

# CV of Dr. Jan-Georg Smaus

Home page: <http://www.informatik.uni-freiburg.de/~smaus/>  
Email: [smaus@informatik.uni-freiburg.de](mailto:smaus@informatik.uni-freiburg.de)  
Address: Lehrstuhl für Grundlagen der Künstlichen Intelligenz  
Institut für Informatik  
Albert-Ludwigs-Universität Freiburg  
79085 Freiburg  
Germany  
Date of Birth: 5th July 1970  
Nationality: German  
Marital Status: Single

## 1 Research Interests

Most of the research has been done in collaboration with various colleagues.

### 1.1 Logic Programming

Until 2004 my interests were mainly in logic programming.

**Verification of logic programs with dynamic scheduling.** My first work in this area was on programs using delay declarations, in particular, proving termination and absence of runtime errors for such programs [5, 6, 1, 28, 33]. I have developed a more abstract view of dynamic scheduling by defining the so-called *input-consuming derivations* [8, 28, 32, 34]. More powerful termination conditions were found later [13, 4, 38, 39]. In 2003 and 2004, I have investigated input-consuming derivations with additional assumptions on the selection rule [17, 24, 41]. I have written surveys on termination of logic programs depending on the selection rule [2, 19].

**Prescriptive Typing.** I have worked on logic programming languages that use “prescriptive” types, such as Gödel or Mercury. In particular I have studied *subject reduction*, i.e. the property that only “well-typed” queries can occur in the computation of a typed program. I am interested in how this property can be regarded as a property of the proof-theoretic semantics of a program, and how the property can be obtained for systems allowing for subtyping [10, 11, 12, 3, 20, 36, 37]. In another work, I investigated the relationship between prescriptive typing and recursion in both logic and functional programming [15].

**Mode analysis of typed logic programs.** I have developed an automatic domain construction for mode analysis of typed logic programs [23, 9, 28, 30, 35]. I have later recast this domain construction by embedding it in an analysis framework based on ACI-unification [14, 40].

**The “Computation as Deduction” Paradigm.** Krzysztof Apt has proposed a new approach to using first order logic as a programming language, based on analogies between logic concepts and features in imperative programming languages. Together we have written an article comparing this approach with logic programming [25]. I have written an article discussing the optimality of a semantics of first-order equations, which is at the heart of Apt’s approach [16].

**Domains of propositional formulae in abstract interpretation.** I have investigated the classic abstract domains such as *Pos*, *Def* and *Share* and the relationship between them [7, 31].

## 1.2 Timed and Hybrid Systems

Since 2004 I have done research in timed and hybrid systems, and related areas, within the AVACS project.

**Heuristic Search in Timed Automata.** I am interested in applying ideas coming from artificial intelligence to model checking of timed automata. In artificial intelligence, heuristics have proven very useful for doing search. In model checking, similar techniques can be applied for generating heuristics that will speed up the search for suitably specified error state of a timed automata system [21].

**Finding error paths in hybrid systems.** The behaviour (flow) of real variables in hybrid systems over time is usually described by differential equations. Falsifying a hybrid system, i.e., give an actual trajectory over time that will lead to a suitably specified error state, is usually difficult since the solutions to the differential equations are trigonometric functions and the like. However, for non-deterministic systems, falsification is often easier because one can find error paths where the flows are described by polynomials or even piecewise linear functions [18].

**Linear pseudo-Boolean constraints.** A *linear pseudo-Boolean constraint* (LPB) is an expression of the form  $a_1 \cdot \ell_1 + \dots + a_m \cdot \ell_m \geq d$ , where each  $\ell_i$  is a *literal* and the  $a_1, \dots, a_m, d$  are natural numbers. An LPB is a generalisation of a propositional clause. It has been said that LPBs can be used to represent Boolean functions more compactly than the well-known *conjunctive* or *disjunctive* normal forms. I am interested in analysing the expressiveness of LPBs more thoroughly. Among other results, I have developed an algorithm that will transform a propositional formula into an LPB if this is possible [22, 42]

## 2 Teaching

During my undergraduate studies, I was a course assistant in an introductory course in computer science and in a course on logic.

