



Universiteit
Leiden
The Netherlands

Team automata : a formal approach to the modeling of collaboration between system components

Beek, M.H. ter

Citation

Beek, M. H. ter. (2003, December 10). *Team automata : a formal approach to the modeling of collaboration between system components*. Retrieved from <https://hdl.handle.net/1887/29570>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/29570>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/29570> holds various files of this Leiden University dissertation.

Author: Beek, Maurice H. ter

Title: Team automata : a formal approach to the modeling of collaboration between system components

Issue Date: 2003-12-10

Bibliography

- [ABC⁺95] M. Ajmone Marson, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with generalized stochastic Petri nets*, John Wiley & Sons, Chichester, 1995.
- [Arn82] A. Arnold, Synchronized Behaviours of Processes and Rational Relations. *Acta Informatica* 17 (1982), 21 – 29.
- [Arn94] A. Arnold, *Finite Transition Systems*, Prentice Hall International Series in Computer Science, London, 1994.
- [AN82] A. Arnold and M. Nivat, Comportements de processus. In *Colloque AFCET Les Mathématiques de l'Informatique*, 1982, 35 – 68. (In French.)
- [BDQT99] E. Badouel, Ph. Darondeau, D. Quichaud, and A. Tokmakoff, Modelling Dynamic Agent Systems with Cooperating Automata. Publication Interne 1253, Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, 1999.
- [BB03] M.H. ter Beek and R.P. Bloem, Model Checking Team Automata for Access Control. Unpublished manuscript, 2003.
- [BCM03] M.H. ter Beek, E. Csuhaj-Varjú, and V. Mitrană, Teams of Push-down Automata. To appear in *Proceedings of the PSI'03 Fifth International Conference on Perspectives of System Informatics, Novosibirsk, Akademgorodok, Russia* (A. Zamulin and M. Broy, eds.), Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2003. (A full version appeared as Technical Report 2002/4, Research Group on Modelling Multi-Agent Systems, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, 2002.)
- [BEKR01a] M.H. ter Beek, C.A. Ellis, J. Kleijn, and G. Rozenberg, Team Automata for CSCW. In *Proceedings of the Second International Colloquium on Petri Net Technologies for Modelling Communication Based Systems, Berlin, Germany* (H. Weber, H. Ehrig, and W. Reisig, eds.), Fraunhofer Institute for Software and Systems Engineering, Berlin, 2001, 1 – 20. (Also appeared as Technical Report TR-01-07, Leiden Institute of Advanced Computer Science, Universiteit Leiden, Leiden, 2001.)
- [BEKR01b] M.H. ter Beek, C.A. Ellis, J. Kleijn, and G. Rozenberg, Team Automata for Spatial Access Control. In *Proceedings of the ECSCW'01 Seventh European Conference on Computer Supported Cooperative Work, Bonn, Germany* (W. Prinz, M. Jarke, Y. Rogers, K. Schmidt, and V. Wulf, eds.), Kluwer Academic Publishers, Dordrecht, 2001, 59 – 77.
- [BEKR03] M.H. ter Beek, C.A. Ellis, J. Kleijn, and G. Rozenberg, Synchronizations in team automata for groupware systems. *Computer Supported*

- Cooperative Work — The Journal of Collaborative Computing* 12, 1 (2003), 21 – 69.
- [BK03] M.H. ter Beek and J. Kleijn, Team Automata Satisfying Compositionality. In *Proceedings of FME 2003: Formal Methods — the Twelfth International Symposium of Formal Methods Europe, Pisa, Italy* (K. Araki, S. Gnesi, and D. Mandrioli, eds.), *Lecture Notes in Computer Science* 2805, Springer-Verlag, Berlin, 2003, 381 – 400.
- [BLP03] M.H. ter Beek, G. Lenzini, and M. Petrocchi, Team Automata for Security Analysis of Multicast/Broadcast Communication. In *Proceedings of the WISP'03 Workshop on Issues in Security and Petri Nets, Eindhoven, The Netherlands* (N. Busi, R. Gorrieri and F. Martinelli, eds.), Beta Research School for Operations Management and Logistics, Department of Technology Management, Eindhoven University of Technology, Eindhoven, 2003, 57 – 71. (Also appeared as Technical Report 2003-TR-13, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, 2003.)
- [BPS01] *Handbook of Process Algebra* (J.A. Bergstra, A. Ponse, and S.A. Smolka, eds.), Elsevier Science Publishers, Amsterdam, 2001.
- [BC92] L. Bernardinello and F. De Cindio, A Survey of Basic Net Models and Modular Net Classes. In *Advances in Petri Nets 1992* (G. Rozenberg, ed.), *Lecture Notes in Computer Science* 609, Springer-Verlag, Berlin, 1992, 304 – 351.
- [BÉ96] S.L. Bloom and Z. Ésik, Free Shuffle Algebras in Language Varieties. *Theoretical Computer Science* 163 (1996), 55 – 98.
- [BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM* 31, 3 (1984), 560 – 599.
- [Bul98] A. Bullock, *SPACE: Spatial Access Control in Collaborative Virtual Environments*. Ph.D. thesis, Department of Computer Science, University of Nottingham, 1998.
- [BB97] A. Bullock and S. Benford, Access Control in Virtual Environments. In *Proceedings of the VRST'97 ACM Symposium on Virtual Reality Software and Technology, Lausanne, Switzerland* (D. Thalman, S. Feiner, and G. Singh, eds.), ACM Press, New York, 1997, 29 – 35.
- [BB99] A. Bullock and S. Benford, An access control framework for multi-user collaborative environments. In *Proceedings of the GROUP'99 International ACM SIGGROUP Conference on Supporting Group Work, Phoenix, Arizona*, ACM Press, New York, 1999, 140 – 149.
- [CH74] R.H. Campbell and A.N. Habermann, The Specification of Process Synchronisation by Path Expressions. In *Proceedings of an International Symposium on Operating Systems, Rocquencourt, France* (E. Gelenbe and C. Kaiser, eds.), *Lecture Notes in Computer Science* 16, Springer-Verlag, Berlin, 1974, 89 – 102.
- [CC02] J. Carmona and J. Cortadella, Input/Output Compatibility of Reactive Systems. In *Fourth International Conference on Formal Methods in Computer-Aided Design, Portland, Oregon* (M.D. Aagaard and J.W. O'Leary, eds.), *Lecture Notes in Computer Science* 2517, Springer-Verlag, Berlin, 2002, 360 – 377.

