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


[Lav00] H. Lavana, A Universally Configurable Architecture for Taskflow-Oriented Design of a Distributed Collaborative Computing Environ-


List of Figures

1.1 A user in front of a coffee vending machine. .......................... 14
3.1 Automaton $W_1$. .......................................................... 32
3.2 Automata $A$ and $A'$. .................................................. 34
3.3 Automata $A$ and $A_i^{(a)}$. ........................................... 38
4.1 Synchronized automata $T_{[1,2]}$ and $T'_{[1,2]}$. ....................... 62
4.2 State-reduced synchronized automaton $\hat{T}_S$. ......................... 63
4.3 Subautomaton $SUB_{\{j\mid j \text{ is odd}\}}$ of synchronized automaton $T$. ................................. 65
4.4 Subautomaton $SUB_{\{1\}}(T_{[1,2]}$) and automaton $(SUB_{\{3,4\}}(T))_S$ ........ 65
4.5 Automata $A_1$ and $A_2$, and synchronized automaton $T$. ........... 67
4.6 Automata $A_1$, $A_2$, and $A_3$, and synchronized automaton $T$. .... 68
4.7 Automata $A_1$ and $A_2$, and synchronized automaton $T$. ........... 73
4.8 Synchronized automaton $T'$. ........................................... 74
4.9 Three synchronized automata constructed from $\{A_i \mid i \in [7]\}$. ...... 75
4.10 Automata $A_1$ and $A_2$. .................................................. 87
4.11 Automata $A_1$, $A_2$, and $A_3$. ....................................... 92
4.12 Automata $A_1$ and $A_2$, and synchronized automaton $T$. ........... 97
4.13 Synchronized automata $T_{free}$ and $T_{si}$. .......................... 98
4.14 Automata $A_1$ and $A_2$. .................................................. 100
4.15 Synchronized automata $T_{free}$ and $T_{si}$. .......................... 101
4.16 Automata $A_1$, $A_2$, and $A_3$. ....................................... 102
4.17 Synchronized automaton $T$ and its subautomaton $SUB_{[1,2]}$. ....... 102
4.18 Automata $A_1$ and $A_2$. .................................................. 104
4.19 Synchronized automaton $T$ and its state-reduced version $T_S$. ....... 105
5.1 Component automaton $C$. ................................................ 117
5.2 Component automaton $A$. ................................................. 118
5.3 Team automaton $T$ over $\{C,A\}$. .................................... 121
5.4 A team automaton $T$ with its subteams $SUB_{a,inp}$ and $SUB_{a,out}$. 128
5.5 A team automaton $T$ with a $sipp/wipp$ action $a$. ................... 130
5.6 A team automaton $T$ with a $sopp/wopp$ action $a$. ................... 131
5.7 A team automaton $\mathcal{T}$ with a $ms/sms/wms$ action $a$. ...................... 133
5.8 Component automata $C_1$, $C_2$, and $C_3$. ........................... 134
5.9 Team automata $\mathcal{T}$ and $\mathcal{T}'$. ................................ 136
5.10 Component automata $C_1$ and $C_2$, and team automaton $\mathcal{T}$. .... 139
5.11 Component automata $C_1$ and $C_2$, and team automaton $\mathcal{T}$. .... 146
5.12 Team automata $\mathcal{T}_1'$ and $\mathcal{T}_2'$. ............................. 148
5.13 Component automata $C_1$ and $C_2$. ................................ 155
5.14 Team automata $\mathcal{T}$ and $\mathcal{T}'$. ................................ 156
5.15 Component automata $C_1$ and $C_2$, and team automaton $\mathcal{T}$. .... 157

6.1 Extracting behavior from team automata to component automata. 167
6.2 Component automata $C_1$ and $C_2$. ................................ 168
6.3 Team automata $\mathcal{T}$ and $\mathcal{T}'$. ................................ 168
6.4 Team automaton $\mathcal{T}''$ and maximal-aut team automaton $\mathcal{T}^{\det}$. 173
6.5 Component automata $C$ and $C'$, and maximal-free team automaton $\mathcal{T}^{\det}$. 177
6.6 Team automata $\mathcal{T}^{\det}$ and $\mathcal{T}^{\det}$. .......................... 180
6.7 Sketch of tree $G = (\bigcup_{n \geq 0} V_n, E)$. .......................... 203