During my PhD studies, I was a course assistant in courses on functional programming, electronic publishing, and structured programming.

In the winter semesters of 01/02 and 02/03, I was in charge of the exercises of the lecture on Computer Supported Modeling and Reasoning (Prof. Dr. Basin, Dr. Wolff). In the winter semesters of 03/04, 04/05, 05/06 and 06/07, I taught this lecture on my own. This course is concerned with how hardware and software systems can be modelled and reasoned about using various logics. In particular, the course uses the interactive theorem prover Isabelle.

### 3 Other Experiences and Qualifications

**Attended Conferences.** International Conferences on Logic Programming 1996 (Bonn), 1997 (Leuven), 1998 (Manchester), 1999 (Las Cruces), 2003 (Mumbai), and 2004 (St. Malo). PAP'96 (London), LOPSTR'98 (Manchester), SAS'98 (Pisa), PLILP/ ALP'98 (Pisa), ETAPS'99 (Amsterdam), LOPSTR'99 (Venice), SAS'99 (Venice), JFPLC'00 (Marseille), CL'00 (London), FLOPS'01 (Tokyo), LPAR'01 (Havana), FLOPS'02 (Aizu-Wakamatsu), ECAI'06 (Riva del Garda), CP'06 (Nantes).

**Attended Workshops.** “Constraint Programming for Reasoning about Programming” 1997 (Leeds, organised within project “Detecting and Exploiting Determinacy in Logic Programs”). Termination 1999 (Dagstuhl). Fall School on Security 2003 (Zürich). Summer School on Hybrid Systems 2005 (Siena).

**Organisation.** Co-organiser of the Workshop on Verification of Logic Programs 1999 (Las Cruces, ICLP'99) [29]. Member of the programme committee of Logic Programming Environments 2003 (Mumbai, ICLP'03) and of the Software Track of the ACM Symposium on Applied Computing (Seoul). Publicity chair of ICLP'03 (Mumbai).

**Refereeing.** I have acted as a referee for the conferences CL'00, ESOP, ICLP, LOPSTR, PPDP, SAS, and ACM SAC, as well as the journals AAEECC, IDPS, JAR, JLC, JLP, SCP, TOCL, TOPLAS and TPLP.

**Other Work Experience.** During my undergraduate studies, I was a part time research assistant, doing implementation work in C.

**Languages.** Apart from German and English, I have a good knowledge of French, Italian and Dutch, and a fair knowledge of Czech.

## 4 Chronology

### 4.1 School and Service

**1976–1989:** School in Saarbrücken, except for the school year 1986–87: High School in South Carolina, USA. Abitur in 1989 with a final grade of “excellent”.

**7/1989–9/1990:** Civil service (as a replacement of military service), providing home care for severely disabled people.

## 4.2 Undergraduate Studies

**10/1990–2/1996:** Study at the University of the Saarland for a “Diplom” degree in Computer Science (approximately equivalent to a Masters) with a Minor in Economics. The final grade was “good”.

**Thesis.** I submitted my Diplom thesis “Clause-K-Transformations” in the field of automated theorem proving. My supervisor was Priv.-Doz. Dr. Hans-Jürgen Ohlbach at the Max-Planck-Institut for Computer Science. Clause-K-transformations are faithful transformations between clause sets which replace clauses such as transitivity clauses with more specialised clauses. This can facilitate finding a refutation for a clause set.

## 4.3 PhD

**3/1996–2/1998:** Employed as a Research Associate under a project grant by the British Engineering and Physical Sciences Research Council (EPSRC). The project titled “Detecting and Exploiting Determinacy in Logic Programs” was a joint project of the University of Kent at Canterbury (grant holder Dr. Andy King) and the University of Leeds (grant holder Dr. Pat Hill).

**3/1998–8/1999:** Postgraduate student at the University of Kent. My PhD supervisor was Andy King. I submitted my thesis “Modes and Types in Logic Programming” in July 1999 and defended it in November 1999. The examiners were Prof. Simon Thompson (internal) and Dr. John Gallagher (external). The thesis was accepted after minor revision in February 2000, and the degree was awarded during the degree congregation at the University of Kent in July 2000.

