REDUCED STATE SPACE CONSTRUCTION
-
STUBBORN SET REDUCED REACHABILITY GRAPH

QUALITATIVE ANALYSIS METHODS, OVERVIEW

Static analysis:

- NET REDUCTION
- STRUCTURAL PROPERTIES
- LINEAR PROGRAMMING
  - place / transition invariants
  - state equation
  - trap equation

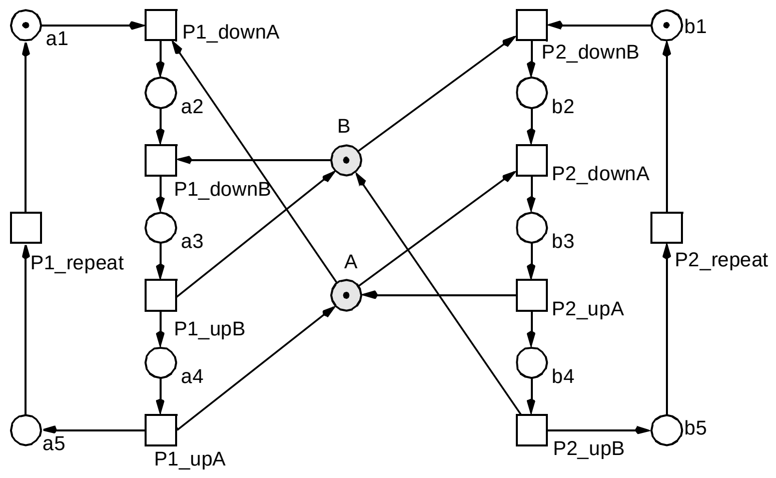
Dynamic analysis *(model checking)*:

- REACHABILITY ANALYSIS
  - (complete) reachability graph
  - compressed state spaces
    - BDDs, NDDs, ..., XDDs
    - Kronecker products
  - reduced state spaces
    - coverability graph
    - symmetry
    - stubborn sets
  - branching process

STUBBORN SETS & REDUCED RG

- basic principle -
  lazy state space construction
  - -> only a subset of the complete rg is constructed
  - -> this subset still allows the decision of certain properties
  - -> **RGred** equiv **RG**
    equivalent with respect to some properties
  - -> suitable equivalence relation ?
- basic idea -
  partial order reduction techniques
  - -> not all interleaving sequences of
    concurrent behavior
    (= partially ordered behavior)
    are considered
- preserved properties
  - -> all dead states
  - -> cyclic behavior