7.1 Team automaton $\mathcal{T}'$ over $\{C, A, A'\}$. ......................... 242
7.2 Vector team automata $\mathcal{T}_1''$ and $\mathcal{T}_2''$. 247
7.3 Subteam $\text{SUB}_{2,3}(\mathcal{T}_1'')$ of vector team automaton $\mathcal{T}_1''$. 247
7.4 Vector team automaton $\mathcal{T}_{(1,2)}''$. ................................. 248
7.5 Component automata $C_1$ and $C_2$, vector team automaton $\mathcal{T}''$, and its flattened version $\mathcal{T}_F''$. 249
7.6 3-ITNC $\mathcal{K}$. ......................................................... 258
7.7 Sketch of the construction of $PN(\mathcal{T}'')$. .......................... 260
7.8 $PN(\mathcal{T}_F'')$. .................................................. 262
7.9 ITNC $PN(\mathcal{T}_{(1,2)}'')$. ......................................... 263
7.10 Component automata $C_1$ and $C_2$. ................................ 264
7.11 Vector team automata $\mathcal{T}_1''$ and $\mathcal{T}_2''$. ........................ 264
7.12 ITNC $PN(\mathcal{T}_1'')$. ............................................. 265
7.13 ITNC $PN(\mathcal{T}_2'')$. ............................................. 267
7.14 Sketch of the idea underlying the simulation. ......................... 267
7.15 ITNC $\text{SUB}_{(1)}(PN(\mathcal{T}_2''))$. .............................. 271
7.16 Subteam $\text{SUB}_{(1)}(\mathcal{T}_F'')$. ................................ 272
7.17 VLITNs und($\text{SUB}_{(1)}(PN(\mathcal{T}_{(1,2)}''))$) and und($\text{SUB}_{(2)}(PN(\mathcal{T}_{(1,2)}''))$). 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 $T_2$ and $T_3$ ................................. 290
8.7 State-reduced team automaton $(T_{2,3})_S$ over $\{T_2, T_3\}$ ........................................ 290
8.8 A team automaton $T$ over $T_1$, $T_{2,3}$, and $T_4$ ................................. 290
8.9 A rooms metaphor for access control ................................ 294
8.10 Component automata $C^C$, $C^B$, and $C^A$: rooms $C$, $B$, and $A$ .................... 295
8.11 State-reduced team automaton $T^{CBA}_S$ over $\{C^C, C^B, C^A\}$ ......................... 296
8.12 Component automaton $C^U$: user Kwaku ........................................ 298
8.13 Team automaton $T_S$ over $\{T^{CBA}, C^U\}$ ..................................... 300
8.14 Component automaton $C^0$: the access building ..................... 302
8.15 Component automaton $C^k$: meta access at layer $k$ ...................... 304
8.16 State-reduced team automaton $(T^k_{k-1})_S$ over $C^{k-1}$ and $C^k$ ........ 305
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊆</td>
<td>set inclusion, 23</td>
</tr>
<tr>
<td>⊂</td>
<td>proper set inclusion, 23</td>
</tr>
<tr>
<td>\</td>
<td>set difference, 23</td>
</tr>
<tr>
<td>#</td>
<td>cardinality (of a set), 23</td>
</tr>
<tr>
<td>Φ</td>
<td>the empty set, 23</td>
</tr>
<tr>
<td>[n]</td>
<td>shorthand for {1, 2, \ldots, n}, 23</td>
</tr>
<tr>
<td>N</td>
<td>set of positive integers, 23</td>
</tr>
<tr>
<td>Π</td>
<td>cartesian product (prefix notation), 23</td>
</tr>
<tr>
<td>×</td>
<td>cartesian product (infix notation), 23</td>
</tr>
<tr>
<td>proj_j</td>
<td>projection on element j, 23</td>
</tr>
<tr>
<td>proj_J</td>
<td>projection on subset J, 23</td>
</tr>
<tr>
<td>proj_j_2</td>
<td>shorthand for (proj_j \times proj_j), 24</td>
</tr>
<tr>
<td>proj_J_2</td>
<td>shorthand for (proj_J \times proj_J), 24</td>
</tr>
<tr>
<td>(f \upharpoonright C)</td>
<td>restriction of function (f) to a subset (C) of its domain, 24</td>
</tr>
<tr>
<td>(\Sigma)</td>
<td>alphabet, 24</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>the empty word, 24</td>
</tr>
<tr>
<td></td>
<td>length (of a word (w)), 24</td>
</tr>
<tr>
<td>(w(i))</td>
<td>(i)-th letter (of a word (w)), 24</td>
</tr>
<tr>
<td>(#_a(w))</td>
<td>total number of occurrences of letter (a) (in a word (w)), 24</td>
</tr>
<tr>
<td>(\text{alph}(w))</td>
<td>alphabet (of a word (w)), 25</td>
</tr>
<tr>
<td>(\Sigma^*)</td>
<td>set of all finite words over (\Sigma), 25</td>
</tr>
<tr>
<td>(\Sigma^+)</td>
<td>set of all nonempty finite words over (\Sigma), 25</td>
</tr>
<tr>
<td>(\Sigma^\omega)</td>
<td>set of all infinite words over (\Sigma), 25</td>
</tr>
<tr>
<td>(\Sigma^\infty)</td>
<td>set of all words over (\Sigma), 25</td>
</tr>
<tr>
<td>(u \cdot v)</td>
<td>concatenation (of words (u) and (v)), 25</td>
</tr>
<tr>
<td>(K \cdot L)</td>
<td>concatenation (of languages (K) and (L)), 25</td>
</tr>
<tr>
<td>(\text{pref}(w))</td>
<td>set of prefixes (of a word (w)), 26</td>
</tr>
<tr>
<td>(w[n])</td>
<td>prefix of length (n) (of a word (w)), 25</td>
</tr>
<tr>
<td>(\lim\ v_n)</td>
<td>limit (of words (v_1 \leq v_2 \leq \cdots)), 26</td>
</tr>
<tr>
<td>(n \to \infty)</td>
<td>function preserving the symbols from (\Gamma) (and erasing all other symbols), 27</td>
</tr>
</tbody>
</table>
3. Automata

