Programming Project – Requirements and Testing

Try not to evaluate the BDD by retrieving from the library all implicants of a function (paths to the 1-terminal). This is inefficient: the number of paths is doubly exponential in the number of BDD variables (nevertheless, the method you use does not have to be efficient but it has to be correct). Do evaluate the BDD by traversing it recursively yourself, and use the computed-table to do it efficiently (see slides by Brayton & Kuehlmann). Use the unique-table to build the BDD from the fault tree.

To be able to reference more than one state per MC from the FT you'll want to use the BDD evaluation method by Zang et al (pp 1613-14). Remember that you don't need to build the BDD differently (as Zang et al propose it), since the same effect can obviously be achieved by tweaking the traversal (consult the document "BDD Evaluation with Restricted Variables_(FDDS15).pdf" for a concise explanation). You'll need to calculate probabilities in a different way to take the stochastic dependence between states of the same MC into account. This cannot efficiently be done by using the set of implicants. Consider the example in Figure 1 to see this (and make a gedankenexperiment with your algorithm, imagining there to be 50 or so variables).

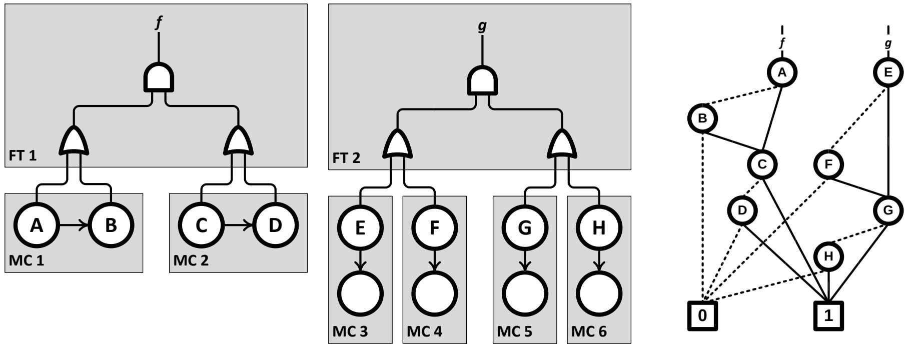

Figure 1. Different HFTs with the same BDD translation.

f and *g* are the same function, differing only in the stochastic dependences of their referenced MC. Also their BDD are identical (see on the right). The cut sets (implicants disregarding negated variables) of the variable order chosen above for *f* are { { A, C }, { A, D }, { B, C }, { B, D } } (*g* is analogous). Pr(*f*) is always 1, Pr(*g*) is not.

Requirements

Let's review the requirements for your FTA tool:

(1) Teams of two program a fault tree analysis tool, using the language and development environment of their choice.

  (a) The tool shall be able to analyse fault trees with Markov chains (MC) as stochastic sources. This model is what we call "hybrid fault trees" (HFT).

    i. The MCs shall be continuous time (CTMC), meaning that transition frequencies are specified as rates (per time unit).

    ii. Time units do not have to be specifiable; in this case, they are assumed to be identical everywhere they would turn up (mission time, transition rates, sampling interval), for example, seconds and per second.

    iii. An initial probability distribution over the states of every MC shall be specifiable.

  (b) The fault tree to analyse may have meshes; it shall use the Boolean functions ('gates') AND and OR (negation/NOT is not required).

(c) It shall be possible, from the fault tree, to reference more than one state of the same MC.
(d) It shall be possible to reference, from the fault tree, the same MC state multiple times.
(e) It shall be possible to specify a sampling interval for the time axis (alternatively, the number of sampling points over the mission time), as well as a global mission time.
(f) It shall be possible to give meaningful names to MC states, FT gates and to the top event.
(g) The tool shall produce a series of top-event probabilities of the fault tree over time; these probabilities shall be exact and calculated using the ROBDD equivalent to the Boolean function the fault tree represents.
(h) Calculation results shall be output as text (list of pairs of point in time and corresponding probability) and visually as a graph (x-axis: time, y-axis: top-event probability).

(2) The program source code, the presentation and a brief documentation for potential users has to be delivered about a week after the team presentations.
(3) All non-private classes and methods/functions of the program shall be well documented in Javadoc style (or equivalent).

