CONTENTS

APPENDIX A

CALCULATING k_e^*

In this appendix, we describe a method to calculate the k_e^* of a Boolean automaton (Boolean function).

A.1 CONCEPTS

The central concepts required to understand the formulation of k_e^* are defined in Chapters 1 and 3. Below, we introduce additional concepts necessary to understand the procedure for calculating k_e^*.

Cubes and subcubes. As described in Chapter 1, every k-input Boolean function can be represented as a k-dimensional hypercube or simply *cube*. Every such cube contains *subcubes* of every possible dimension from k down to 0 — the entire cube is the only subcube of dimension k, and every individual input vector (corner) is a subcube of dimension 0. Every subcube can be uniquely identified by an *identifier* that consists of input variables and their values that remain constant in the subcube. For example, consider the following 2-dimensional subcube consisting of the following input vectors in a 3-dimensional cube: $\{(0,1,0),(0,1,1),(1,1,0),(1,1,1)\}$. In this subcube, the only input variable whose value is a constant is i_2 with value 1. Thus, the identifier of this subcube is: $(i_2 = 1)$. A subcube identifier could involve multiple input variables, e.g., $(i_1 = 0, i_3 = 1, i_4 = 0)$. As a special case, the identifier of an input vector is that vector itself. A set of *parallel subcubes* consists of subcubes whose identifiers contain the same set of input variables; each subcube is distinguished by the values of the identifier variables. The following is a single set of parallel subcubes, for example: $\{(i_1 = 0, i_3 = 0), (i_1 = 0, i_3 = 1), (i_1 = 1, i_3 = 0), (i_1 = 1, i_3 = 1)\}$. The dimension of a subcube is equal to k minus the number of its identifier variables. In the parallel subcube above, $k = 4$ means that then dimension of every subcube in the set is equal to 2. For a cube of dimension k, the number of possible sets of parallel subcubes of dimension D is equal to $C(k, D)$. For example, in a $k = 3$ cube, the number of sets of parallel subcubes of dimension $D = 2$ is equal to $C(3, 2) = 3$ — the 3 pairs of parallel faces (left-right, front-back, top-bottom) of the cube.

Composite schemata. A *composite schema* is a symmetric schema with at least one fixed (non-permuting) '#' and at least one permuting permuting '#'; it is named so because it contains the characteristic features of both an ordinary schema and a two-symbol schema. All other symmetric schemata shall be referred to as *ordinary symmetric schemata*. For example, $(1, \#, \dot{0}, \dot{\#})$ is a composite schema, whereas $(1, 0, \dot{0}, \dot{\#})$ or $(1, 0, \dot{\#}, \dot{\#})$ are ordinary symmetric schemata. One way to identify a composite schema is by combining two or more symmetric schemata from a set of parallel subcubes. For example, $(\#, \dot{0}, \dot{1}, \dot{\#})$ can be obtained by combining $(0, \dot{0}, \dot{1}, \dot{\#})$ and $(1, \dot{0}, \dot{1}, \dot{\#})$, or by combining $(0, \dot{0}, \dot{\#}, \dot{\#})$ and $(1, \dot{1}, \dot{\#}, \dot{\#})$. Notice in the former that the composite schema is a full union of the combining schemata, whereas in the latter it is a union of portions of the combining schemata. In either case, the composite schema acts as "bridge" that unites ordinary parallel symmetric schemata — this is essentially why we consider parallel subcubes. Also, two symmetric schemata can combine to more than one composite schemata. For example, the symmetric schemata in the set $\{(0, 0, \dot{0}, \dot{\#}), (0, 1, \dot{0}, \dot{\#}), (1, 0, \dot{0}, \dot{\#})\}$ combine to produce the following set of composite schemata: $\{(0, \#, \dot{0}, \dot{\#}), (\#, 0, \dot{0}, \dot{\#})\}$. Naturally, if the original set was $\{(0, 1, \dot{0}, \dot{\#}), (1, 0, \dot{0}, \dot{\#})\}$, no composite schemata would be possible since the values of subcube identifier variables (i_1 and i_2)

can't combine.

A.2 METHOD

The procedure for calculating k_e^* involves the following steps:

Step 0: Initialize a "cover list" of length 2^k with all zeros — this list shall contain the dimensions of the largest covering symmetric schemata corresponding to each input vector, and will be updated throughout the procedure.

Repeat steps 1 to 4 below for every possible dimension D in decreasing order from k to 1, and for every possible set of parallel cubes of a given dimension.

Step 1: Consider a single set of parallel subcubes of a given dimension.