EXAMPLE SYSTEM DEADLOCK, PETRI NET

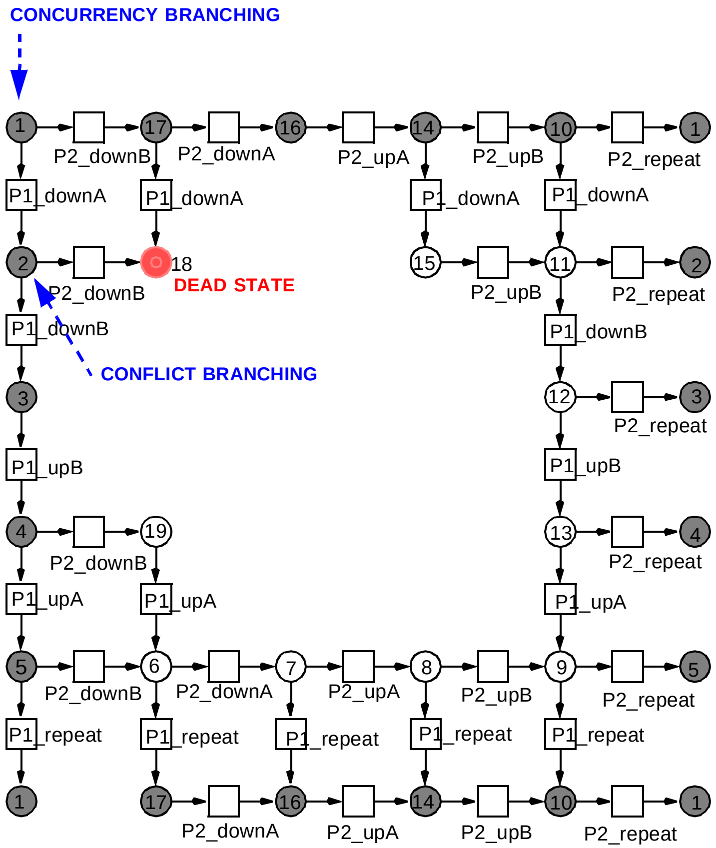


INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y

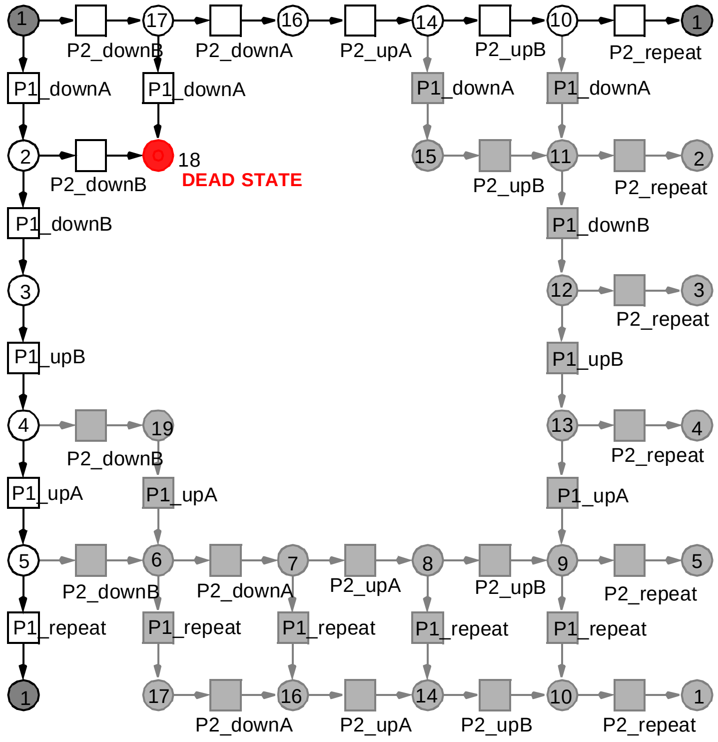
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	Y	N	Y	Y	Y	Y	N	Y	?	N	N	N	N	N

EXAMPLE SYSTEM DEADLOCK, (COMPLETE) RG



19 nodes,
32 arcs

EXAMPLE, SYSTEM DEADLOCK, REDUCED RG



10 nodes,
12 arcs

STUBBORN SET, CHARACTERISTICS

- a marking-dependent selection of a set of independent transitions

- a set of independent transitions
  - -> their behavior cannot be influenced by the excluded transitions
    - -> *“they are stubborn”*
  - -> any sequence of excluded transitions cannot enable or disable an included transition
    - -> their firing can be postponed
  - -> contains at least one enabled transition

- stubborn set reduced rg
  - -> slight variation of the standard algorithm
  - -> at each marking (node):

    instead of firing all enabled transitions, only transitions of a stubborn set are fired

REACHABILITY GRAPH, CONSTRUCTION ALGORITHM

ROCEDURE rg (**IN** Net pn, **IN** Marking $m0$,
OUT MSet $nodes$, **OUT** ArcSet $arcs$);

MSet $U = \{m0\}$, // unprocessed markings
$N = \emptyset$; // rg nodes
ArcSet $E = \emptyset$; // rg arcs (pre, post, t)
Marking m'; // successor marking
Transition t;

WHILE $U \neq \emptyset$ **DO**
choose one $m \in U$;
$U = U - \{m\}$; $N = N \cup \{m\}$;

FOR ALL t enabled at m **DO**
$m' = m + \Delta t$;
IF $m' \notin N \cup U$ // new marking
THEN $U = U \cup \{m'\}$
ENDIF;
$E = E \cup \{(m, m', t)\}$
ENDFOR

ENDWHILE;

$nodes = N$; $arcs = E$;

ENDPROC rg.