\( A \) automaton, 29
\( Q \) set of states (of \( A \)), 29
\( \Sigma \) set of actions or alphabet (of \( A \)), 29
\( \delta \) set of labeled transitions (of \( A \)), 29
\( I \) set of initial states (of \( A \)), 29
\( \delta_a \) set of \( a \)-transitions (of \( A \)), 30
\( C_\lambda \) set of finite computations of \( A \), 30
\( C_\omega \) set of infinite computations of \( A \), 30
\( C_\infty \) set of computations of \( A \), 30
\( B_\Theta \) \( \Theta \)-behavior of \( A \), 31
\( B_\Theta^f \) finitary \( \Theta \)-behavior of \( A \), 31
\( B_\Theta^\omega \) infinitary \( \Theta \)-behavior of \( A \), 31
\( Q_\lambda \) set of reachable states (of \( A \)), 36
\( \Sigma_\lambda \) set of active actions (of \( A \)), 36
\( \delta_T \) set of useful transitions (of \( A \)), 36
\( A_1 \subseteq A_2 \) containment (of \( A_1 \) in \( A_2 \)), 36
\( A_\Theta^f \) \( \Theta \)-action-reduced version of \( A \), 37
\( A_\Theta^T \) \( \Theta \)-transition-reduced version of \( A \), 38
\( A_S \) state-reduced version of \( A \), 46
\( A_A \) action-reduced version of \( A \), 50
\( A_T \) transition-reduced version of \( A \), 50
\( A_R \) reduced version of \( A \), 50