As previously stated, the fault tree to analyse can be specified in any format you like: programmatic, as text/XML, as a graph etc. In any case, it has to be parsed by the program and converted to a BDD. The BDD in turn is evaluated for every sampling point in time with the corresponding basic-event probabilities to produce the resulting time series of top-event probabilities.

Testing

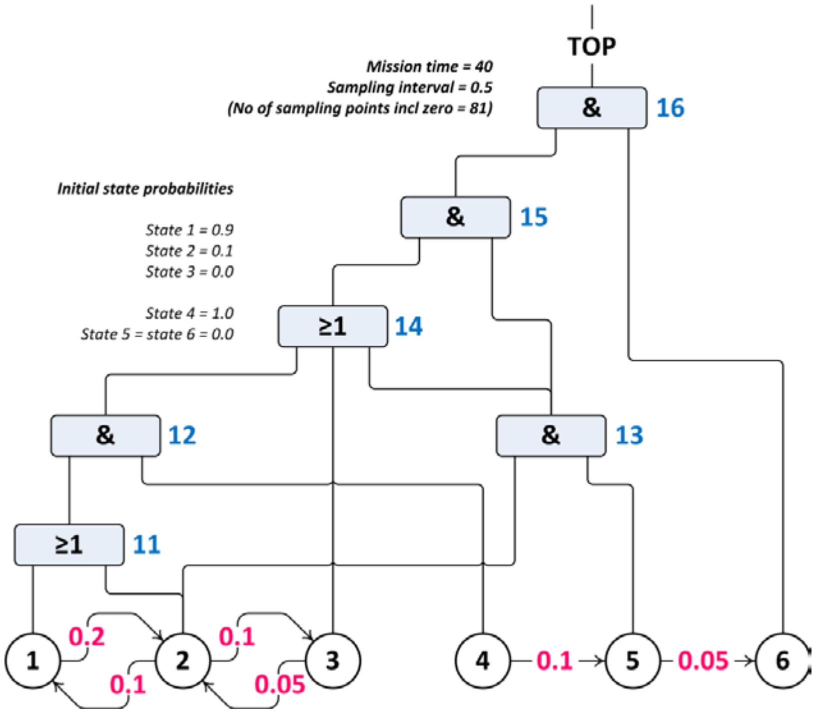

Figure 2. Test case HFT.

To make sure that the hybrid fault tree (HFT) evaluation program works correctly, you'll need test cases. Figure 2 shows a HFT model containing six of them, where every fault tree gate (11-16) represents one function that can serve as the top event, and whose probability can be calculated. Obviously, (the probabilities of) the functions of gate 13 and of gate 15 are identical and have to yield the same probability series. The probability of the function of gate 16 is a constant zero. The parameters of evaluation are: Mission time = 40; Sampling interval = 0.5 (equivalently, 81 sampling points, including t = 0); Initial state probabilities: State 1 = 0.9, state 2 = 0.1, state 3 = 0.0, state 4 = 1.0, state 5 = 0.0, state 6 = 0.0. Gates and transition rates as in Figure 2. Table 1 contains the time series of the correct results.