STUBBORN REDUCED RG, CONSTRUCTION ALGORITHM

ROCEDURE rg (**IN** Net *pn*, **IN** Marking $m0$,
OUT MSet *nodes*, **OUT** ArcSet *arcs*);

MSet $U = \{m0\}$, // unprocessed markings
$N = \emptyset$; // rg nodes
ArcSet $E = \emptyset$; // rg arcs (pre, post, t)
Marking m'; // successor marking
Transition t;

WHILE $U \neq \emptyset$ **DO**
choose one $m \in U$;
$U = U - \{m\}$; $N = N \cup \{m\}$;

FOR ALL enabled t of a stubborn set at m **DO**
$m' = m + \Delta t$;
IF $m' \notin N \cup U$ // new marking
THEN $U = U \cup \{m'\}$
ENDIF;
$E = E \cup \{(m, m', t)\}$
ENDFOR

ENDWHILE;

$nodes = N$; $arcs = E$;

ENDPROC rg.

HOW TO CONSTRUCT STUBBORN SETS

- three basic steps

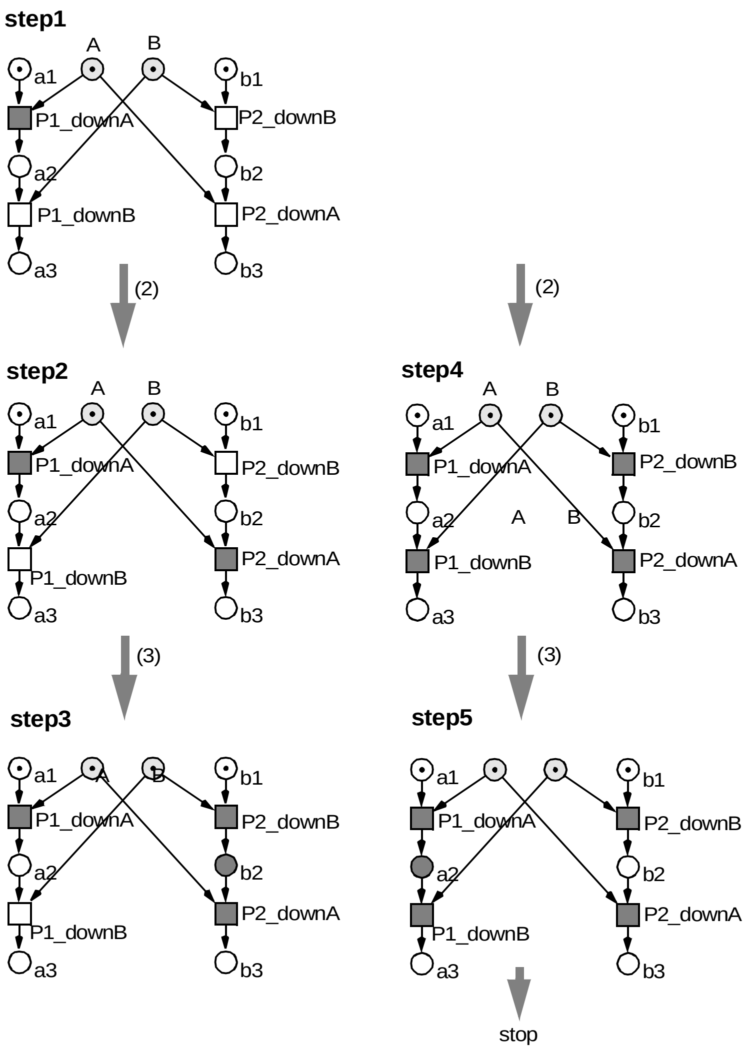
(1) choose an enabled transition t and put it into U