4. Synchronized Automata

\( I \) index set, 59
\( A_I \) automaton, 59
\( S \) set of automata, 59
\( \Delta_a(S) \) complete transition space of \( a \) in \( S \), 60
\( T \) synchronized automaton, 60
\( SUB_J(T) \) the subautomaton of \( T \) determined by \( J \), 64
\( SUB_J \) the subautomaton (of \( T \)) determined by \( J \), 64
\( \pi_{A_J} \) projection on automaton \( A_J \), 70
\( \pi_{SUB_J} \) projection on subautomaton \( SUB_J \), 70
\( \mathcal{D} \) indexed set, 76
\( V(\mathcal{D}) \) 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
\( \langle \langle T \rangle \rangle_S \)  reordered version of synchronized automaton \( T \) (w.r.t. \( S \)), 81
\( T \)  synchronized automaton, 84
\( \text{Free}(T) \)  set of free actions of \( T \), 85
\( \text{AI}(T) \)  set of ai actions of \( T \), 85
\( \text{SI}(T) \)  set of si actions of \( T \), 86
\( \mathcal{R}_a^\text{na}(S) \)  predicate no-constraints, 88
\( \mathcal{R}_a^\text{is-free}(S) \)  predicate is-free for \( a \) in \( S \), 88
\( \mathcal{R}_a^\text{is-ai}(S) \)  predicate is-ai for \( a \) in \( S \), 89
\( \mathcal{R}_a^\text{is-si}(S) \)  predicate is-si for \( a \) in \( S \), 89
\( j \)  element of \( I \), 90
\( J \)  subset of \( I \), 90
\( \Theta \)  arbitrary alphabet disjoint from set \( Q \) of states (of \( T \)), 90

5. Team Automata

\( C \)  component automaton, 116
\( \Sigma_{\text{inp}} \)  set of input actions or input alphabet (of \( C \)), 116
\( \Sigma_{\text{out}} \)  set of output actions or output alphabet (of \( C \)), 116
\( \Sigma_{\text{int}} \)  set of internal actions or internal alphabet (of \( C \)), 116
\( \text{und}(C) \)  underlying automaton of \( C \), 116
\( \Sigma \)  set of actions or (full) alphabet (of \( C \)), 116
\( \Sigma_{\text{ext}} \)  set of external actions or external alphabet (of \( C \)), 116
\( \Sigma_{\text{loc}} \)  set of locally-controlled actions or locally-controlled alphabet (of \( C \)), 117
\( \mathcal{B}_{\Sigma_{\text{inp}}}^\infty \)  input behavior (of \( C \)), 117
\( \mathcal{B}_{\Sigma_{\text{out}}}^\infty \)  output behavior (of \( C \)), 117
\( \mathcal{B}_{\Sigma_{\text{int}}}^\infty \)  internal behavior (of \( C \)), 117
\( \mathcal{B}_{\Sigma_{\text{ext}}}^\infty \)  external behavior (of \( C \)), 117
\( \mathcal{B}_{\Sigma_{\text{loc}}}^\infty \)  locally-controlled behavior (of \( C \)), 117
\( I \)  index set, 118
\( C_i \)  component automaton, 118
\( \Sigma_i \)  set of actions (of \( C_i \)), 118
\( S \)  set of component automata, 118
\( S \)  composable system, 118
\( T \)  team automaton, 120
\( \text{und}(T) \)  underlying synchronized automaton of \( T \), 120
\( \text{SUB}_J(T) \)  the subteam of \( T \) determined by \( J \), 122
\( \text{SUB}_J \)  the subteam (of \( T \)) determined by \( J \), 122
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>composable system, 123</td>
</tr>
<tr>
<td>${(\mathcal{T})}_S$</td>
<td>reordered version of team automaton $\mathcal{T}$ w.r.t. $S$, 125</td>
</tr>
<tr>
<td>$\mathcal{T}$</td>
<td>team automaton, 126</td>
</tr>
<tr>
<td>$\Sigma_{inp}$</td>
<td>set of input actions (of $\mathcal{T}$), 126</td>
</tr>
<tr>
<td>$\Sigma_{out}$</td>
<td>set of output actions (of $\mathcal{T}$), 126</td>
</tr>
<tr>
<td>$\Sigma_{int}$</td>
<td>set of internal actions (of $\mathcal{T}$), 126</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>set of actions (of $\mathcal{T}$), 126</td>
</tr>
<tr>
<td>$\Sigma_{ext}$</td>
<td>set of external actions (of $\mathcal{T}$), 126</td>
</tr>
<tr>
<td>$\Sigma_{loc}$</td>
<td>set of locally-controlled actions (of $\mathcal{T}$), 126</td>
</tr>
<tr>
<td>$I_{a,inp}(S)$</td>
<td>input domain of $a$ in $S$, 126</td>
</tr>
<tr>
<td>$I_{a,out}(S)$</td>
<td>output domain of $a$ in $S$, 126</td>
</tr>
<tr>
<td>$I_{a,inp}(\mathcal{T})$</td>
<td>input subteam of $a$ in $\mathcal{T}$, 127</td>
</tr>
<tr>
<td>$I_{a,out}(\mathcal{T})$</td>
<td>output subteam of $a$ in $\mathcal{T}$, 127</td>
</tr>
<tr>
<td>$\Sigma_{i,ext}$</td>
<td>set of external actions (of $C_i$), 150</td>
</tr>
<tr>
<td>$\Sigma_{i,loc}$</td>
<td>set of locally-controlled actions (of $C_i$), 150</td>
</tr>
<tr>
<td>$j$</td>
<td>element of $I$, 150</td>
</tr>
<tr>
<td>$J$</td>
<td>subset of $I$, 150</td>
</tr>
<tr>
<td>$\Sigma_{j,ext}$</td>
<td>set of external actions (of $SUB_j$), 150</td>
</tr>
<tr>
<td>$\Sigma_{j,loc}$</td>
<td>set of locally-controlled actions (of $SUB_j$), 150</td>
</tr>
</tbody>
</table>

