard to age, gender, sexual orientation or national origin, within the confines of US law.  I believe the ideal academic leader has read widely, and is hence conversant with a variety of views, is a person with impeccable integrity and honour, and one who realizes that respect and trust are the foundation of effective collaboration in the workplace, and that commitments matter. In a work place with thinkers, disagreements are inevitable. Bob Taylor at Xerox PARC developed the famed “CSL Dealer” meetings in which “Type 2” disagreements were encouraged – disagreements over ideas not egos. In a Type 1 disagreement neither party understands the other’s position; such disagreements are discouraged. In a [Type 2 disagreement](http://sourcesofinsight.com/fight-over-ideas-not-egos/), each side can explain the other side’s point of view, to the satisfaction of the other side. Such disagreements are often productive, leading to better quality ideas. Ultimately disagreements may be inevitable, but if they are understandable then they may become acceptable.  I believe strongly that we lead by example. No task that we ask others to do should be such that we would not do it ourselves. For instance, the best way to teach good software engineering practice and team work is to practice it – not ask students to read papers describing the theory. The best way to show the value of diversity is to practice it. I have personally collaborated (and published papers) with over a hundred researchers (including a dozen women) across many different nationalities (see [www.saraswat.org](http://www.saraswat.org). I count Prof Mary Dalrymple as one of my closest collaborators. I believe we must understand that people come to research and education with very different backgrounds, perspectives and expectations, and we must be open to many different ways of thinking. Our diversity is a cause for celebration. In my own work I have shown my appreciation for diversity by collaborating with researchers across a wide variety of technical areas – programming languages, systems, theoretical computer science, algorithms, AI, anthropology, education. My research contributions are the richer because of these collaborations. |
| 5.2. | *Management education*   * *List your education within management and the area of gender equality, diversity and equal treatment.*   I attended training groups for rising leaders at Xerox PARC in the early 90s. |
| 5.3. | *Management tasks and administration*   * *List your current and earlier management and/or administrative tasks. Describe your experience of unit supervision specifying the duration and the unit's size. Unit refers to research group, department and school etc.* * *Membership in boards/councils within universities over the past five years.* * *Own initiatives and work within the area of gender equality, diversity and equal treatment.*   I have had extensive experience running small to large groups in Corporate Research Labs and startups. Here is a list of my management and leadership experience:   1. May 2016 – current: *Chief Scientist, Deep Compliance*, IBM Research. Providing technical leadership for a group of about thirty researchers world-wide across IBM Research focusing on natural language understanding, machine learning, knowledge representation and reasoning in service of applications in the compliance space. Responsible for customer presentations and for building the strategy together for Business Development group within IBM Research. 2. February 2014 – January 2015: *Chief Scientist, Computing As a Service*, IBM Research. With Mark Wegman formed the Office of the Chief Scientist. Responsible for developing the strategic vision for CaaS, and also evaluating research projects for various levels of awards. 3. February 2008 -- February 2014: *Manager, Advanced Programming Languages*, IBM TJ Watson. Continued development of the X10 project, growing a team of about ten researchers.   During this period VP Research invested significant resources in X10, leading to a peak of about 40 people working on X10 at IBM Research Labs around the world, under my technical co-direction.  One highlight of this work was a large multi-million dollar “Open Collaborative Research” program with awards to about twenty universities world-wide for research in X10. This program was highly successful, leading to significant interest in X10, which is continuing to this day. Please see x10-lang.org for details.   1. June 2001 -- August 2002: *Vice President, Engineering*, Kirusa (startup). Responsible for recruiting the engineering team, developing product plans, architecting the product, and delivering the product. Product trialled with carrier customers in Europe in June 2002. Responsible for growing the engineering group from 0 employees to 10, establishing engineering practices, getting the product built. Also responsible for the company's Applications group, Product Delivery group, and the company's IT infrastructure. 2. July 2000 – June 2001: *Director, Engineering*, later *Chief Technology Officer*. Monsoon featured a scalable, open, extensible architecture for Instant Messaging, with two-way interop with email. Assembled the engineering and QA team from scratch, and had the product ready for beta in under three months. Development and QA teams were scattered over Mountain Lakes (NJ), Mountain View (Ca), San Diego (Ca), Oakland (Ca) and Indiana, and at peak numbered approx 20 people. 3. Nov 1998 – July 2000: *District Manager*, Network Communities Platform Group, and later *Technology Leader*, Instant Messaging and Presence, AT&T Labs. Started a new group from scratch to a size of about 40 researchers, engineers, testers, documentation writers and project managers. The group built AT&T’s Instant Messaging System and trialled with AT&T Wireless. Responsible for setting direction, recruiting, technical vision – creating the group and the product from nothing. A key feature was bringing in and mentoring a group of young contractors from India who eventually grew into leadership roles within the group and are now in positions of engineering leadership in the US and India. 4. Nov 1987 – Sep 1996: Research Staff Member, Xerox PARC. Led the Model-based Computing project (up to half a dozen researchers), engaging with the parent Xerox company in Rochester NY to develop constraint-based schedulers for the then new mid-range line of plug-and-play copiers.   Additionally, I co-organized the International Symposium on Logic Programming in San Diego (1991) with Prof Kazunori Ueda (as co-Program Chair). I was also Program Chair for the ASIAN 2005 conference. With Pascal van Hentenryck, I organized a meeting on Constraint Programming in 1994 and edited a collection of papers “Constraint Programming: The Newport Papers” published by MIT Press. |