- [CCP02] J. Carmona, J. Cortadella, and E. Pastor, Synthesis of Reactive Systems: Application to Asynchronous Circuit Design. In *Concurrency and Hardware Design — Advances in Petri Nets* (J. Cortadella, A. Yakovlev, and G. Rozenberg, eds.), Springer-Verlag, Berlin, 2002, 107 – 151.
- [CW96] E.M. Clarke and J.M. Wing, Formal methods: State of the art and future directions. *ACM Computing Surveys* 28, 4 (1996), 626 – 643.
- [Dar91] S. Dart, Concepts in Configuration Management Systems. In *Proceedings of the Third International Workshop on Software Configuration Management, Trondheim, Norway* (P.H. Feiler, ed.), ACM Press, New York, 1991, 1 – 18.
- [DKW99] *Software Process: Principles, Methodology, Technology* (J.-C. Derniame, A.B. Kaba, and D. Wastell, eds.), *Lecture Notes in Computer Science* 1500, Springer-Verlag, Berlin, 1999.
- [DeS84] R. De Simone, Langages Infinitaires et Produit de Mixage. *Theoretical Computer Science* 31 (1984), 83 – 100.
- [Dew01] P. Dewan, An integrated Approach to Designing and Evaluating Collaborative Applications and Infrastructures. *Computer Supported Cooperative Work — The Journal of Collaborative Computing* 10, 1 (2001), 75 – 111.
- [DCS94] P. Dewan, R. Choudhary, and H. Shen, An Editing-Based Characterization of the Design Space of Collaborative Applications. *Journal of Organizational Computing* 4, 3 (1994), 219 – 240.
- [DS98] P. Dewan and H. Shen, Flexible Meta Access-Control for Collaborative Applications. In *Proceedings of the CSCW'98 ACM Conference on Computer Supported Cooperative Work, Seattle, Washington* (E. Churchill, D. Snowdon, and G. Golovchinsky, eds.), ACM Press, New York, 1998, 247 – 256.
- [DR95] V. Diekert and G. Rozenberg, *Book of Traces*, World Scientific, Singapore, 1995.
- [DH94] D. Drusinsky and D. Harel, On the Power of Bounded Concurrency I: Finite Automata. *Journal of the ACM* 41, 3 (1994), 517 – 539.
- [Dub86] C. Duboc, Mixed Product and Asynchronous Automata. *Theoretical Computer Science* 42 (1986), 183 – 199.
- [Ell97] C.A. Ellis, Team Automata for Groupware Systems. In *Proceedings of the GROUP'97 International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge, Phoenix, Arizona* (S.C. Hayne and W. Prinz, eds.), ACM Press, New York, 1997, 415 – 424.
- [EGR90] C.A. Ellis, S.J. Gibbs, and G. Rein, Design and Use of a Group Editor. In *Engineering for Human Computer Interaction* (G. Cockton, ed.), North-Holland Publishing Company, Amsterdam, 1990, 13 – 25.
- [EK00] C.A. Ellis and K.-H. Kim, A Framework and Taxonomy for Workflow Architecture. In *Designing Cooperative Systems: The Use of Theories and Models — Proceedings of the COOP'2000 International Conference on the Design of Cooperative Systems, Sophia Antipolis, France* (R. Dieng, A. Giboin, L. Karsenty, and G. De Michelis, eds.), *Frontiers in Artificial Intelligence and Applications* 58, IOS Press, Amsterdam, 2000, 99 – 112.

- [EN93] C.A. Ellis and G.J. Nutt, Modelling and Enactment of Workflow Systems. In *Proceedings of the ATPN'93 International Conference on Application and Theory of Petri Nets, Chicago, Illinois* (M. Ajmone Marsan, ed.), *Lecture Notes in Computer Science* 691, Springer-Verlag, Berlin, 1993, 1 – 16.
- [EG02] G. Engels and L.P.J. Groenewegen, Towards Team-Automata-Driven Object-Oriented Collaborative Work. In *Formal and Natural Computing - Essays Dedicated to Grzegorz Rozenberg* (W. Brauer, H. Ehrig, J. Karhumäki, and A. Salomaa, eds.), *Lecture Notes in Computer Science* 2300 (2002), 257 – 276.
- [GSSL94] R. Gawlick, R. Segala, F.F. Søggaard-Andersen, and N. Lynch, Liveness in Timed and Untimed Systems. In *Proceedings of the ICALP'94 Twenty-first International Colloquium on Automata, Languages and Programming, Jerusalem, Israel* (S. Abiteboul and E. Shamir, eds.), *Lecture Notes in Computer Science* 820, Springer-Verlag, Berlin, 1994, 166 – 177. (A full version appeared as Technical Report MIT/LCS/TR-587, Massachusetts Institute of Technology, Cambridge, Massachusetts.)
- [GS65] S. Ginsburg and E.H. Spanier, Mappings of Languages by Two-Tape Devices. *Journal of the ACM* 12, 3 (1965), 423 – 434.
- [Gis81] J.L. Gischer, Shuffle Languages, Petri Nets, and Context Sensitive Grammars. *Communications of the ACM* 24 (1981), 597 – 605.
- [Gru94] J. Grudin, CSCW: History and Focus. *IEEE Computer* 27, 5 (1994), 19 – 26.
- [Har87] D. Harel, Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming* 8 (1987), 231 – 274.
- [HH94] T. Hirst and D. Harel, On the Power of Bounded Concurrency II: Pushdown Automata. *Journal of the ACM* 41, 3 (1994), 540 – 554.
- [Hoa78] C.A.R. Hoare, Communicating Sequential Processes. *Communications of the ACM* 21, 8 (1978), 666 – 677.
- [Hoa85] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice Hall International Series in Computer Science, London, 1985.
- [Hoe01] P.J. 't Hoen, *Towards Distributed Development of Large Object-Oriented Models — Views of Packages as Classes*. Ph.D. thesis, Leiden Institute of Advanced Computer Science, Leiden University, 2001.
- [HB00] P.J. 't Hoen and M.H. ter Beek, A Conflict-Free Strategy for Team-Based Model Development. In *Proceedings of the PDTSD'00 International Workshop on Process support for Distributed Team-based Software Development in Volume IX: Industrial Systems of the Proceedings of the SCI'00 World MultiConference on Systemics, Cybernetics and Informatics, Orlando, Florida* (B. Sanchez, R. Hammel II, M. Soriano, and P. Tiako, eds.), International Institute of Informatics and Systemics, 2000, 720 – 725.
- [Hol91] G.J. Holzmann, *Design and Validation of Computer Protocols*, Prentice Hall International, Inc., Englewood Cliffs, New Jersey, 1991.
- [Hol97] G.J. Holzmann, The model checker SPIN. *IEEE Transactions on Software Engineering* 23, 5 (1997), 279 – 295.

- [Hol03] G.J. Holzmann, *The SPIN Model Checker: Primer and Reference Manual*, Addison Wesley Publishers, Reading, Massachusetts, 2003.
- [IEEE93] ANSI/IEEE Standard 1042-1987, *IEEE Guide to Software Configuration Management. IEEE Standards Collection — Software Engineering, 1993 Edition*, Institute of Electrical and Electronics Engineers, Inc., New York, 1993.
- [JL92] R. Janicki and P.E. Laurer, *Specification and Analysis of Concurrent Systems, The COSY Approach. EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, Berlin, 1992.
- [Jan81] M. Jantzen, The Power of Synchronizing Operations on Strings. *Theoretical Computer Science* 14 (1981), 127 – 154.
- [Jon87] B. Jonsson, Compositional Verification of Distributed Systems. Ph.D. thesis, Department of Computer Systems, Uppsala University, 1987.
- [Jon94] B. Jonsson, Compositional Specification and Verification of Distributed Systems. *ACM Transactions on Programming Languages and Systems* 16, 2 (1994), 259 – 303.
- [Kee96] N.W. Keesmaat, *Vector Controlled Concurrent Systems*. Ph.D. thesis, Department of Computer Science, Leiden University, 1996.
- [KK97] N.W. Keesmaat and H.C.M. Kleijn, Net-based Control versus Rational Control: The Relation between ITNC Vector Languages and Rational Relations. *Acta Informatica* 34 (1997), 23 – 57.
- [KKR90] N.W. Keesmaat, H.C.M. Kleijn, and G. Rozenberg, Vector Controlled Concurrent Systems, Part I: Basic Classes. *Fundamenta Informaticae* 13 (1990), 275 – 316.
- [KKR91] N.W. Keesmaat, H.C.M. Kleijn, and G. Rozenberg, Vector Controlled Concurrent Systems, Part II: Comparisons. *Fundamenta Informaticae* 14 (1991), 1 – 38.
- [KB95] S. Khoshafian and M. Buckiewicz, *Introduction to Groupware, Workflow, and Workgroup Computing*, John Wiley & Sons, New York, 1995.
- [Kim76] T. Kimura, An Algebraic System for Process Structuring and Interprocess Communication. In *Proceedings of the Eighth ACM SIGACT Symposium on Theory of Computing, Hershey, Pennsylvania*, ACM Press, New York, 1976, 92 – 100.
- [Kur94] R.P. Kurshan, *Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach*, Princeton University Press, Princeton, New Jersey, 1994.
- [LMP00] R. Lanotte, A. Maggiolo-Schettini, and A. Peron, Timed Cooperating Automata. *Fundamenta Informaticae* 42 (2000), 1 – 21.
- [LR99] M. Latteux and Y. Roos, Synchronized Shuffle and Regular Languages. In *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa* (J. Karhumäki, H.A. Maurer, Gh. Păun, and G. Rozenberg, eds.), Springer-Verlag, Berlin, 1999, 35 – 44.
- [LTS79] P.E. Laurer, P.R. Torregiani, and M.W. Shields, COSY — A System Specification Language based on Paths and Processes. *Acta Informatica* 12 (1979), 109 – 158.
- [Lav00] H. Lavana, *A Universally Configurable Architecture for Taskflow-Oriented Design of a Distributed Collaborative Computing Environ-*