Time	Markov chain A, state probabilities			Markov chain B, state probabilities			Functions, probabilities					
	State 1	State 2	State 3	State 4	State 5	State 6	Gate 11	Gate 12	Gate 13	Gate 14	Gate 15	Gate 16
0.0	0.90000	0.10000	0.00000	1.00000	0.00000	0.00000	1.00000	1.00000	0.00000	1.00000	0.00000	0.00000
0.5	0.82092	0.17229	0.00679	0.95123	0.04816	0.00061	0.99321	0.94477	0.00830	0.95986	0.00830	0.00000
1.0	0.75246	0.23090	0.01664	0.90484	0.09278	0.00238	0.98336	0.88978	0.02142	0.92784	0.02142	0.00000
1.5	0.69303	0.27813	0.02884	0.86071	0.13407	0.00522	0.97116	0.83588	0.03729	0.90201	0.03729	0.00000
2.0	0.64126	0.31591	0.04283	0.81873	0.17221	0.00906	0.95717	0.78366	0.05440	0.88090	0.05440	0.00000
2.5	0.59602	0.34583	0.05815	0.77880	0.20739	0.01381	0.94185	0.73351	0.07172	0.86339	0.07172	0.00000
3.0	0.55634	0.36926	0.07439	0.74082	0.23978	0.01940	0.92561	0.68571	0.08854	0.84864	0.08854	0.00000
3.5	0.52143	0.38731	0.09126	0.70469	0.26954	0.02577	0.90874	0.64038	0.10440	0.83603	0.10440	0.00000
4.0	0.49058	0.40093	0.10849	0.67032	0.29682	0.03286	0.89151	0.59760	0.11901	0.82509	0.11901	0.00000
4.5	0.46323	0.41091	0.12587	0.63763	0.32178	0.04060	0.87413	0.55737	0.13222	0.81546	0.13222	0.00000
5.0	0.43888	0.41789	0.14323	0.60653	0.34454	0.04893	0.85677	0.51965	0.14398	0.80687	0.14398	0.00000
5.5	0.41711	0.42243	0.16045	0.57695	0.36524	0.05781	0.83955	0.48438	0.15429	0.79912	0.15429	0.00000
6.0	0.39759	0.42499	0.17742	0.54881	0.38401	0.06718	0.82258	0.45144	0.16320	0.79206	0.16320	0.00000
6.5	0.38000	0.42594	0.19406	0.52205	0.40096	0.07699	0.80594	0.42074	0.17079	0.78558	0.17079	0.00000
7.0	0.36410	0.42560	0.21030	0.49659	0.41621	0.08721	0.78970	0.39216	0.17714	0.77959	0.17714	0.00000
7.5	0.34968	0.42423	0.22609	0.47237	0.42985	0.09779	0.77391	0.36557	0.18235	0.77401	0.18235	0.00000
8.0	0.33654	0.42206	0.24141	0.44933	0.44198	0.10869	0.75859	0.34086	0.18654	0.76881	0.18654	0.00000
8.5	0.32453	0.41925	0.25622	0.42741	0.45271	0.11988	0.74378	0.31790	0.18980	0.76392	0.18980	0.00000
9.0	0.31351	0.41597	0.27052	0.40657	0.46212	0.13131	0.72948	0.29659	0.19223	0.75933	0.19223	0.00000
9.5	0.30339	0.41233	0.28429	0.38674	0.47029	0.14297	0.71571	0.27679	0.19391	0.75500	0.19391	0.00000
10.0	0.29404	0.40843	0.29753	0.36788	0.47730	0.15482	0.70247	0.25842	0.19494	0.75090	0.19494	0.00000
10.5	0.28539	0.40435	0.31026	0.34994	0.48324	0.16683	0.68974	0.24137	0.19540	0.74702	0.19540	0.00000
11.0	0.27737	0.40017	0.32246	0.33287	0.48816	0.17897	0.67754	0.22553	0.19534	0.74334	0.19534	0.00000
11.5	0.26992	0.39593	0.33415	0.31664	0.49214	0.19123	0.66585	0.21083	0.19485	0.73984	0.19485	0.00000
12.0	0.26297	0.39168	0.34535	0.30119	0.49523	0.20357	0.65465	0.19718	0.19398	0.73650	0.19398	0.00000