## 4.4 Postdoc

**10/1999–05/2001:** ERCIM fellow in the Postdoctoral Fellowship Programme. First, I was at INRIA-Rocquencourt, working for Prof. Pierre Deransart. From July 2000, I was at CWI Amsterdam, working for Prof. Krzysztof Apt.

**06/2001–04/2004:** Employed as Scientific Assistant (wissenschaftlicher Assistent) at the University of Freiburg in the project MMiSS: Multimedia Instruction in Secure Systems. I was a member of the Software Engineering Group of Prof. Dr. David Basin.

**07/2004–12/07:** Employed as Scientific Assistant (wissenschaftlicher Assistent) at the University of Freiburg in the project AVACS: Automatic Verification and Analysis of Complex Systems. I am a member of the Artificial Intelligence Group of Prof. Dr. Bernhard Nebel.

## 5 Publications

Reviewed journal articles: [1, 2, 3, 4]

Reviewed conference papers: [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]

Reviewed articles in special volumes: [19]

Reviewed workshop papers: [20]  
Reviewed posters: [23, 24]  
Invited: [25, 26]  
Theses: [27, 28]  
Editorship: [29]  
Technical Reports: [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 43]

## References

- [1] J.-G. Smaus, P. M. Hill, and A. M. King. Verifying termination and error-freedom of logic programs with `block` declarations. *Theory and Practice of Logic Programming*, 1(4):447–486, 2001.
- [2] D. Pedreschi, S. Ruggieri, and J.-G. Smaus. Classes of terminating logic programs. *Theory and Practice of Logic Programming*, 2(3):369–418, 2002.
- [3] P. Deransart and J.-G. Smaus. Subject reduction of logic programs as proof-theoretic property. *Journal of Functional and Logic Programming*, 2002(2), 2002.
- [4] A. Bossi, S. Etalle, S. Rossi, and J.-G. Smaus. Semantics and termination of simply moded logic programs with dynamic scheduling. *Transactions on Computational Logic*, 5(3):470–507, 2004.
- [5] J.-G. Smaus, P. M. Hill, and A. M. King. Termination of logic programs with `block` declarations running in several modes. In C. Palamidessi, editor, *Proceedings of the 10th Symposium on Programming Language Implementations and Logic Programming*, volume 1490 of *LNCS*, pages 73–88. Springer-Verlag, 1998.
- [6] J.-G. Smaus, P. M. Hill, and A. M. King. Preventing instantiation errors and loops for logic programs with multiple modes using `block` declarations. In P. Flener, editor, *Proceedings of the 8th International Workshop on Logic-based Program Synthesis and Transformation*, volume 1559 of *LNCS*, pages 289–307. Springer-Verlag, 1999.
- [7] A. M. King, J.-G. Smaus, and P. M. Hill. Quotienting *Share* for dependency analysis. In D. Swierstra, editor, *Proceedings of the European Symposium on Programming*, volume 1576, pages 59–73, 1999.
- [8] J.-G. Smaus. Proving termination of input-consuming logic programs. In D. De Schreye, editor, *Proceedings of the 16th International Conference on Logic Programming*, pages 335–349. MIT Press, 1999.
- [9] J.-G. Smaus, P. M. Hill, and A. M. King. Mode analysis domains for typed logic programs. In A. Bossi, editor, *Proceedings of the 9th International Workshop on Logic-based Program Synthesis and Transformation*, volume 1817 of *LNCS*, pages 83–102, 2000.
- [10] J.-G. Smaus and P. Deransart. Les programmes bien typés ont tout bon. In Touraïvane, editor, *Proceedings of Journées Francophones de Programmation en Logique avec Contraintes*, pages 49–65. Hermes Science Publications, 2000.