- ment. Ph.D. thesis, Department of Electrical and Computer Engineering, North Carolina State University, 2000.
- [Lyn96] N.A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers, San Mateo, California, 1996.
- [LT87] N.A. Lynch and M.R. Tuttle, Hierarchical Correctness Proofs for Distributed Algorithms. In *Proceedings of the Sixth ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Vancouver, British Colombia, Canada*, 1987, 137 – 151.
- [LT89] N.A. Lynch and M.R. Tuttle, An Introduction to Input/Output Automata. *CWI Quarterly* 2, 3 (1989), 219 – 246. (Also appeared as Technical Memo MIT/LCS/TM-373, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988.)
- [Maz89] A. Mazurkiewicz, Basic Notions of Trace Theory. In *Lecture Notes in Computer Science* 354, Springer-Verlag, Berlin, 1989, 285 – 363.
- [Mey92] B. Meyer, Applying Design by Contract. *IEEE Computer* 25, 10 (1992), 40 – 51.
- [Mil80] R. Milner, *A Calculus of Communicating Systems, Lecture Notes in Computer Science* 92, Springer-Verlag, Berlin, 1980.
- [Mil89] R. Milner, *Communication and Concurrency*, Prentice Hall International Series in Computer Science, London, 1989.
- [Niv79] M. Nivat, Sur la synchronisation des processus. *Revue Technique Thomson-CSF* 11 (1979), 899 – 919. (In French.)
- [Ohe03] D. von Oheimb, Interacting State Machines: A Stateful Approach to Proving Security. To appear in *Proceedings of the BCS-FACS International Conference on Formal Aspects of Security* (A. Abdallah, P. Ryan, and S. Schneider, eds.), *Lecture Notes in Computer Science* 2629, Springer-Verlag, 2003.
- [OL02] D. von Oheimb and V. Lotz, Formal Security Analysis with Interacting State Machines. In *Proceedings of the Seventh ESORICS'02 European Symposium on Research in Computer Security* (D. Gollmann, G. Karjoth, M. Waidner, eds.), *Lecture Notes in Computer Science* 2502, Springer-Verlag, 2002, 212 – 228.
- [Par79] D. Park, On the Semantics of fair parallelism. In *Lecture Notes in Computer Science* 86, Springer-Verlag, Berlin, 1979, 504 – 526.
- [Pet62] C.A. Petri, *Kommunikation mit Automaten*. Rheinisch-Westfälisches Institut für Instrumentelle Mathematik an der Universität Bonn, Schrift Nr. 2, 1962. (In German.)
- [RR98a] *Lectures on Petri Nets I: Basic Models* (W. Reisig and G. Rozenberg, eds.), *Lecture Notes in Computer Science* 1491, Springer-Verlag, Berlin, 1998.
- [RR98b] *Lectures on Petri Nets II: Applications* (W. Reisig and G. Rozenberg, eds.), *Lecture Notes in Computer Science* 1492, Springer-Verlag, Berlin, 1998.
- [Rod96] T. Rodden, Populating the Application: A Model of Awareness for Cooperative Applications. In *Proceedings of the CSCW'96 ACM Conference on Computer Supported Cooperative Work, Boston, Massachusetts* (M. Ackerman, ed.), ACM Press, New York, 1996, 87 – 96.

- [Ros97] A.W. Roscoe, *The Theory and Practice of Concurrency*, Prentice Hall International Series in Computer Science, London, 1997.
- [RS97] *Handbook of Formal Languages* (G. Rozenberg and A. Salomaa, eds.), Springer-Verlag, Berlin, 1997.
- [RBP⁺91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall International, Inc., Englewood Cliffs, New Jersey, 1991.
- [Sha78] A.C. Shaw, Software Descriptions with Flow Expressions. *IEEE Transactions on Software Engineering* SE-4, 3 (1978), 242 – 254.
- [SD92] H. Shen and P. Dewan, Access Control for Collaborative Environments. In *Proceedings of the CSCW'92 ACM Conference on Computer Supported Cooperative Work, Toronto, Canada* (J. Turner and R. Kraut, eds.), ACM Press, New York, 1992, 51 – 58.
- [Shi79] M.W. Shields, Adequate Path Expressions. In *Proceedings of the Symposium on the Semantics of Concurrent Computation, Evian, France* (G. Kahn, ed.), *Lecture Notes in Computer Science* 70, Springer-Verlag, Berlin, 1979, 249 – 265.
- [Shi97] M.W. Shields, *Semantics of Parallelism — Non-Interleaving Representation of Behaviour*, Springer-Verlag, Berlin, 1997.
- [Sik97] K. Sikkil, A Group-based Authorization Model for Cooperative Systems. In *Proceedings of the ECSCW'97 Fifth European Conference on Computer Supported Cooperative Work, Lancaster, UK* (J. Hughes, W. Prinz, T. Rodden, and K. Schmidt, eds.), Kluwer Academic Publishers, Dordrecht, 1997, 345 – 360.
- [Smi94] J. Smith, *Collective Intelligence in Computer Based Collaboration — A Volume in the Computers, Cognition, and Work Series*, Lawrence Erlbaum Associates, Mahwah, New Jersey, 1994.
- [vdS85] J.L.A. van de Snepscheut, *Trace Theory and VLSI Design, Lecture Notes in Computer Science* 200, Springer-Verlag, Berlin, 1985.
- [TH98] P.S. Thiagarajan and J.G. Henriksen, Distributed Versions of Linear Temporal Logic: A Trace Perspective. In [RR98a] (1998), 643 – 681.
- [Tut87] M.R. Tuttle, *Hierarchical Correctness Proofs for Distributed Algorithms*. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1987. (Also appeared as Technical Report MIT/LCS/TR-387, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1987.)
- [UML99] *Unified Modeling Language 1.3*, Technical Report, Rational Software Corporation, 1999.
- [Zie87] W. Zielonka, Notes on Finite Asynchronous Automata. *RAIRO Informatique Théoretique et Applications* 21 (1987), 99 – 135.

