This additional file has some benchmarks of Antelope. The table below shows the average verification times for a few random GRNs without indeterminations, with respect to some properties. These figures do not include the time taken by the web browser to display the results. \( N \) denotes the number of genes and \( K \) denotes the average number of inputs of a gene in the interaction diagram. In the generation of these networks, the probability of the value of a gene being 1 (or 0) was \( p = 0.5 \). We have separated the time taken to build the BDD of the dynamics of the GRN from the time taken to perform the verification itself: The column named “BDD” shows the time taken to build such a BDD. These benchmarks were run on a laptop with 3 Gb RAM, a Core(TM)2 Duo CPU T6670 at 1.2 GHz, under Ubuntu 10.04.2. We used the Java implementation of JavaBDD [3] (as opposed to BuDDy [1]).

The formula
\[
!s.\text{AX } s
\]
holds exactly at all stable steady states, and can be taken to be representative of Hybrid CTL formulas involving a single traversal of all states of a GRN, such that for each state its predecessors are obtained. Next, the formula schema
\[
\bigwedge_{i=0}^{n-1} g_i \leftrightarrow f_i(g_0, \ldots, g_{n-1}),
\]
which we call *equational fixed point*, and abbreviate as \texttt{efp}, results in a more efficient way of computing the set of steady states for GRN without indeterminations; this method is used in [2], and can be used in Antelope by selecting the option “Use conjunctive equational fixed point formula”. Finally, the formula
\[
\text{EF } s
\]
holds at all nodes in the basin of attraction of state \( s \). This formula can be taken to be representative of formulas with a single CTL operator different from \texttt{AX} or \texttt{EX}. The entry “of.” means memory overflow.
<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>BDD</th>
<th>formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.4</td>
<td>41 ms</td>
<td>!s.AX s</td>
</tr>
<tr>
<td>5</td>
<td>2.8</td>
<td>53 ms</td>
<td>4 ms</td>
</tr>
<tr>
<td>10</td>
<td>1.4</td>
<td>44 ms</td>
<td>71 ms</td>
</tr>
<tr>
<td>10</td>
<td>2.5</td>
<td>29 ms</td>
<td>225 ms</td>
</tr>
<tr>
<td>15</td>
<td>1.4</td>
<td>167 ms</td>
<td>568 ms</td>
</tr>
<tr>
<td>15</td>
<td>2.3</td>
<td>40 ms</td>
<td>863 ms</td>
</tr>
<tr>
<td>20</td>
<td>1.7</td>
<td>166 ms</td>
<td>35.3 s</td>
</tr>
<tr>
<td>20</td>
<td>3.2</td>
<td>103 ms</td>
<td>51.8 s</td>
</tr>
<tr>
<td>25</td>
<td>2.3</td>
<td>158 ms</td>
<td>of.</td>
</tr>
<tr>
<td>25</td>
<td>3.3</td>
<td>687 ms</td>
<td>of.</td>
</tr>
<tr>
<td>30</td>
<td>2.1</td>
<td>833 ms</td>
<td>of.</td>
</tr>
<tr>
<td>30</td>
<td>3.2</td>
<td>62.9 s</td>
<td>of.</td>
</tr>
<tr>
<td>35</td>
<td>2.2</td>
<td>9.2 min</td>
<td>of.</td>
</tr>
<tr>
<td>35</td>
<td>3.2</td>
<td>of.</td>
<td>of.</td>
</tr>
<tr>
<td>40</td>
<td>1.8</td>
<td>of.</td>
<td>of.</td>
</tr>
<tr>
<td>40</td>
<td>2.7</td>
<td>of.</td>
<td>of.</td>
</tr>
</tbody>
</table>

Note first that times are not always greater for bigger networks. This can be explained by recalling that the size of a BDD is not necessarily directly proportional to the cardinality of the represented set.

We also remark that the instances causing overflow (of.) would presumably produce a result on a computer with more memory.

On the other hand, the times reported for computing stable stationary states with formulas of the form $\text{efp}$ might seem surprisingly low. A reason for these low figures is that such formulas are propositional, and the BDD of the dynamics of the network is not used.

Finally, observe the time in the computation of a basin of attraction when $N = 25$ and $K = 2.3$. In this case, Antelope built a set with $2 \times 10^7$ states in 112 ms. on average.

References

