

Trans1 vs. Trans2

(without further optimisations, \SM=state merging disabled, +DS=manually added d_steps)

Example	Version	StateVector (byte)	#stored states	#matched states	#transitions	depth	totalMemory (.MByte)	time (sec)
HalfAdder	Trans1	44	14	0	14	17	< 2.501	0, __
<i>spin -a -o3</i>	Trans1\SM	44	18	0	18	21	< 0.403	0, __
	Trans2	40	22	0	22	25	< 2.501	0, __
<i>(manually)</i>	Trans2+DS	40	15	0	15	18	< 0.403	0, __
FullAdder	Trans1	128	97	0	97	130	< 2.501	0, 01
<i>spin -a -o3</i>	Trans1\SM	128	170	0	170	203	< 0.403	0, __
	Trans2	116	178	0	178	211	< 2.501	0, 01
<i>(manually)</i>	Trans2+DS	116	122	0	122	155	< 0.403	0, __
Adder [_X]	Trans1	224	9.083	517	9.600	2.336	3.868	0, 07
<i>spin -a -o3</i>	Trans1\SM	224	11.387	517	11.904	2.930	2.063	0, 26
<i>cc -DNOREDUCE pan</i>	Trans1\POR	224	12.222	5.919	18.141	2.336	4.258	0, 1_
	Trans2	192	11.067	517	11.584	2.864	3.770	0, 05
<i>(manually)</i>	Trans2+DS	192	10.220	517	10.737	2.627	1.575	0, 23
RedundantBatterySystem2	<i>Both translations have been manually extended with c_code { printf(“...”) }, to see output of reals in pan.</i> <i>Pan has been run with option -m1000000 to avoid “error: max. search depth too small”</i>							
	Trans1	164	1.108.206	141.535	1.249.741	497.670	193.650	5, 7_
<i>spin -a -o3</i>	Trans1\SM	164	1.191.866	141.535	1.333.401	540.249	203.221	6, 07
	Trans2	128	1.136.883	142.519	1.279.402	526.019	163.182	5, 18
<i>(manually)</i>	Trans2+DS	128	1.011.719	141.875	1.153.594	469.213	151.268	4, 49
TransitionChangesDataport	Trans1	224	148	0	148	192	< 2.501	0, 01

<i>spin -a -o3</i>	Trans1\SM	224	212	0	212	256	< 0.403	0, __
	Trans2	136	201	0	201	243	< 2.501	0,01
<i>(manually)</i>	Trans2+DS	136	167	0	167	209	< 0.403	0, __

Even without explicit d_steps, Trans1 has a smaller state space than Trans2, due to State Merging (which implicitly adds d_steps to sequence of assignments to *local* variables in Trans1). This explains why adding explicit d_steps does not improve Trans1 and suggests the solution to reduce state space of Trans2: add explicit d_steps!

⌞ Even if state space of Trans2 should still be a little bit greater than that for Trans1 (partially due to initialisations missing in Trans1), Trans2 saves memory (and time) as its state vectors are smaller!