Step 2: Identify all ordinary symmetric schemata in each subcube in the set obtained in the previous step and for each output value.

Note that a subcube may contain more than one symmetric schema. Every symmetric schema in a subcube of dimension D must contain exactly D permuting symbols. One way to identify a symmetic schema is to compute the number of input vectors in the subcube that contain a certain number of 1s (n_1). If that number is equal to $C(D, n_1)$, then those input vectors may constitute a symmetric schema. Note that we identify only those symmetric schemata that cover input vectors with the associated number of 1s in some interval $[n_1, n_1 + w]$ where $w \geq 1$.

Step 3: Identify all composite schemata from the set of ordinary symmetric schemata obtained in the previous step and for each output value.

One way to identify a composite schema is to enumerate "signatures" of all possible ordinary symmetric schemata of dimension D and match them against the permuting symbols of the schemata obtained in the previous step. If more than one symmetric schema in the set matches a given signature, then it is an indication that they might combine to form a composite schema. An important point to note here is that a signature may only *partially* match the schemata and yet produce a composite schema. A representative example is: the signature $(\dot{0}, \dot{1}, \dot{\#})$ partially matches the permuting symbols of both the symmetric schemata $(0, \dot{0}, \dot{\#}, \dot{\#})$ and $(1, \dot{1}, \dot{\#}, \dot{\#})$, to produce the composite schema $(\#, \dot{0}, \dot{1}, \dot{\#})$.

Step 4: Record the dimension of the symmetric schema against every input vector it covers if and only if the current largest covering symmetric schema's dimension is smaller.

Finally, compute k_r^* by averaging over the covering dimensions of all 2^k input vectors. Compute $k_e^* = k - k_r^*$.

A.3 EXAMPLE

In this section, we apply the procedure described above to calculate the k_e^* of an example $k = 3$ function (Fig. A.1), and describe the steps involved in detail.

Initialize cover list: $(0, 0, 0, 0, 0, 0, 0, 0)$, where the order is the same as the numbers indicated in red in the figure.

D = 3:

Step 1. Since $D = k$, the only subcube in the set of parallel subcubes is the full cube itself.

Step 2. The only symmetric schema in this subcube is: $\{(\dot{1}, \dot{1}, \dot{\#})\}$, corresponding to output 1.

Step 3. No composite schemata exist since there is only symmetric schema.

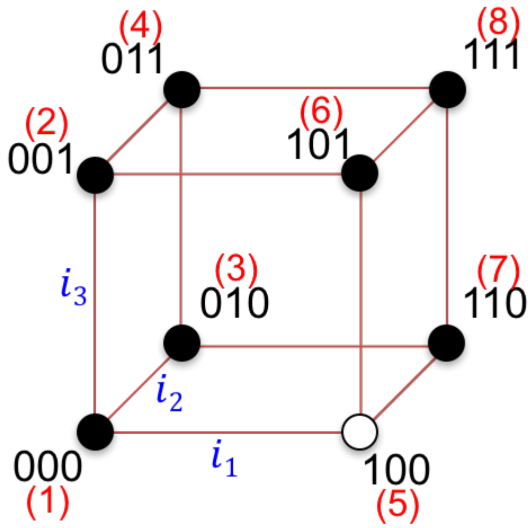


Figure A.1: An example $k = 3$ Boolean function $x = f(i_1, i_2, i_3)$. Each corner (input vector) is marked with a decimal number in red — to be used as an index to refer to the input vectors in the cover list.

Step 4. The set of all symmetric schemata now consists of $\{(\dot{1}, \dot{1}, \dot{\#})\}$, with dimension equal to $\log_2(4) = 2$, and covering the corners {(4,6,7,8)}. The updated cover list is: (0,0,0,2,0,2,2,2).

D = 2:

There are 3 sets of parallel subcubes: $\{(i_1 = 0), (i_1 = 1)\}$, $\{(i_2 = 0), (i_2 = 1)\}$ and $\{(i_3 = 0), (i_3 = 1)\}$.

Step 1a. Consider the set of parallel subcubes $\{(i_1 = 0), (i_1 = 1)\}$.

Step 2a. The set of symmetric schema in this set of parallel subcubes consists of: $\{(0, \dot{\#}, \dot{\#})$, $(1, \dot{1}, \dot{\#})\}$, corresponding to output 1.

Step 3a. The set of composite schemata obtained by combining the symmetric schemata in the set from above consists of: $\{(\#, \dot{1}, \dot{\#})\}$.