List of Figures

1.1	A user in front of a coffee vending machine.	14
3.1	Automaton W_1	32
3.2	Automata \mathcal{A} and \mathcal{A}'	34
3.3	Automata \mathcal{A} and $\mathcal{A}_T^{\{a\}}$	38
4.1	Synchronized automata $\mathcal{T}_{\{1,2\}}$ and $\mathcal{T}'_{\{1,2\}}$	62
4.2	State-reduced synchronized automaton $\hat{\mathcal{T}}_S$	63
4.3	Subautomaton $SUB_{\{j \in [n] j \text{ is odd}\}}$ of synchronized automaton \mathcal{T}	65
4.4	Subautomaton $SUB_{\{1\}}(\mathcal{T}_{\{1,2\}})$ and automaton $(SUB_{\{3,4\}}(\hat{\mathcal{T}}))_S$	65
4.5	Automata \mathcal{A}_1 and \mathcal{A}_2 , and synchronized automaton \mathcal{T}	67
4.6	Automata \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3 , and synchronized automaton \mathcal{T}	68
4.7	Automata \mathcal{A}_1 and \mathcal{A}_2 , and synchronized automaton \mathcal{T}	73
4.8	Synchronized automaton \mathcal{T}'	74
4.9	Three synchronized automata constructed from $\{A_i \mid i \in [7]\}$	75
4.10	Automata \mathcal{A}_1 and \mathcal{A}_2	87
4.11	Automata \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3	92
4.12	Automata \mathcal{A}_1 and \mathcal{A}_2 , and synchronized automaton \mathcal{T}	97
4.13	Synchronized automata \mathcal{T}^{free} and \mathcal{T}^{si}	98
4.14	Automata \mathcal{A}_1 and \mathcal{A}_2	100
4.15	Synchronized automata \mathcal{T}^{free} and \mathcal{T}^{si}	101
4.16	Automata \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3	102
4.17	Synchronized automaton \mathcal{T} and its subautomaton $SUB_{\{1,2\}}$	102
4.18	Automata \mathcal{A}_1 and \mathcal{A}_2	104
4.19	Synchronized automaton \mathcal{T} and its state-reduced version \mathcal{T}_S	105
5.1	Component automaton \mathcal{C}	117
5.2	Component automaton \mathcal{A}	118
5.3	Team automaton \mathcal{T} over $\{\mathcal{C}, \mathcal{A}\}$	121
5.4	A team automaton \mathcal{T} with its subteams $SUB_{a,inp}$ and $SUB_{a,out}$	128
5.5	A team automaton \mathcal{T} with a <i>sipp/wipp</i> action a	130
5.6	A team automaton \mathcal{T} with a <i>sopp/wopp</i> action a	131

5.7	A team automaton \mathcal{T} with a <i>ms/sms/wms</i> action a	133
5.8	Component automata \mathcal{C}_1 , \mathcal{C}_2 , and \mathcal{C}_3	134
5.9	Team automata \mathcal{T} and \mathcal{T}'	136
5.10	Component automata \mathcal{C}_1 and \mathcal{C}_2 , and team automaton \mathcal{T}	139
5.11	Component automata \mathcal{C}_1 and \mathcal{C}_2 , and team automaton \mathcal{T}	146
5.12	Team automata \mathcal{T}^1 and \mathcal{T}^2	148
5.13	Component automata \mathcal{C}_1 and \mathcal{C}_2	155
5.14	Team automata \mathcal{T} and \mathcal{T}'	156
5.15	Component automata \mathcal{C}_1 and \mathcal{C}_2 , and team automaton \mathcal{T}	157
6.1	Extracting behavior from team automata to component automata.	167
6.2	Component automata \mathcal{C}_1 and \mathcal{C}_2	168
6.3	Team automata \mathcal{T} and \mathcal{T}'	168
6.4	Team automaton \mathcal{T}'' and <i>maximal-ai</i> team automaton \mathcal{T}^{ai}	173
6.5	Component automata \mathcal{C} and \mathcal{C}' , and <i>maximal-free</i> team automaton \mathcal{T}^{free}	177
6.6	Team automata \mathcal{T}^{free} and \mathcal{T}^{fa}	180
6.7	Sketch of tree $G = (\bigcup_{n \geq 0} V_n, E)$	203
7.1	Team automaton \mathcal{T}' over $\{\mathcal{C}, \mathcal{A}, \mathcal{A}'\}$	242
7.2	Vector team automata \mathcal{T}_1^v and \mathcal{T}_2^v	247
7.3	Subteam $SUB_{\{2,3\}}(\mathcal{T}_1^v)$ of vector team automaton \mathcal{T}_1^v	247
7.4	Vector team automaton $\mathcal{T}_{\{1,2\}}^v$	248
7.5	Component automata \mathcal{C}_1 and \mathcal{C}_2 , vector team automaton \mathcal{T}^v , and its flattened version \mathcal{T}_F^v	249
7.6	3-ITNC \mathcal{K}	258
7.7	Sketch of the construction of $PN(\mathcal{T}^v)$	260
7.8	$PN(\mathcal{T}_2^v)$	262
7.9	ITNC $PN(\mathcal{T}_{\{1,2\}}^v)$	263
7.10	Component automata \mathcal{C}_1 and \mathcal{C}_2	264
7.11	Vector team automata \mathcal{T}_1^v and \mathcal{T}_2^v	264
7.12	ITNC $PN(\mathcal{T}_1^v)$	265
7.13	ITNC $PN(\mathcal{T}_2^v)$	267
7.14	Sketch of the idea underlying the simulation.	267
7.15	ITNC $SUB_{\{1\}}(PN(\mathcal{T}_2^v))$	271
7.16	Subteam $SUB_{\{1\}}(\mathcal{T}_2^v)$	272
7.17	VLITNs $\text{und}(SUB_{\{1\}}(PN(\mathcal{T}_{\{1,2\}}^v)))$ and $\text{und}(SUB_{\{2\}}(PN(\mathcal{T}_{\{1,2\}}^v)))$	273
7.18	Sketch of iteratively composing ITNCs.	275
8.1	The GROVE document editor architecture.	281
8.2	The departments of a bank.	284

8.3	A package is added.	285
8.4	Hierarchical teams.	287
8.5	Merging teams.	288
8.6	Component automata \mathcal{T}_2 and \mathcal{T}_3	290
8.7	State-reduced team automaton $(\mathcal{T}_{2,3})_S$ over $\{\mathcal{T}_2, \mathcal{T}_3\}$	290
8.8	A team automaton \mathcal{T} over $\mathcal{T}_1, \mathcal{T}_{2,3}$, and \mathcal{T}_4	290
8.9	A rooms metaphor for access control.	294
8.10	Component automata \mathcal{C}^C , \mathcal{C}^B , and \mathcal{C}^A : rooms C , B , and A	295
8.11	State-reduced team automaton \mathcal{T}_S^{CBA} over $\{\mathcal{C}^C, \mathcal{C}^B, \mathcal{C}^A\}$	296
8.12	Component automaton \mathcal{C}^U : user Kwaku.	298
8.13	Team automaton \mathcal{T}_S over $\{\mathcal{T}^{CBA}, \mathcal{C}^U\}$	300
8.14	Component automaton \mathcal{C}^0 : the access building.	302
8.15	Component automaton \mathcal{C}^k : meta access at layer k	304
8.16	State-reduced team automaton $(\mathcal{T}_{k-1}^k)_S$ over \mathcal{C}^{k-1} and \mathcal{C}^k	305