(2) **FOR ALL** enabled transition t in U:
put into U also all transitions in conflict with t
END FOR
-> conflict transitions: (Ft)F
-> any sequence of excluded transitions cannot disable an included transition

(3) **FOR ALL** disabled transition t in U:
choose a scapegoat
(a place p which prevents t from being enabled),
and put all pre-transitions of p (Fp) into U
END FOR
-> any sequence of excluded transitions cannot enable an included transition

- repeat (2) and (3) as long as necessary

STUBBORN SETS, EXAMPLES (1)



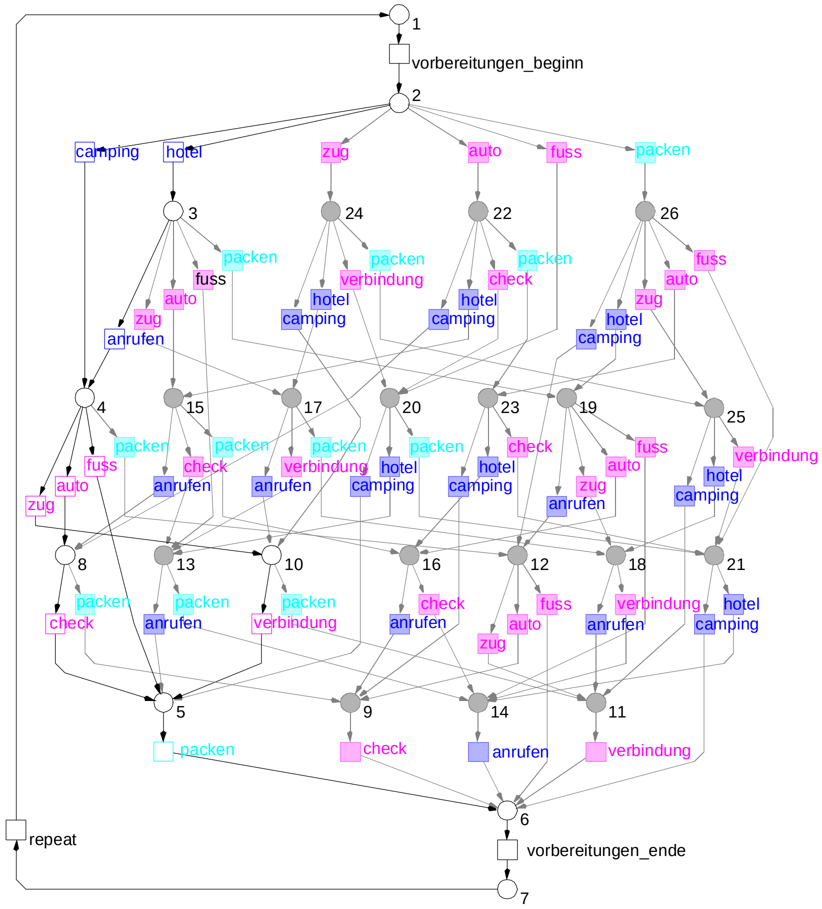
STUBBORN SETS, EXAMPLES (2)

- any conflict-free enabled transition
  - -> is a stubborn set for itself
- for any dead state
  - -> there is no stubborn set
- for non-dead states
  - -> set of all transitions is a stubborn set