12.5	0.25648	0.38746	0.35606	0.28650	0.49751	0.21598	0.64394	0.18449	0.19277	0.73332	0.19277	0.00000
13.0	0.25040	0.38330	0.36630	0.27253	0.49903	0.22844	0.63370	0.17270	0.19128	0.73028	0.19128	0.00000
13.5	0.24471	0.37921	0.37608	0.25924	0.49983	0.24093	0.62392	0.16175	0.18954	0.72736	0.18954	0.00000
14.0	0.23937	0.37521	0.38542	0.24660	0.49998	0.25343	0.61458	0.15155	0.18760	0.72457	0.18760	0.00000
14.5	0.23435	0.37131	0.39433	0.23457	0.49951	0.26592	0.60567	0.14207	0.18547	0.72188	0.18547	0.00000
15.0	0.22963	0.36754	0.40284	0.22313	0.49847	0.27840	0.59716	0.13324	0.18321	0.71929	0.18321	0.00000
15.5	0.22517	0.36388	0.41095	0.21225	0.49691	0.29084	0.58905	0.12502	0.18081	0.71679	0.18081	0.00000
16.0	0.22097	0.36034	0.41868	0.20190	0.49486	0.30324	0.58132	0.11737	0.17832	0.71437	0.17832	0.00000
16.5	0.21701	0.35694	0.42606	0.19205	0.49237	0.31558	0.57394	0.11023	0.17575	0.71203	0.17575	0.00000
17.0	0.21326	0.35366	0.43308	0.18268	0.48946	0.32785	0.56692	0.10357	0.17310	0.70975	0.17310	0.00000
17.5	0.20972	0.35051	0.43977	0.17377	0.48618	0.34005	0.56023	0.09735	0.17041	0.70754	0.17041	0.00000
18.0	0.20636	0.34749	0.44615	0.16530	0.48254	0.35216	0.55385	0.09155	0.16768	0.70538	0.16768	0.00000
18.5	0.20319	0.34459	0.45222	0.15724	0.47859	0.36417	0.54778	0.08613	0.16492	0.70327	0.16492	0.00000
19.0	0.20018	0.34182	0.45800	0.14957	0.47434	0.37609	0.54200	0.08107	0.16214	0.70121	0.16214	0.00000
19.5	0.19733	0.33916	0.46351	0.14227	0.46984	0.38789	0.53649	0.07633	0.15935	0.69919	0.15935	0.00000
20.0	0.19463	0.33663	0.46875	0.13534	0.46509	0.39958	0.53125	0.07190	0.15656	0.69720	0.15656	0.00000
20.5	0.19207	0.33420	0.47373	0.12873	0.46012	0.41114	0.52627	0.06775	0.15377	0.69526	0.15377	0.00000
21.0	0.18963	0.33188	0.47848	0.12246	0.45496	0.42258	0.52152	0.06386	0.15099	0.69334	0.15099	0.00000
21.5	0.18732	0.32967	0.48300	0.11648	0.44963	0.43389	0.51700	0.06022	0.14823	0.69145	0.14823	0.00000
22.0	0.18513	0.32756	0.48730	0.11080	0.44414	0.44506	0.51270	0.05681	0.14548	0.68959	0.14548	0.00000
22.5	0.18305	0.32555	0.49140	0.10540	0.43851	0.45609	0.50860	0.05361	0.14276	0.68776	0.14276	0.00000
23.0	0.18108	0.32363	0.49529	0.10026	0.43276	0.46699	0.50471	0.05060	0.14005	0.68595	0.14005	0.00000
23.5	0.17920	0.32180	0.49900	0.09537	0.42690	0.47773	0.50100	0.04778	0.13738	0.68416	0.13738	0.00000
24.0	0.17741	0.32006	0.50253	0.09072	0.42095	0.48833	0.49747	0.04513	0.13473	0.68239	0.13473	0.00000
24.5	0.17572	0.31840	0.50588	0.08629	0.41493	0.49878	0.49412	0.04264	0.13211	0.68063	0.13211	0.00000
25.0	0.17411	0.31682	0.50907	0.08208	0.40884	0.50908	0.49093	0.04030	0.12953	0.67890	0.12953	0.00000
25.5	0.17258	0.31531	0.51211	0.07808	0.40270	0.51922	0.48789	0.03810	0.12697	0.67718	0.12697	0.00000
26.0	0.17112	0.31388	0.51500	0.07427	0.39652	0.52921	0.48500	0.03602	0.12446	0.67548	0.12446	0.00000
26.5	0.16974	0.31251	0.51775	0.07065	0.39030	0.53905	0.48225	0.03407	0.12197	0.67380	0.12197	0.00000
27.0	0.16842	0.31121	0.52037	0.06721	0.38407	0.54872	0.47963	0.03223	0.11953	0.67213	0.11953	0.00000
27.5	0.16717	0.30997	0.52286	0.06393	0.37782	0.55825	0.47714	0.03050	0.11711	0.67047	0.11711	0.00000
28.0	0.16599	0.30879	0.52522	0.06081	0.37157	0.56762	0.47478	0.02887	0.11474	0.66883	0.11474	0.00000