- [11] J.-G. Smaus, F. Fages, and P. Deransart. Using modes to ensure subject reduction for typed logic programs with subtyping. In S. Kapoor and S. Prasad, editors, *Proceedings of the 20th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1974 of *LNCS*, pages 214–226. Springer-Verlag, 2000.
- [12] P. Deransart and J.-G. Smaus. Well-typed logic programs are not wrong. In H. Kuchen and K. Ueda, editors, *Proceedings of the 5th International Symposium on Functional and Logic Programming*, volume 2024 of *LNCS*, pages 280–295. Springer-Verlag, 2001.
- [13] A. Bossi, S. Etalle, S. Rossi, and J.-G. Smaus. Semantics and termination of simply-moded logic programs with dynamic scheduling. In D. Sands, editor, *Proceedings of the European Symposium on Programming*, volume 2028 of *LNCS*, pages 402–416. Springer-Verlag, 2001.
- [14] J.-G. Smaus. Analysis of polymorphically typed logic programs using ACI-unification. In R. Nieuwenhuis and A. Voronkov, editors, *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 2250 of *LNAI*, pages 280–295. Springer-Verlag, 2001.
- [15] J.-G. Smaus. The head condition and polymorphic recursion. In Z. Hu and M. Rodríguez-Artalejo, editors, *Proceedings of the 6th International Symposium on Functional and Logic Programming*, volume 2441 of *LNCS*, pages 259–274. Springer-Verlag, 2002.
- [16] J.-G. Smaus. Is there an optimal generic semantics for first-order equations? In C. Palamidessi, editor, *Proceedings of the 19th International Conference on Logic Programming*, volume 2916 of *LNCS*, pages 438–450. Springer-Verlag, 2003.
- [17] J.-G. Smaus. Termination of logic programs using various dynamic selection rules. In B. Demoen and V. Lifschitz, editors, *Proceedings of the 20th International Conference on Logic Programming*, volume 3132 of *LNCS*, pages 43–57. Springer-Verlag, 2004.
- [18] Stefan Ratschan and Jan-Georg Smaus. Verification-integrated falsification of non-deterministic hybrid systems. In C. G. Cassandras, A. Giua, and J. Zaytoon, editors, *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, 2006.
- [19] D. Pedreschi, S. Ruggieri, and J.-G. Smaus. Characterisations of termination in logic programming. In M. Bruynooghe and K. Lau, editors, *Program Development in Computational Logic*, volume 3049 of *LNCS*, pages 376–431. Springer-Verlag, 2004.
- [20] J.-G. Smaus and P. Deransart. Well-typed programs are not wrong. In M. Leuschel et al., editor, *Verification and Computational Logic Workshop*. Technical report DSEE-TR-2000-6 of the University of Southampton, 2000. Held within CL’2000.