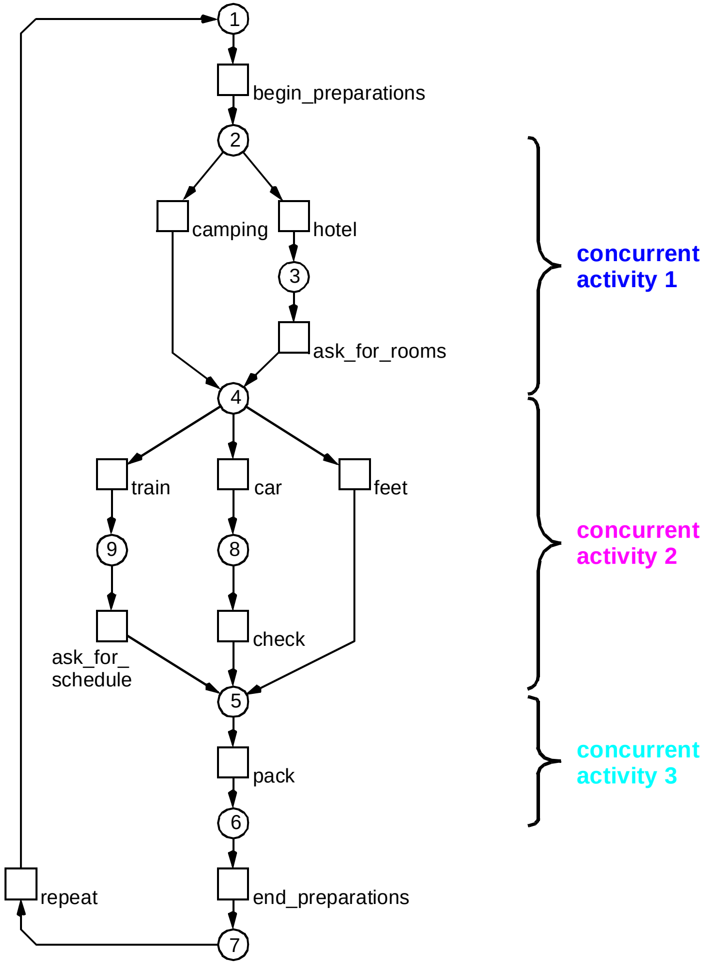
STUBBORN SETS, OBSERVATIONS

- each set U constructed by this way is a stubborn set at m
- result U depends on the current marking m
- non-deterministic stubborn set construction
  - -> result depends on non-deterministic choices
    - -> choose an enabled transition t
    - -> choose a scapegoat p
- smaller stubborn sets result generally into smaller reduced rg
  - -> **BUT,** there are counter examples
- there are various heuristics to determine smaller stubborn sets, -> basic step (3)
- at best: minimal stubborn sets (contain no stubborn subset)
- **BUT**, increasing computational effort
  - -> may exceed benefit gained
  - -> what is more worth: space or run time ?

EXAMPLE TRAVEL PLANNING, RG



EXAMPLE TRAVEL PLANING



-> only one interleaving sequence is represented

DINING PHILOSOPHERS, RG AND REDUCED RG SIZES

# phils	P / T	R_{stub}	R
1	6 / 4	4	4
2	10 / 8	8	10
3	15 / 12	20	35
4	20 / 16	38	118
5	25 / 20	62	392
6	30 / 24	92	1.297
7	35 / 28	128	4.286
8	40 / 32	170	14.158
9	45 / 36	218	46.763
10	50 / 40	272	154.450
11	55 / 44	332	510.116
12	60 / 48	398	
13	65 / 52	470	(5.56 e+6)
14	70 / 56	548	
15	75 / 60	632	(60.7 e+6)

PRODUCTION CELL, COOPERATION MODEL

	places/ transitions	DTP	R_{stub}	R
table / press with init part without init part	 13 / 9 12 / 8	 (N) 28	 12 8	 28 24
crane	12 / 8	31	11	48
arms version 1 version 2 version 3	 13 / 8 17 / 12 17 / 12	 38 109 88	 11 15 15	 48 112 96
belts	12 / 8	26	8	36
subsystem with arm version 1 arm version 2 arm version 3	 25 / 16 33 / 24 33 / 24	 175 3.851 (N) 725	 47 75 140	 640 1.984 1.800
open system	51 / 36	1.145	299	77.760
closed system with 1 plate with 2 plates with 3 plates with 4 plates with 5 plates	51 / 36	1.140	 36 72 94 98 121	 864 4.776 12.102 16.362 12.144