| 5.4. | *Research policy assignments*   * *Member of state research councils or committees within them.* * *Member of other boards or committees providing grants.* * *Assessment of Swedish and foreign research applications (number/year over the past five years).* * *Member of international research councils, programmes, committees or advisory groups.* * *Other important expert and leadership assignments.*   N/A |
| 5.5. | *External contacts and external activities (experience of and plans for collaboration directly linked to individual research and education activities will be accounted for under points 3.2 and 4.2)*   * *Collaboration with trade and industry as well as authorities.* * *Member of boards within companies and authorities.*   Most of my work has been within the context of Corporate Research Labs and start-ups. However, I have also participated in and led industry research groups.   * During 1998/1999, together with David Marvit I proposed the creation of an IETF Working Group on Instant Messaging and Presence, and then led it (with him). The group eventually led to the creation of three separate Working Groups (for IM and P on top of different communication technologies). Eventually, this led to the publication of the Jabber protocol. * In 2001-2002 I was an Invited Expert of the W3C Working Group on Multi-Modal Interaction, and created the initial Requirements document. |
|  | |
| 6. Attached publications | |
| 6.1. | List maximum ten of your publications which you would like to cite in the first instance. Write a list with a brief explanation to the choice. Publications should be uploaded separately in the KTH recruitment system.  **List of uploaded papers:**  **Concurrent Constraint Programming Papers**   1. Semantics of concurrent constraint programming, POPL 91. (key: popl91.pdf) 2. V Saraswat, R Jagadeesan and V Gupta “Timed Default Concurrent Constraint Programming”, In *Journal of Symbolic Computation* 22 (5,6) 475--520, 1996. (key: tdcc.pdf) 3. V Gupta, R Jagadeesan and V Saraswat “Computing with Continuous Change”, *Science of Computer Programming*, 30 (1:2) 3-49, 1998. (key: ccc.pdf) 4. V Gupta, R Jagadeesan and V Saraswat ``Probabilistic Concurrent Constraint Programming'', Proceedings of the International Conference on Concurrency Theory, LNCS, CONCUR '97, 243-257. (key: pcc.pdf)   **X10 Papers**   1. Philippe Charles, Christian Grothoff, Kemal Ebcioglu, Allan Kielstra, Christoph von Praun, V Saraswat and Vivek Sarkar ``X10: An Object-oriented Approach to Non-Uniform Cluster Computing'', Onwards! Track of the Proceedings of OOPSLA 2005. (Key: x10.pdf) 2. Tardieu, B Herta, D Cunningham, D Grove, P Kambadur, V Saraswat, A Shinar, M Takeuchi, M Vaziri, W Zhang “X10 and APGAS at Petascale”, *ACM Transactions on Parallel Computing*, March 2016. (key: x10-impl.pdf)   **Natural Language Understanding: Glue Semantics Papers**   1. Mary Dalrymple, John Lamping, F Pereira, and V Saraswat “Quantification, Anaphora and Intensionality”, In *Journal of Logic, Language and Information* 6 (3), pp 219-273, July 1997. (key: glue-quant.pdf) 2. M Dalrymple, John Lamping, F Pereira, and V Saraswat “Intensional Verbs Without Type-Raising or Lexical Ambiguity”, In *Logic, Language and Computation*, volume 1, ed. Jerry Seligman and Dag Westerstaahl. Stanford, California: Center for the Study of Language and Information. 1996. Also in Proceedings of the Conference on Information-Oriented Approaches to Logic, Language and Computation/Fourth Conference on Situation Theory and its Applications, Saint Mary's College of California, Moraga, California. June 1994.   **Towards Deep Learning for Reasoning / Programming**   1. A Loreggia, H Samulowitz, Y Malitsky, V Saraswat ``Deep Learning for Algorithm Portfolios'', Proceedings of the National Conference of American Association of Artificial Intelligence 2016.   **Program Sketching**   1. Armando Solar-Lezama, Liviu Tancau, Ratislav Bodik, V Saraswat ``Combinatorial Sketching for Finite Programs'', in ASPLOS 2006. |
| 7 | Summary of documents and certificates that should be attached to your application |
| 7.1 | Scan your documents and certificates in the same order as listed below. The file should be attached to your complete CV and uploaded in the KTH recruitment system. Enclose following where relevant:   1. Section 1.6 Certificate of employment from your current employer with title, period of employment and placement. 2. Section 2.1 Certificates for Higher Education degrees. Specify year of graduation, type of qualification (for example, Licentiate of Technology, Doctor of Philosophy). 3. Section 2.2 Certificate for docent with year of appointment. 4. Section 2.4 You may attach previous expert opinions (for the last five years). 5. Section 3.5 Certificates for funding you have received 6. Section 4.2 Documents supporting your role as main supervisor should be attached to the application. If you have been de facto main supervisor, this should be supported by certificates. 7. Section 4.3 Certificates confirming completed higher education courses in teaching and learning. Attached course certificate with passing grade for each course. See section 4.3 regarding See section 4.3 regarding knowledge/competencies acquired in other ways and requested certificates or equivalent documents. 8. Section 4.5 Proficiency as a teacher.Verify your work with course analyses and other documents showing your qualifications in teaching. |
|  |  |