### 6. Behavior of Team Automata

$p\text{REG}$

family of prefix-closed regular finitary languages, 164
List of Symbols

REG family of regular languages, 164
FIN family of finite languages, 164
CA \{ B_C^\Sigma \mid C \text{ is a finite component automaton with alphabet } \Sigma \}, 164
CA^{alph} \{ B_C^{alph} \mid 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
\Sigma_i set of actions (of \mathcal{C}_i), 166
\mathcal{S} composable system, 166
\mathcal{T} team automaton, 166
\Theta set of actions (of \mathcal{T}), 166
j element of \mathcal{I}, 166
uAI_j(\mathcal{T}) set of useful j-ai actions (of \mathcal{T}), 169
|| shuffle, 183
||\text{fair shuffle, 183}
||d|| norm (of decomposition d), 198
||_{i \in [n]} n-ary fair shuffle, 205
||_{i \in [n]} n-ary shuffle, 205
||\Gamma|| S-shuffle on \Gamma, 207
||\text{fair S-shuffle on } \Gamma, 207
alph(L) alphabet (of a language L), 208
\Sigma_1 \Sigma_2 fS-shuffle w.r.t. \Sigma_1 and \Sigma_2, 208
\Sigma_1 \Sigma_2 fS-shuffle w.r.t. \Sigma_1 and \Sigma_2, 208
\Sigma_1 \Sigma_2 rS-shuffle on \Gamma w.r.t. \Sigma_1 and \Sigma_2, 209
\Sigma_1 \Sigma_2 rS-shuffle on \Gamma w.r.t. \Sigma_1 and \Sigma_2, 209
||_{i \in [n]} n-ary fair S-shuffle on \Gamma, 227
||_{i \in [n]} n-ary S-shuffle on \Gamma, 227
||_{i \in [n]} \Sigma_i n-ary fair fS-shuffle w.r.t. \bigcup_{i \in [n]} \Sigma_i, 228
||_{i \in [n]} \Sigma_i n-ary fS-shuffle w.r.t. \bigcup_{i \in [n]} \Sigma_i, 228
||_{i \in [n]} \Sigma_i n-ary fair rS-shuffle on \Gamma w.r.t. \bigcup_{i \in [n]} \Sigma_i, 228
||_{i \in [n]} \Sigma_i n-ary rS-shuffle on \Gamma w.r.t. \bigcup_{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
\Sigma_i set of actions (of \mathcal{C}_i), 233
\mathcal{S} composable system, 233
\mathcal{T} team automaton, 233
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Sigma)</td>
<td>set of actions (of (T)), 233</td>
</tr>
<tr>
<td>(\Sigma_{ext})</td>
<td>set of external actions (of (T)), 233</td>
</tr>
<tr>
<td>(\Sigma_{loc})</td>
<td>set of locally-controlled actions (of (T)), 233</td>
</tr>
<tr>
<td>(\Theta)</td>
<td>arbitrary alphabet disjoint from set (Q) of states (of (T)), 233</td>
</tr>
<tr>
<td>(S)</td>
<td>compatible system, 237</td>
</tr>
<tr>
<td>(T)</td>
<td>team I/O automaton, 239</td>
</tr>
<tr>
<td>IOCA ({B} \cup C)</td>
<td>({B} \cup C) is an alphabet and (C) is a finite input-enabling component automaton with alphabet (\Gamma), 240</td>
</tr>
<tr>
<td>IOCA(\text{alph}) ({B} \cup C)</td>
<td>({B} \cup C) is a finite input-enabling component automaton with alphabet (\Gamma), 240</td>
</tr>
<tr>