List of Symbols

2. Preliminaries

\subseteq	set inclusion, 23
\subset	proper set inclusion, 23
\setminus	set difference, 23
$\#$	cardinality (of a set), 23
\emptyset	the empty set, 23
$[n]$	shorthand for $\{1, 2, \dots, n\}$, 23
\mathbb{N}	set of positive integers, 23
\prod	cartesian product (prefix notation), 23
\times	cartesian product (infix notation), 23
proj_j	projection on element j , 23
proj_J	projection on subset J , 23
$\text{proj}_j^{[2]}$	shorthand for $\text{proj}_j \times \text{proj}_j$, 24
$\text{proj}_J^{[2]}$	shorthand for $\text{proj}_J \times \text{proj}_J$, 24
$f \upharpoonright C$	restriction of function f to a subset C of its domain, 24
Σ	alphabet, 24
λ	the empty word, 24
$ w $	length (of a word w), 24
$w(i)$	i -th letter (of a word w), 24
$\#_a(w)$	total number of occurrences of letter a (in a word w), 24
$\text{alph}(w)$	alphabet (of a word w), 25
Σ^*	set of all finite words over Σ , 25
Σ^+	set of all nonempty finite words over Σ , 25
Σ^ω	set of all infinite words over Σ , 25
Σ^∞	set of all words over Σ , 25
$u \cdot v$	concatenation (of words u and v), 25
$K \cdot L$	concatenation (of languages K and L), 25
$\text{pref}(w)$	set of prefixes (of a word w), 26
$w[n]$	prefix of length n (of a word w), 25
$\lim_{n \rightarrow \infty} v_n$	limit (of words $v_1 \leq v_2 \leq \dots$), 26
pres_Γ	function preserving the symbols from Γ (and erasing all other symbols), 27

3. Automata

\mathcal{A}	automaton, 29
Q	set of states (of \mathcal{A}), 29
Σ	set of actions or alphabet (of \mathcal{A}), 29
δ	set of labeled transitions (of \mathcal{A}), 29
I	set of initial states (of \mathcal{A}), 29
δ_a	set of a -transitions (of \mathcal{A}), 30
$C_{\mathcal{A}}$	set of finite computations of \mathcal{A} , 30
$C_{\mathcal{A}}^{\omega}$	set of infinite computations of \mathcal{A} , 30
$C_{\mathcal{A}}^{\infty}$	set of computations of \mathcal{A} , 30
$B_{\mathcal{A}}^{\Theta, \infty}$	Θ -behavior of \mathcal{A} , 31
$B_{\mathcal{A}}^{\Theta}$	finitary Θ -behavior of \mathcal{A} , 31
$B_{\mathcal{A}}^{\Theta, \omega}$	infinitary Θ -behavior of \mathcal{A} , 31
Q_S	set of reachable states (of \mathcal{A}), 36
Σ_A	set of active actions (of \mathcal{A}), 36
δ_T	set of useful transitions (of \mathcal{A}), 36
$\mathcal{A}_1 \sqsubseteq \mathcal{A}_2$	containment (of \mathcal{A}_1 in \mathcal{A}_2), 36
\mathcal{A}_A^{Θ}	Θ -action-reduced version of \mathcal{A} , 37
\mathcal{A}_T^{Θ}	Θ -transition-reduced version of \mathcal{A} , 38
\mathcal{A}_S	state-reduced version of \mathcal{A} , 46
\mathcal{A}_A	action-reduced version of \mathcal{A} , 50
\mathcal{A}_T	transition-reduced version of \mathcal{A} , 50
\mathcal{A}_R	reduced version of \mathcal{A} , 50

4. Synchronized Automata

\mathcal{I}	index set, 59
\mathcal{A}_i	automaton, 59
\mathcal{S}	set of automata, 59
$\Delta_a(\mathcal{S})$	complete transition space of a in \mathcal{S} , 60
\mathcal{T}	synchronized automaton, 60
$SUB_J(\mathcal{T})$	the subautomaton of \mathcal{T} determined by J , 64
SUB_J	the subautomaton (of \mathcal{T}) determined by J , 64
$\pi_{\mathcal{A}_j}$	projection on automaton \mathcal{A}_j , 70
π_{SUB_J}	projection on subautomaton SUB_J , 70
\mathcal{D}	indexed set, 76
$\mathcal{V}(\mathcal{D})$	all finitely nested cartesian products of sets from \mathcal{D} , 76
$\text{dom}(V)$	domain of an element V , 76
u_V	function unpacking elements v from V , 77

$\langle v \rangle_V$	reordering of an element $v \in V$ relative to the construction of V , 77
$\langle\langle \mathcal{T} \rangle\rangle_S$	reordered version of synchronized automaton \mathcal{T} (w.r.t. \mathcal{S}), 81
\mathcal{T}	synchronized automaton, 84
$Free(\mathcal{T})$	set of <i>free</i> actions of \mathcal{T} , 85
$AI(\mathcal{T})$	set of <i>ai</i> actions of \mathcal{T} , 85
$SI(\mathcal{T})$	set of <i>si</i> actions of \mathcal{T} , 86
$\mathcal{R}_a^{no}(\mathcal{S})$	predicate <i>no-constraints</i> , 88
$\mathcal{R}_a^{free}(\mathcal{S})$	predicate <i>is-free</i> for a in \mathcal{S} , 88
$\mathcal{R}_a^{ai}(\mathcal{S})$	predicate <i>is-ai</i> for a in \mathcal{S} , 89
$\mathcal{R}_a^{si}(\mathcal{S})$	predicate <i>is-si</i> for a in \mathcal{S} , 89
j	element of \mathcal{I} , 90
J	subset of \mathcal{I} , 90
\emptyset	arbitrary alphabet disjoint from set Q of states (of \mathcal{T}), 90

5. Team Automata

\mathcal{C}	component automaton, 116
Σ_{inp}	set of input actions or input alphabet (of \mathcal{C}), 116
Σ_{out}	set of output actions or output alphabet (of \mathcal{C}), 116
Σ_{int}	set of internal actions or internal alphabet (of \mathcal{C}), 116
$und(\mathcal{C})$	underlying automaton of \mathcal{C} , 116
Σ	set of actions or (full) alphabet (of \mathcal{C}), 116
Σ_{ext}	set of external actions or external alphabet (of \mathcal{C}), 116
Σ_{loc}	set of locally-controlled actions or locally-controlled alphabet (of \mathcal{C}), 117
$\mathbf{B}_C^{\Sigma_{inp}, \infty}$	input behavior (of \mathcal{C}), 117
$\mathbf{B}_C^{\Sigma_{out}, \infty}$	output behavior (of \mathcal{C}), 117
$\mathbf{B}_C^{\Sigma_{int}, \infty}$	internal behavior (of \mathcal{C}), 117
$\mathbf{B}_C^{\Sigma_{ext}, \infty}$	external behavior (of \mathcal{C}), 117
$\mathbf{B}_C^{\Sigma_{loc}, \infty}$	locally-controlled behavior (of \mathcal{C}), 117
\mathcal{I}	index set, 118
\mathcal{C}_i	component automaton, 118
Σ_i	set of actions (of \mathcal{C}_i), 118
\mathcal{S}	set of component automata, 118
\mathcal{S}	composable system, 118
\mathcal{T}	team automaton, 120
$und(\mathcal{T})$	underlying synchronized automaton of \mathcal{T} , 120
$SUB_J(\mathcal{T})$	the subteam of \mathcal{T} determined by J , 122
SUB_J	the subteam (of \mathcal{T}) determined by J , 122