- [21] Jörg Hoffmann, Jan-Georg Smaus, Andrey Rybalchenko, Sebastian Kupferschmid, and Andreas Podelski. Using predicate abstraction to generate heuristic functions in uppaal. In Stefan Edelkamp and Alessio R. Lomuscio, editors, *Post-Proceedings of the 4th Workshop on Model Checking and Artificial Intelligence (MoChArt 2006)*, 2006.
- [22] Jan-Georg Smaus. Representing Boolean functions as linear pseudo-Boolean constraints. In Youssef Hamadi, editor, *Proceedings of the CP 2006 Workshop on the Integration of SAT and CP techniques*, 2006.
- [23] J.-G. Smaus, P. M. Hill, and A. M. King. Domain construction for mode analysis of typed logic programs. In L. Naish, editor, *Proceedings of the 14th Joint International Conference and Symposium on Logic Programming*, page 418. MIT Press, 1997. Abstract of Poster Presentation.
- [24] J.-G. Smaus. Termination of logic programs for various dynamic selection rules. In C. Palamidessi, editor, *Proceedings of the 19th International Conference on Logic Programming*, volume 2916 of *LNCS*, pages 511–512. Springer-Verlag, 2003. Abstract of Poster Presentation.
- [25] K. R. Apt and J.-G. Smaus. Rule-based versus procedure-based view of logic programming. *Joint Bulletin of the Novosibirsk Computing Center and Institute of Informatics Systems; Series: Computer Science*, 16:75–97, 2001.
- [26] B. Krieg-Brückner, D. Hutter, A. Lindow, C. Lüth, A. Mahnke, E. Melis, P. Meier, A. Poetzsch-Heffter, M. Roggenbach, G. Russell, J.-G. Smaus, and M. Wirsing. Multimedia instruction in safe and secure systems. In M. Wirsing, D. Pattinson, and R. Henecker, editors, *Recent Trends in Algebraic Development Techniques, 16th International Workshop, WADT 2002*, volume 2755 of *LNCS*, pages 82–117. Springer-Verlag, 2003.
- [27] J.-G. Smaus. Resolution K-transformations. Master’s thesis, Universität des Saarlandes (Max-Planck-Institut für Informatik), 1996.
- [28] J.-G. Smaus. *Modes and Types in Logic Programming*. PhD thesis, University of Kent at Canterbury, 1999.
- [29] S. Etalle and J.-G. Smaus, editors. *Proceedings of the Workshop on Verification of Logic Programs held at ICLP’99*, volume 30(1) of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1999.
- [30] J.-G. Smaus, P. M. Hill, and A. M. King. Domain construction for mode analysis of typed logic programs. Technical Report 8-97, Computing Laboratory, University of Kent at Canterbury, United Kingdom, 1997. Early version of [9].
- [31] A. M. King, J.-G. Smaus, and P. M. Hill. Practical dependency analysis through a *Share* quotient. Technical Report 11-98, Computing Laboratory, University of Kent, 1998. Early version of [7].
- [32] J.-G. Smaus. Well-terminating, input-driven logic programs. Technical Report 16-98, University of Kent at Canterbury, 1998. Early version of [8].

- [33] J.-G. Smaus, P. M. Hill, and A. M. King. Verification of logic programs with `block` declarations running in several modes. Technical Report 7-98, University of Kent at Canterbury, Canterbury, CT2 7NF, United Kingdom, July 1998. Contains proofs for [5, 6]; see also [1].
- [34] J.-G. Smaus. Proving termination of input-consuming logic programs. Technical Report 10-99, Computing Laboratory, University of Kent at Canterbury, United Kingdom, 1999. Long version of [8].
- [35] J.-G. Smaus, P.M. Hill, and A.M. King. Mode analysis domains for typed logic programs. Technical Report 2000.06, University of Leeds, 2000. Long version of [9].
- [36] J.-G. Smaus, F. Fages, and P. Deransart. Using modes to ensure subject reduction for typed logic programs with subtyping. Technical Report RR-4020, INRIA, 2000. Long version of [11], available via CoRR: <http://arXiv.org/archive/cs/intro.html>.
- [37] P. Deransart and J.-G. Smaus. Well-typed logic programs are not wrong. Technical Report RR-4082, INRIA, 2000. Long version of [12], available via CoRR: <http://arXiv.org/archive/cs/intro.html>.
- [38] A. Bossi, S. Etalle, S. Rossi, and J.-G. Smaus. On the semantics and termination of logic programs with dynamic scheduling. Technical Report CS2000-11, Università Ca' Foscari di Venezia, 2000. Early version of [13].
- [39] A. Bossi, S. Etalle, S. Rossi, and J.-G. Smaus. Semantics and termination of simply-moded logic programs with dynamic scheduling. Long version of [13], available via CoRR: <http://arXiv.org/archive/cs/intro.html>, 2001.
- [40] J.-G. Smaus. Analysis of polymorphically typed logic programs using ACI-unification. Available via CoRR: <http://arXiv.org/archive/cs/intro.html>, 2001.
- [41] J.-G. Smaus. Termination of logic programs for various dynamic selection rules. Technical Report 191, Institut für Informatik, Universität Freiburg, 2003.
- [42] Jan-Georg Smaus. Representing Boolean functions as linear pseudo-Boolean constraints. Technical Report 227, Institut für Informatik, Universität Freiburg, 2006.
- [43] J.-G. Smaus. Termination of logic programs using various dynamic selection rules. Technical Report 203, Institut für Informatik, Universität Freiburg, 2004.