28.5	0.16486	0.30767	0.52748	0.05784	0.36533	0.57683	0.47252	0.02733	0.11240	0.66721	0.11240	0.00000
29.0	0.16378	0.30660	0.52962	0.05502	0.35909	0.58588	0.47038	0.02588	0.11010	0.66560	0.11010	0.00000
29.5	0.16276	0.30558	0.53166	0.05234	0.35288	0.59478	0.46834	0.02451	0.10783	0.66400	0.10783	0.00000
30.0	0.16179	0.30462	0.53360	0.04979	0.34669	0.60353	0.46640	0.02322	0.10561	0.66242	0.10561	0.00000
30.5	0.16086	0.30370	0.53544	0.04736	0.34052	0.61212	0.46456	0.02200	0.10342	0.66086	0.10342	0.00000
31.0	0.15998	0.30282	0.53719	0.04505	0.33440	0.62055	0.46281	0.02085	0.10126	0.65931	0.10126	0.00000
31.5	0.15915	0.30199	0.53886	0.04285	0.32831	0.62884	0.46114	0.01976	0.09915	0.65777	0.09915	0.00000
32.0	0.15835	0.30120	0.54045	0.04076	0.32227	0.63697	0.45955	0.01873	0.09707	0.65625	0.09707	0.00000
32.5	0.15760	0.30044	0.54196	0.03877	0.31627	0.64495	0.45804	0.01776	0.09502	0.65474	0.09502	0.00000
33.0	0.15688	0.29972	0.54340	0.03688	0.31033	0.65278	0.45660	0.01684	0.09301	0.65325	0.09301	0.00000
33.5	0.15619	0.29904	0.54477	0.03508	0.30445	0.66047	0.45523	0.01597	0.09104	0.65178	0.09104	0.00000
34.0	0.15554	0.29839	0.54607	0.03337	0.29862	0.66801	0.45393	0.01515	0.08911	0.65032	0.08911	0.00000
34.5	0.15492	0.29777	0.54730	0.03175	0.29285	0.67540	0.45270	0.01437	0.08720	0.64888	0.08720	0.00000
35.0	0.15433	0.29719	0.54848	0.03020	0.28715	0.68265	0.45152	0.01363	0.08534	0.64745	0.08534	0.00000
35.5	0.15377	0.29663	0.54960	0.02872	0.28152	0.68976	0.45040	0.01294	0.08351	0.64604	0.08351	0.00000
36.0	0.15324	0.29610	0.55066	0.02732	0.27595	0.69673	0.44934	0.01228	0.08171	0.64465	0.08171	0.00000
36.5	0.15273	0.29559	0.55168	0.02599	0.27045	0.70356	0.44832	0.01165	0.07994	0.64327	0.07994	0.00000
37.0	0.15225	0.29511	0.55264	0.02472	0.26503	0.71025	0.44736	0.01106	0.07821	0.64191	0.07821	0.00000
37.5	0.15179	0.29465	0.55356	0.02352	0.25967	0.71681	0.44644	0.01050	0.07651	0.64057	0.07651	0.00000
38.0	0.15136	0.29421	0.55443	0.02237	0.25440	0.72323	0.44557	0.00997	0.07485	0.63924	0.07485	0.00000
38.5	0.15094	0.29380	0.55526	0.02128	0.24919	0.72953	0.44474	0.00946	0.07321	0.63793	0.07321	0.00000
39.0	0.15055	0.29341	0.55604	0.02024	0.24406	0.73569	0.44396	0.00899	0.07161	0.63664	0.07161	0.00000
39.5	0.15017	0.29303	0.55680	0.01925	0.23901	0.74173	0.44320	0.00853	0.07004	0.63537	0.07004	0.00000
40.0	0.14982	0.29267	0.55751	0.01832	0.23404	0.74765	0.44249	0.00810	0.06850	0.63411	0.06850	0.00000

Table 1. Probability time series of MC states and gate functions of the HFT from Figure 2.

These series are also in a spreadsheet, in Open Office and Microsoft Office format: “TestCaseForProgrammingProject.ods/.xlsx”. Figure 3 shows them in a graph.