<td>(\Delta_v(S))</td>
<td>complete vector transition space (of (a) in (S)), 245</td>
</tr>
<tr>
<td>(a)</td>
<td>vector action (a), 245</td>
</tr>
<tr>
<td>(T^v)</td>
<td>vector team automaton, 245</td>
</tr>
<tr>
<td>(\delta^v)</td>
<td>set of labeled vector transitions (of (T^v)), 245</td>
</tr>
<tr>
<td>(\delta^v_a)</td>
<td>set of vector (a)-transitions (of (T^v)), 245</td>
</tr>
<tr>
<td>(\text{SUB}_J(T^v))</td>
<td>the subteam of (T^v) determined by (J), 246</td>
</tr>
<tr>
<td>(T^\nu)</td>
<td>the flattened version (of (T^v)), 247</td>
</tr>
<tr>
<td>(t\text{Free}(T^v))</td>
<td>set of truly free actions (of (T^v)), 250</td>
</tr>
<tr>
<td>(t\text{AI}(T^v))</td>
<td>set of truly ai actions (of (T^v)), 250</td>
</tr>
<tr>
<td>(t\text{SI}(T^v))</td>
<td>set of truly si actions (of (T^v)), 250</td>
</tr>
<tr>
<td>(\Lambda)</td>
<td>empty word vector, 252</td>
</tr>
<tr>
<td>(\text{tot}({\Delta_j \mid j \in J}))</td>
<td>total vector alphabet (over ({\Delta_j \mid j \in J})), 252</td>
</tr>
<tr>
<td>(\Delta^v)</td>
<td>subset of uniform vector letters of vector alphabet (\Delta), 252</td>
</tr>
<tr>
<td>(v \odot w)</td>
<td>component-wise concatenation (of two (n)-dimensional vector letters (v) and (w)), 252</td>
</tr>
<tr>
<td>(\text{coll})</td>
<td>collapse of a sequence of vector letters into a word vector, 252</td>
</tr>
<tr>
<td>(\text{und}(T^v))</td>
<td>underlying vector automaton (of (T^v)), 253</td>
</tr>
<tr>
<td>(V_{T^v})</td>
<td>finitary vector behavior (of (T^v)), 253</td>
</tr>
<tr>
<td>(V_{T^\nu})</td>
<td>infinitary vector behavior (of (T^v)), 253</td>
</tr>
<tr>
<td>(V_{\text{\nu}})</td>
<td>vector behavior (of (T^v)), 253</td>
</tr>
<tr>
<td>(N)</td>
<td>(n)-VLITN, 254</td>
</tr>
<tr>
<td>(P)</td>
<td>finite set of places (of (N)), 254</td>
</tr>
<tr>
<td>(T)</td>
<td>finite set of events (of (N)), 254</td>
</tr>
<tr>
<td>(O)</td>
<td>finite set of (n) integers, called tokens (of (N)), 254</td>
</tr>
<tr>
<td>(F)</td>
<td>flow function (of (N)), 254</td>
</tr>
<tr>
<td>(V)</td>
<td>vector alphabet of vector labels (of (N)), 255</td>
</tr>
<tr>
<td>(\ell)</td>
<td>event labeling homomorphism (of (N)), 255</td>
</tr>
<tr>
<td>(\text{use}(t))</td>
<td>set of tokens used (by event (t)), 255</td>
</tr>
<tr>
<td>(M_N)</td>
<td>set of all markings of (N), 255</td>
</tr>
<tr>
<td>(\mu[t]_N)</td>
<td>enabled (an event (t) of (N) at a marking (\mu) of (N)), 256</td>
</tr>
<tr>
<td>(\mu[t]_N)</td>
<td>fires (an event (t) of (N) from a marking (\mu) of (N) to a marking (\nu) of (N)), 256</td>
</tr>
</tbody>
</table>
List of Symbols