\mathcal{S}	composable system, 123
$\langle\langle\mathcal{T}\rangle\rangle_{\mathcal{S}}$	reordered version of team automaton \mathcal{T} w.r.t. \mathcal{S} , 125
\mathcal{T}	team automaton, 126
Σ_{inp}	set of input actions (of \mathcal{T}), 126
Σ_{out}	set of output actions (of \mathcal{T}), 126
Σ_{int}	set of internal actions (of \mathcal{T}), 126
Σ	set of actions (of \mathcal{T}), 126
Σ_{ext}	set of external actions (of \mathcal{T}), 126
Σ_{loc}	set of locally-controlled actions (of \mathcal{T}), 126
$\mathcal{I}_{a,inp}(\mathcal{S})$	input domain of a in \mathcal{S} , 126
$\mathcal{I}_{a,out}(\mathcal{S})$	output domain of a in \mathcal{S} , 126
$\mathcal{I}_{a,inp}$	input domain of a (in \mathcal{S}), 127
$\mathcal{I}_{a,out}$	output domain of a (in \mathcal{S}), 127
$SUB_{a,inp}(\mathcal{T})$	input subteam of a in \mathcal{T} , 127
$SUB_{a,out}(\mathcal{T})$	output subteam of a in \mathcal{T} , 127
$SUB_{a,inp}$	input subteam of a (in \mathcal{T}), 127
$SUB_{a,out}$	output subteam of a (in \mathcal{T}), 127
$SIPP(\mathcal{T})$	set of <i>sipp</i> actions of \mathcal{T} , 129
$WIPP(\mathcal{T})$	set of <i>wipp</i> actions of \mathcal{T} , 129
$SOPP(\mathcal{T})$	set of <i>sopp</i> actions of \mathcal{T} , 129
$WOPP(\mathcal{T})$	set of <i>wopp</i> actions of \mathcal{T} , 129
$MS(\mathcal{T})$	set of <i>ms</i> actions of \mathcal{T} , 131
$SMS(\mathcal{T})$	set of <i>sms</i> actions of \mathcal{T} , 131
$WMS(\mathcal{T})$	set of <i>wms</i> actions of \mathcal{T} , 132
$\mathcal{I}_{a,inp}$	input domain of a (in \mathcal{S}), 141
$\mathcal{I}_{a,out}$	output domain of a (in \mathcal{S}), 141
$\mathcal{R}_a^{sipp}(\mathcal{S})$	predicate <i>is-sipp</i> for a in \mathcal{S} , 141
$\mathcal{R}_a^{wipp}(\mathcal{S})$	predicate <i>is-wipp</i> for a in \mathcal{S} , 141
$\mathcal{R}_a^{sopp}(\mathcal{S})$	predicate <i>is-sopp</i> for a in \mathcal{S} , 142
$\mathcal{R}_a^{wopp}(\mathcal{S})$	predicate <i>is-wopp</i> for a in \mathcal{S} , 142
$\mathcal{R}_a^{ms}(\mathcal{S})$	predicate <i>is-ms</i> for a in \mathcal{S} , 144
$\mathcal{R}_a^{sms}(\mathcal{S})$	predicate <i>is-sms</i> for a in \mathcal{S} , 144
$\mathcal{R}_a^{wms}(\mathcal{S})$	predicate <i>is-wms</i> for a in \mathcal{S} , 144
$\Sigma_{i,ext}$	set of external actions (of \mathcal{C}_i), 150
$\Sigma_{i,loc}$	set of locally-controlled actions (of \mathcal{C}_i), 150
j	element of \mathcal{I} , 150
J	subset of \mathcal{I} , 150
$\Sigma_{J,ext}$	set of external actions (of SUB_J), 150
$\Sigma_{J,loc}$	set of locally-controlled actions (of SUB_J), 150

6. Behavior of Team Automata

pREG family of prefix-closed regular finitary languages, 164

REG	family of regular languages, 164
FIN	family of finite languages, 164
CA	$\{\mathbf{B}_{\mathcal{C}}^{\Sigma} \mid \mathcal{C} \text{ is a finite component automaton with alphabet } \Sigma\}$, 164
CA^{alph}	$\{\mathbf{B}_{\mathcal{C}}^{alph} \mid \mathcal{C} \text{ is a finite component automaton}\}$ (with $alph \in \{inp, out, int, ext, loc\}$), 165
\mathcal{I}	index set, 166
\mathcal{C}_i	component automaton, 166
Σ_i	set of actions (of \mathcal{C}_i), 166
\mathcal{S}	composable system, 166
\mathcal{T}	team automaton, 166
Σ	set of actions (of \mathcal{T}), 166
Θ	arbitrary alphabet disjoint from set Q of states (of \mathcal{T}), 166
j	element of \mathcal{I} , 166
$uAI_j(\mathcal{T})$	set of useful j - <i>ai</i> actions (of \mathcal{T}), 169
\parallel	shuffle, 183
$\parallel\parallel$	fair shuffle, 183
$\parallel d \parallel$	norm (of decomposition d), 198
$\parallel\parallel_{i \in [n]}$	n -ary fair shuffle, 205
$\parallel_{i \in [n]}$	n -ary shuffle, 205
\parallel^{Γ}	S-shuffle on Γ , 207
$\parallel\parallel^{\Gamma}$	fair S-shuffle on Γ , 207
$\text{alph}(L)$	alphabet (of a language L), 208
$\Sigma_1 \parallel \Sigma_2$	fS-shuffle w.r.t. Σ_1 and Σ_2 , 208
$\Sigma_1 \parallel\parallel^{\Sigma_2}$	fair fS-shuffle w.r.t. Σ_1 and Σ_2 , 208
$\Sigma_1 \parallel\parallel^{\Gamma \Sigma_2}$	rS-shuffle on Γ w.r.t. Σ_1 and Σ_2 , 209
$\Sigma_1 \parallel\parallel^{\Gamma}_{\Sigma_2}$	fair rS-shuffle on Γ w.r.t. Σ_1 and Σ_2 , 209
$\parallel\parallel_{i \in [n]}^{\Gamma}$	n -ary fair S-shuffle on Γ , 227
$\parallel_{i \in [n]}^{\Gamma}$	n -ary S-shuffle on Γ , 227
$\parallel\parallel_{\cup_{i \in [n]} \Sigma_i}$	n -ary fair fS-shuffle w.r.t. $\cup_{i \in [n]} \Sigma_i$, 228
$\parallel_{\cup_{i \in [n]} \Sigma_i}$	n -ary fS-shuffle w.r.t. $\cup_{i \in [n]} \Sigma_i$, 228
$\parallel\parallel_{\cup_{i \in [n]} \Sigma_i}^{\Gamma}$	n -ary fair rS-shuffle on Γ w.r.t. $\cup_{i \in [n]} \Sigma_i$, 228
$\parallel_{\cup_{i \in [n]} \Sigma_i}^{\Gamma}$	n -ary rS-shuffle on Γ w.r.t. $\cup_{i \in [n]} \Sigma_i$, 228

7. Team Automata, I/O Automata, Petri Nets

\mathcal{I}	index set, 233
\mathcal{C}_i	component automaton, 233
Σ_i	set of actions (of \mathcal{C}_i), 233
\mathcal{S}	composable system, 233
\mathcal{T}	team automaton, 233