PRODUCTION CELL, CONTROL MODEL

system part	P / T	PROD					
		R	time	R_{stub} a)	time	R_{stub} b)	time
controllers							
crane	45/34	256	0.78”	51	0.16”	38	0.08”
feed belt	22/16	69	0.20”	31	0.10”	16	0.07”
table	32/24	88	0.38”	36	0.15”	24	0.09”
arm, v3	66/60	365”	1.19”	62	0.23”	51	0.09”
press	28/20	140	0.42”	48	0.10”	20	0.09”
deposit belt	22/16	69	0.20”	31	0.11”	16	0.07”
composed systems							
robot	124/120	63,232	11.26 ’	992	5.99”	205	0.21”
robot/ press	140/132	18,344	3.10”	557	3.46”	305	0.35”
open system	198/176	2,776,936	?	798	5.90”	507	0.62”
closed system 1 plate 2 plates 3 plates 4 plates 5 plates	231/202	 30,952 543,480 > 1,7 Mio > 3.1 Mio 1,657,242	 7.54’ 3.3 h >20 h >42 h 14 h	 162 406 523 471 585	 0.68” 2.53” 4.51” 4.02” 5.05”	 163 456 635 678 608	 0.32” 0.72” 0.95” 1.06” 0.98”

a) deletion algorithm
b) incremental algorithm

EXAMPLE PUSHERS

# pushers	R	version 1 P / T	version 1 R_{stub}	version 2 P / T	version 2 R_{stub}
1	88	24 / 25	22	24 / 21	22
2	464	42 / 46	42	42 / 38	42
3	3.088	60 / 67	79	60 / 55	79
4	18.848	78 / 88	133	78 / 72	133
5	118.624	96 / 109	204	96 / 89	204
6	0.7 e+6	114 / 130	292	114 / 106	292
7	4.6 e+6	132 / 151	397	132 / 123	397
8	28.9 e+6	150 / 172	519	150 / 140	519
9	179.8 e+6	168 / 193	658	168 / 157	658
10	1.1 e+9	186 / 214	814	186 / 174	814

- ❑ version 1 - many dynamic conflicts
- ❑ version 2 - persistent

dependability engineering & Petri nets *May 2005*

SUMMARY, OUTLOOK

- ❑ reduction effect needs concurrently enabled transitions
  - -> more than one
  - -> no conflict in between
- ❑ for system without concurrency
  - -> RG = RGred
- ❑ on-the-fly model checking of LTL\X

REFERENCES

[Gerth 95]
Gerth, R., Peled, D., Vardi, M. Y., Wolper, P.:
Simple On-the-fly Automatic Verification of Linear Temporal Logic;
Proc. of the 15th International Symposium on Protocol Specification, Testing and Verification (PSTV'95), Warsaw 1995, 3-18.

[Godefroid 96]
Godefroid, P.:
Partial-Order Methods for the Verification of concurrent Systems;
LNCS 1032, 1996.

[Pogrell 95]
Master Thesis, Humboldt Univ. at Berlin, 1995.

[Starke 92]
Starke, P. H.; Roch, S.:
INA - Integrated Net Analyser version 1.7;
Technical report, Humbold University at Berlin, 1997,
http://www.informatik.hu-berlin.de/lehrstuehle/automaten/ina.

[Valmari 92]
Valmari, A.:
A Stubborn Attack on State Explosion;
Formal Methods in System Design 1(1992)4, 297-322.

[Valmari 92]
Valmari, A.:
Alleviating State Explosion during Verification of Behavioral Equivalence;
Univ. of Helsinki, Department of Computer Science, Report A-1992-4, Helsinki 1992.

[Varpaaniemi 95]
Varpaaniemi, K.; Halme, J.; Hiekkanen, K.; Pyssysalo, T.:
PROD Reference Manual;
Helsinki Univ. of Technology, Digital Systems Laboratory, Series B: Techn. Report No. 13, August 1995, ftp://saturn.hut.fi/pub/reports