- $\mu_0[t_1 t_2 \cdots t_m]_N$: firing sequence (of events $t_1, t_2, \ldots, t_m$) of $N$ starting from $\mu_0$, 256
- $\mu_0[t_1 t_2 \cdots t_m]_N \mu_m$: firing sequence (of events $t_1, t_2, \ldots, t_m$) of $N$ starting from $\mu_0$ and leading to $\mu_m$, 256
- $\mu_0[t_1 t_2 \cdots]_N$: infinite firing sequence (of events $t_1, t_2, \ldots$) of $N$ starting from $\mu_0$, 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
- $\mathcal{M}_\mathcal{K}$: 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 ($\mathfrak{a}$): carrier (of $\mathfrak{a}$), 260
- $\text{PN}(\mathcal{T}^v)$: ITNC obtained from $\mathcal{T}^v$, 261
- $\text{SUB}_J(\mathcal{K})$: 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
- $\Sigma$: set of actions (of $\mathcal{T}$), 278
- $\Sigma_{ext}$: set of external actions (of $\mathcal{T}$), 278
- $\Sigma_{H}$: the $\Delta$-hiding version (of $\mathcal{C}$), 278
- $\Sigma_{com}$: set of communicating actions (in $\mathcal{S}$), 279
- $[\mathcal{T}]$: (communication) closed version (of $\mathcal{T}$), 279
- $\mathcal{C}_h$: $h$-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
communicating, 279
complementary, 17
enabled, 51
external, 117
free, 85
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
complementary, 17
enabled, 51
external, 117
free, 85
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
state-indispensable, 86
useful j-action-indispensable, 169
vector, 17, 244, 245
wipp, 129
wms, 132
wopp, 129
active collaboration, 161
alphabet, 24
strong, 129
weak, 129
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
state-indispensable, 86
useful j-action-indispensable, 169
vector, 17, 244, 245
wipp, 129
wms, 132
wopp, 129
active collaboration, 161
alphabet, 24
strong, 129
weak, 129
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
output, 116
alphabetized parallel composition, 206
automaton, 29
action-reduced version of, 50
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
infinitary, 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
coopetration 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
infinitary, 256
vector, 17
formal methods, 12
fs-shuffle, 208, 209
fair, 208, 209
Index 335

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-si, 89
is-sipp, 141
is-smms, 144
is-wopp, 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 $A_j$, 70
on subautomaton $SUB_j$, 70
R-synchronized automaton, 88
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
\(\Theta\)-action-reduced, 104
\(\Theta\)-deterministic, 104
\(\Theta\)-enabling, 93
\(\Theta\)-J-loop-limited, 94
\(\Theta\)-j-loop-limited, 94
\(\Theta\)-loop-limited, 106
\(\Theta\)-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
\(\Delta\)-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
\(\Theta\)-behavior, 31, 117
finitary, 31, 117
infinitary, 31, 117
\(\Theta\)-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