Σ	set of actions (of \mathcal{T}), 233
Σ_{ext}	set of external actions (of \mathcal{T}), 233
Σ_{loc}	set of locally-controlled actions (of \mathcal{T}), 233
Θ	arbitrary alphabet disjoint from set Q of states (of \mathcal{T}), 233
S	compatible system, 237
\mathcal{T}	team I/O automaton, 239
IOCA	$\{\mathbf{B}_C^\Gamma \mid \Gamma \text{ is an alphabet and } C \text{ is a finite input-enabling component automaton with alphabet } \Gamma\}$, 240
IOCA ^{alph}	$\{\mathbf{B}_C^{alph} \mid C \text{ is a finite input-enabling component automaton}\}$ (with $alph \in \{inp, out, int, ext, loc\}$), 240
$\Delta_a^v(S)$	complete vector transition space (of a in S), 245
\underline{a}	vector action a , 245
\mathcal{T}^v	vector team automaton, 245
δ^v	set of labeled vector transitions (of \mathcal{T}^v), 245
$\delta_{\underline{a}}^v$	set of vector \underline{a} -transitions (of \mathcal{T}^v), 245
$SUB_J(\mathcal{T}^v)$	the subteam of \mathcal{T}^v determined by J , 246
\mathcal{T}_F^v	the flattened version (of \mathcal{T}^v), 247
$tFree(\mathcal{T}^v)$	set of truly <i>free</i> actions (of \mathcal{T}^v), 250
$tAI(\mathcal{T}^v)$	set of truly <i>ai</i> actions (of \mathcal{T}^v), 250
$tSI(\mathcal{T}^v)$	set of truly <i>si</i> actions (of \mathcal{T}^v), 250
Λ	empty word vector, 252
$\text{tot}(\{\Delta_j \mid j \in J\})$	total vector alphabet (over $\{\Delta_j \mid j \in J\}$), 252
Δ^u	subset of uniform vector letters of vector alphabet Δ , 252
$v \circ w$	component-wise concatenation (of two n -dimensional vector letters v and w), 252
coll	collapse of a sequence of vector letters into a word vector, 252
$\text{und}(\mathcal{T}^v)$	underlying vector automaton (of \mathcal{T}^v), 253
$\mathbf{V}_{\mathcal{T}^v}$	finitary vector behavior (of \mathcal{T}^v), 253
$\mathbf{V}_{\mathcal{T}^v}^\omega$	infinitary vector behavior (of \mathcal{T}^v), 253
$\mathbf{V}_{\mathcal{T}^v}^\infty$	vector behavior (of \mathcal{T}^v), 253
\mathcal{N}	n -VLITN, 254
P	finite set of places (of \mathcal{N}), 254
T	finite set of events (of \mathcal{N}), 254
O	finite set of n integers, called tokens (of \mathcal{N}), 254
F	flow function (of \mathcal{N}), 254
V	vector alphabet of vector labels (of \mathcal{N}), 255
ℓ	event labeling homomorphism (of \mathcal{N}), 255
$\text{use}(t)$	set of tokens used (by event t), 255
$\mathbf{M}_{\mathcal{N}}$	set of all markings of \mathcal{N} , 255
$\mu[t]_{\mathcal{N}}$	enabled (an event t of \mathcal{N} at a marking μ of \mathcal{N}), 256
$\mu[t]_{\mathcal{N}\nu}$	fires (an event t of \mathcal{N} from a marking μ of \mathcal{N} to a marking ν of \mathcal{N}), 256

$\mu_0[t_1 t_2 \cdots t_m]_{\mathcal{N}}$	firing sequence (of events t_1, t_2, \dots, t_m) of \mathcal{N} starting from μ_0 , 256
$\mu_0[t_1 t_2 \cdots t_m]_{\mathcal{N}} \mu_m$	firing sequence (of events t_1, t_2, \dots, t_m) of \mathcal{N} starting from μ_0 and leading to μ_m , 256
$\mu_0[t_1 t_2 \cdots]_{\mathcal{N}}$	infinite firing sequence (of events t_1, t_2, \dots) of \mathcal{N} starting from μ_0 , 256
\mathcal{K}	n -ITNC, 256
$\text{und}(\mathcal{K})$	underlying n -VLITN (of \mathcal{K}), 256
\mathcal{M}_0	set of initial markings (of \mathcal{K}), 256
\mathcal{M}_f	set of final markings (of \mathcal{K}), 256
$\mathbf{FS}_{\mathcal{K}}$	set of all firing sequences (of \mathcal{K}), 257
$\mathbf{M}_{\mathcal{K}}$	the set of all reachable markings (of \mathcal{K}), 257
$\mathbf{B}_{\mathcal{K}}$	behavior of \mathcal{K} , 257
$\mathbf{V}_{\mathcal{K}}$	vector behavior of \mathcal{K} , 257
carrier (\underline{a})	carrier (of \underline{a}), 260
$PN(\mathcal{T}^v)$	ITNC obtained from \mathcal{T}^v , 261
$SUB_J(\mathcal{K})$	the subnet (of \mathcal{K}) determined by J , 270

8. Applying Team Automata

\mathcal{I}	index set, 278
\mathcal{C}_i	component automaton, 278
$\Sigma_{i,ext}$	set of external actions (of \mathcal{C}_i), 278
\mathcal{S}	composable system, 278
\mathcal{T}	team automaton, 278
Σ	set of actions (of \mathcal{T}), 278
Σ_{ext}	set of external actions (of \mathcal{T}), 278
\mathcal{C}_H^Δ	the Δ -hiding version (of \mathcal{C}), 278
Σ_{com}	set of communicating actions (in \mathcal{S}), 279
$\boxed{\mathcal{T}}$	(communication) closed version (of \mathcal{T}), 279
\mathcal{C}_N^h	h -renamed version (of \mathcal{C}), 280

Index

- a*-transition, 30
- access control, 292
 - distributed, 306
 - meta, 301
 - spatial, 291
- action, 29, 117
 - action-indispensable, 85
 - active, 35
 - ai*, 85
 - truly, 250
 - communicating, 279
 - complementary, 17
 - enabled, 51
 - external, 117
 - free*, 85
 - truly, 250
 - input, 116
 - input peer-to-peer
 - strong, 129
 - weak, 129
 - internal, 116
 - locally-controlled, 117
 - master-slave, 131
 - strong, 131
 - weak, 132
 - maximal-free*, 89
 - maximal-ms*, 147
 - maximal-sipp*, 147
 - maximal-ai*, 89
 - maximal-sms*, 147
 - maximal-sopp*, 147
 - maximal-wipp*, 147
 - maximal-si*, 89
 - maximal-wms*, 147
 - maximal-wopp*, 147
 - ms*, 131
 - output, 116
 - output peer-to-peer
 - strong, 129
 - weak, 129
 - si*, 86
 - truly, 250
 - silent, 17
 - sipp*, 129
 - sms*, 131
 - sopp*, 129
 - state-indispensable, 86
 - useful *j*-action-indispensable, 169
 - vector, 17, 244, 245
 - wipp*, 129
 - wms*, 132
 - wopp*, 129
- active collaboration, 161
- alphabet, 24
 - external, 116
 - (full), 116
 - input, 116
 - internal, 116
 - locally-controlled, 117
 - output, 116
 - vector, 252
 - n*-dimensional, 252
 - total, 252
- alphabetized parallel composition, 206
- automaton, 29
 - action-reduced version of, 50
 - component, *see* component automaton
 - cooperating, 18
 - timed, 18
 - cooperating pushdown, 18
 - finite (state), 29
 - finite asynchronous, 257
 - I/O, *see* I/O automaton