To test that your conversion of the FT to a BDD is correct, create some fault trees of your own, convert them to a BDD by hand and compare. It will be handy to include an automatic export of BDDs to a dot file in your program so you can inspect the BDD graph with Dotty or Graphviz. You can also use the corresponding function of JavaBDD, it’s already there but may need some adaptation to make it work.

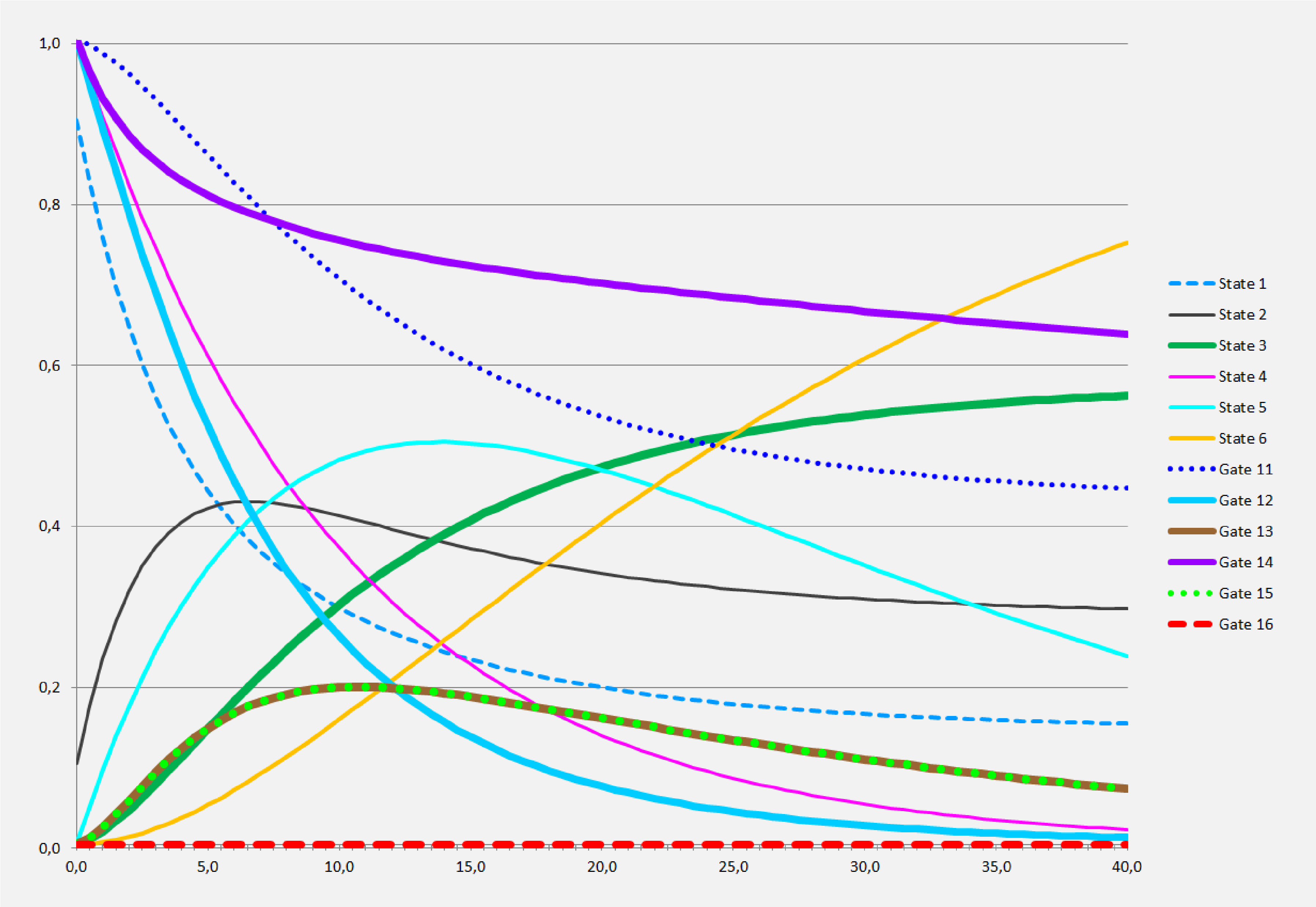

Figure 3. Probability values of MC states and gate functions of the HFT from Figure 2, as a graph.