Step 4a. The set of all symmetric schemata now consists of $\{(0, \dot{\#}, \dot{\#}), (1, \dot{1}, \dot{\#}), (\#, \dot{1}, \dot{\#})\}$, with dimensions respectively equal to $\log_2(4) = 2$, $\log_2(3) \approx 1.58$ and $\log_2(6) \approx 2.58$, and covering the corners {(1,2,3,4),(6,7,8),(2,3,4,6,7,8)} respectively. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58).

Step 1b. Consider the set of parallel subcubes $\{(i_2 = 0), (i_2 = 1)\}$.

Step 2b. The set of symmetric schema in this set of parallel subcubes consists of: $\{(\dot{\#}, 1, \dot{\#})\}$, corresponding to output 1.

Step 3b. No composite schemata exist since there is only symmetric schema.

Step 4b. The set of all symmetric schemata now consists of $\{(\dot{\#}, 1, \dot{\#})\}$, with a dimensions equal to $\log_2(4) = 2$, and covering the corners {(3,4,7,8)}. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

Step 1c. Consider the set of parallel subcubes $\{(i_3 = 0), (i_3 = 1)\}$.

Step 2c. The set of symmetric schema in this set of parallel subcubes consists of: $\{(\dot{\#}, \dot{\#}, 1)\}$, corresponding to output 1.

Step 3c. No composite schemata exist since there is only symmetric schema.

Step 4c. The set of all symmetric schemata now consists of $\{(\dot{\#}, \dot{\#}, 1)\}$, with a dimension equal to $\log_2(4) = 2$, and covering the corners {(2,4,6,8)}. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

D = 1:

There are 3 sets of parallel subcubes: $\{(i_1 = 0, i_2 = 0), (i_1 = 0, i_2 = 1), (i_1 = 1, i_2 = 0), (i_1 = 1, i_2 = 1)\}$, $\{(i_1 = 0, i_3 = 0), (i_1 = 0, i_3 = 1), (i_1 = 1, i_3 = 0), (i_1 = 1, i_3 = 1)\}$ and $\{(i_2 = 0, i_3 = 0), (i_2 = 0, i_3 = 1), (i_2 = 1, i_3 = 0), (i_2 = 1, i_3 = 1)\}$.

Step 1a. Consider the set of parallel subcubes $\{(i_1 = 0, i_2 = 0), (i_1 = 0, i_2 = 1), (i_1 = 1, i_2 =$

$0), (i_1 = 1, i_2 = 1)\}$.

Step 2a. The set of symmetric schema in this set of parallel subcubes consists of: $\{(0, 0, \dot{\#})$, $(0, 1, \dot{\#}), (1, 1, \dot{\#}\}$, corresponding to output 1.

Step 3a. No composite schemata exist. Note that $(0, 0, \dot{\#})$ and $(0, 1, \dot{\#})$ can combine to form $(0, \dot{\#}, \#)$ but it is not a valid composite schema (see definition in Sec. A.1).

Step 4a. The set of all symmetric schemata now consists of $\{(0, 0, \dot{\#}), (0, 1, \dot{\#}), (1, 1, \dot{\#}\}$, with dimensions respectively equal to $\log_2(2) = 1$ each, and covering the corners {(1,2),(3,4),(7,8)} respectively. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58)s (no alteration).

Step 1b. Consider the set of parallel subcubes $\{(i_1 = 0, i_3 = 0), (i_1 = 0, i_3 = 1), (i_1 = 1, i_3 = 0), (i_1 = 1, i_3 = 1)\}$.

Step 2b. The set of symmetric schema in this set of parallel subcubes consists of: $\{(0, \dot{\#}, 0)$, $(0, \dot{\#}, 1), (1, \dot{\#}, 1)\}$, corresponding to output 1.

Step 3b. No composite schemata exist.

Step 4b. The set of all symmetric schemata now consists of $\{(0, \dot{\#}, 0), (0, \dot{\#}, 1), (1, \dot{\#}, 1\}$, with dimensions respectively equal to $\log_2(2) = 1$ each, and covering the corners {(1,3),(2,4),(6,8)} respectivelu. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

Step 1c. Consider the set of parallel subcubes $\{(i_2 = 0, i_3 = 0), (i_2 = 0, i_3 = 1), (i_2 = 1, i_3 = 0), (i_2 = 1, i_3 = 1)\}$.

Step 2c. The set of symmetric schema in this set of parallel subcubes consists of: $\{(\dot{\#}, 0, 1)$, $(\dot{\#}, 1, 1), (\dot{\#}, 1, 0)\}$, corresponding to output 1.

Step 3c. No composite schemata exist.