- Input/Output, *see* I/O automaton
 - product, 17
 - reduced version of, 50
 - set of, *see* set of automata
 - state-reduced version of, 46
 - synchronized, *see* synchronized automaton
 - team, *see* team automaton
 - Θ -action-reduced version of, 37
 - Θ -deterministic, 55
 - Θ -enabling, 51
 - Θ -transition-reduced version of, 38
 - transition-reduced version of, 50
 - trivial, 30
- behavior, 31, 117, 253, 257
 - external, 117
 - finitary, 31, 117, 253
 - infinitary, 31, 117, 253
 - input, 117
 - internal, 117
 - locally-controlled, 117
 - output, 117
 - vector, 253, 257
 - finitary, 253
 - infinitary, 253
- bijection, 24
- Calculus of Communicating Systems, 17
- cardinality, 23
- carrier, 260
- cartesian product, 23
- CCS, 17
- coding, 27
 - weak, 27
- collapse, 253
- communicating relation, 279
- Communicating Sequential Processes, 18
 - Theoretical, 18
- compatible system, 234
- complete transition space, 60
- complete vector transaction space, 245
- component automaton, 116
 - communicating, 279
 - (communication) closed version of, 279
 - Δ -hiding version of, 278
 - finite, 116
 - h -renamed version of, 280
 - Θ -deterministic, 150
 - Θ -enabling, 150
 - trivial, 116
 - underlying automaton of, 116
- composable system, 118
 - ai*-consistent, 176
- compositionality, 163
- computation, 30, 117, 253
 - finite, 30, 117, 253
 - infinite, 30, 117, 253
 - trivial, 30
- Computer Supported Cooperative Work, 12
- concatenation, 25
 - component-wise, 252
- concurrent composition of synchronizing processes, 206
- COncurrent SYstem, 17
- contained in, 36
- cooperation strategy, 283
 - conflict-free, 284
 - conservative, 283
 - optimistic, 283
- COSY, 17
- CSCW, 12
- CSP, 18
- decomposition, 198
 - norm of, 198
- domain, 76
 - input, 126
 - output, 126
- empty set, 23
- event, 254
 - enabled, 256
 - independent, 257
- family of languages, 25
- fire, 256
- firing sequence, 256, 257
 - infinite, 256
 - vector, 17
- formal methods, 12
- fS-shuffle, 208, 209
 - fair, 208, 209

- n -ary, 228
- n -ary, 228
- function, 24
 - flow, 254
 - injective, 24
 - restriction of, 24
 - surjective, 24
- groupware, 12
- handshake communication, 17
- homomorphism, 26
 - erasing, 26
 - event labeling, 255
- I/O automaton, 234
 - safe, 234
 - team, 235
 - iterated, 237
 - unfair, 234
- I/O system, 17
- index set, 59, 118, 166, 233, 278
- Individual Token Net Controller, 254
 - n -dimensional, 256
- input enabling, 234
- interacting state machines, 17
- ITNC, 254, 256
 - underlying VLITN of, 256
- König's Lemma, 202
- language, 25
 - alphabet of, 208
 - finitary, 25
 - infinitary, 25
 - limit-closed, 202
 - prefix-closed, 26
 - vector, 252
 - n -dimensional, 252
- limit, 26
- loop, 30
- marking, 255
 - complete, 256
 - final, 256
 - initial, 256
 - reachable, 257
- n -ITNC, 256
 - underlying n -VLITN of, 256
- n -VLITN, 254
- ω -language, 25
- ω -word, 24
- partition, 23
- passive cooperation, 161
- path expression, 17
- Petri net, 243
- place, 254
- precedes, 198
 - directly, 198
- predicate (of synchronizations), 88
 - is-ai*, 89
 - is-free*, 88
 - is-ms*, 144
 - is-st*, 89
 - is-sipp*, 141
 - is-sms*, 144
 - is-sopp*, 142
 - is-wipp*, 141
 - is-wms*, 144
 - is-wopp*, 142
 - no-constraints*, 88
- prefix, 25
 - finite, 25
- product
 - free, 17
 - mixed, 17
 - synchronous, 17
- produit de mixage, 206
- projection
 - on automaton \mathcal{A}_j , 70
 - on subautomaton SUB_J , 70
- \mathcal{R} -synchronized automaton, 88
- \mathcal{R} -team automaton, 140
- record, 31, 117
 - external, 117
 - input, 117
 - internal, 117
 - locally-controlled, 117
 - output, 117
- renegotiation phase, 286
- reordering, 77
- revocation
 - deep, 303
 - delayed, 298

- immediate, 298
 - shallow, 303
- rS-shuffle, 209
 - fair, 209
 - n -ary, 228
 - n -ary, 228
- S-shuffle, 207
 - fair, 207
 - n -ary, 227
 - n -ary, 227
- set difference, 23
- set inclusion, 23
 - proper, 23
- set of automata, 59
 - state-reduced, 104
 - Θ -action-reduced, 104
 - Θ -deterministic, 104
 - Θ -enabling, 93
 - Θ - J -loop-limited, 94
 - Θ - j -loop-limited, 94
 - Θ -loop-limited, 106
 - Θ -transition-reduced, 104
- shuffle, 182, 183
 - fair, 183
 - n -ary, 205
 - n -ary, 205
 - synchronized, *see* S-shuffle
 - fully, *see* fS-shuffle
 - relaxed, *see* rS-shuffle
- software configuration management, 283
- software engineering, 283
- state, 29
 - initial, 29
 - irregular, 302
 - reachable, 35
- state machine decomposable net, 259
- state space
 - finite, 257
- statecharts, 18
- subnet, 270
- synchronization
 - pluriform, 17
 - uniform, 17
- synchronization, 60
- synchronized automaton, 60
 - iterated, 79
 - reordered version of, 81
 - maximal-ai*, 89
 - maximal-free*, 89
 - maximal-si*, 89
 - subautomaton of, 64
- synchronized shuffle, 206
- system, 11
 - compatible, *see* compatible system
 - composable, *see* composable system
 - distributed, 11
 - groupware, 12
 - I/O, *see* I/O system
 - reactive, 11
 - transformational, 11
 - transition, *see* transition system
- TCSP, 18
- team automaton, 120
 - collaborating, 160
 - (communication) closed version of, 279
 - cooperating, 160
 - Δ -hiding version of, 278
 - h -renamed version of, 280
 - heterogeneous, 147
 - homogeneous, 147
 - iterated, 123
 - reordered version of, 125
 - maximal-ai*, 141
 - maximal-free*, 141
 - maximal-ms*, 147
 - maximal-si*, 141
 - maximal-sipp*, 147
 - maximal-sms*, 147
 - maximal-sopp*, 147
 - maximal-wipp*, 147
 - maximal-wms*, 147
 - maximal-wopp*, 147
 - subteam of, 122
 - input, 127
 - output, 127
 - underlying synchronized automaton of, 120
- vector, 245
 - flattened version of, 247
 - non-state-sharing, 266
 - subteam of, 245
 - underlying vector automaton of, 253

- Θ -behavior, 31, 117
 - finitary, 31, 117
 - infinitary, 31, 117
- Θ -record, 31, 117
- token, 254
- trace theory, 257
- transition, 30
 - clone, 268
 - incoming, 30
 - labeled, 29
 - omnipresent, 90
 - outgoing, 30
 - present, 90
 - useful, 35
 - vector, 244
 - (labeled), 245
- transition system, 13
 - labeled, 13
 - reactive, 17
- unpack, 77
- VCCS, 17, 252
- vector (of computations), 23
 - ai*-consistent, 174
 - n*-dimensional, 23
 - used, 172
- word, *see* word vector
- Vector Controlled Concurrent System, 17, 252
- vector label, 255
- Vector Labeled Individual Token Net, 254
 - n*-dimensional, 254
- vector letter, 252
 - n*-dimensional, 252
 - uniform, 252
- vector representation, 245
- VLITN, 254
 - 1-throughput, 255
 - label-consistent, 255
- weave, 206
- word, 24
 - alphabet of, 24
 - empty, 24
 - finite, 24
 - infinite, 24
 - length of, 24
- word vector, 252
 - empty, 252
 - n*-dimensional, 252
 - n*-dimensional, 252