Step 4c. The set of all symmetric schemata now consists of $\{(\dot{\#}, 0, 1), (\dot{\#}, 1, 1), (\dot{\#}, 1, 0)\}$, with dimensions respectively equal to $\log_2(2) = 1$ each, and covering the corners {(2,6),(4,8),(3,7)} respectively. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

Finally, k_r^* is the mean of the values in the cover list: $k_r^* = 2.185 \implies k_e^* = k - k_r^* = 0.815$ (exact value is 0.8112781 if the log values above are not rounded).

A.4 SOURCE CODE

An implementation of the above is available in **R**; the link to the *GitHub* repository is listed in Ref. [59]. The main files are:

1. *ComputeGeneralizedKeff.R*: The main file containing an implementation of the procedure to compute k_e^*.
2. *ComputeDetectCubes.R*: A supporting file containing an implementation of a part of Step 3 of the procedure that helps identify composite schemata. Specifically, it helps identify the non-permuting wildcard symbols in a symmetric schema.

APPENDIX B

INTEGRATING A BOOLEAN NETWORK

In this appendix, we describe a method to integrate a Boolean network.

B.1 CONCEPTS

The central concepts and notations required to understand the integration procedure are described in Chapter 4. Below, we introduce additional concepts necessary to understand the details of the procedure.

Sets of schemata. A 'set' of schemata is defined as a set where the logical condition specified by at least of one of the schemata is true. In other words, a set of schemata specifies a logical condition in the form of a disjunction of conjunctive clauses. A set of schemata naturally redescribes a set of LUT entries. For example, $\{10\#, \#\#1\}$ specifies the logical condition: $(x_1 = 1 \wedge x_2 = 0) \vee (x_3 = 1)$, and redescribes the set of LUT entries $\{(1,0,0),(1,0,1),(0,0,1),(0,1,1),(1,1,1)\}$.

Union of a set of schemata. 'Union' is defined as a unary operation on a set of schemata that returns the set of *all* possible *minimal* schemata which jointly cover all of the LUT entries that the original set covers. In other words, the union of a set of implicants or prime implicants is the set of *all* prime implicants that covers the same set of LUT entries that the original set does. The result of an union could comprise more or fewer schemata than the original set depending on its composition. For example, the union of the set $\{00\#, 01\#\}$ is the set $\{0\#\#\}$; whereas, the union of $\{0\#0, 11\#\}$ is the set $\{0\#0, \#10, 11\#\}$. The union operation is equivalent to converting all the schemata in the original set into LUT entries first and then compressing it using a standard logic minimization method such as Quine-McCluskey to obtain the set of all prime implicants. This latter procedure is clearly inefficient since it involves "decompression" first followed by a compression from scratch; utilizing the compression that already comes with the original set of schemata would be more efficient. In fact, the union operation could be thought of as a nonlinear extension of Quine-McCluskey because, in the latter smaller schemata (fewer wildcards) combine to form only larger schemata, whereas a union of larger schemata could result in smaller schemata as well. For example, the union of the set $\{00\#0\#0, \#\#111\#\}$ is the set $\{00\#0\#0, \#\#111\#, 001\#10\}$ (notice the additional schema with just one wildcard).

Intersection of sets of schemata. 'Intersection' is defined as a binary operation on a pair of sets of schemata that returns a single set of *all* possible *minimal* schemata which jointly cover all of the LUT entries common to both sets. In other words, the intersection of a pair of sets of implicants or prime implicants is the set of all *all* prime implicants that covers the set of all LUT entries that both the intersecting sets cover. Thus, the intersection operation is nothing but an implementation of the distributive law of Boolean algebra [24]. The result of an intersection is the empty set $\{\phi\}$ if the intersecting sets have no LUT entries in common. Here are a few examples: $\{\#1\#\}$ intersection $\{1\#\#\} = \{11\#\}$; $\{\#\#1\}$ intersection $\{11\#, \#11\} = \{\#11\}$; $\{1\#\#, \#00\}$ intersection $\{\#11, 11\#\} = \{11\#\}$; $\{\#\#1\}$ intersection $\{10\#, \#11\} = \{1\#1, \#11\}$; $\{1\#\#, \#\#1\}$ intersection $\{\#1\#\} = \{11\#, \#11\}$; $\{0\#\}$ intersection $\{1\#\} = \{\phi\}$; and $\{0\#\}$ intersection $\{1\#, \#1\} = \{01\}$. Naturally, any number of sets of schemata can be intersected by intersecting the first pair of sets, then replacing the pair in the original set with their intersection, and continue so on until a single (potentially empty) set of schemata remains.

B.2 METHOD

The procedure for integrating a BN involves the following steps:

For each node and for each output (0 and 1), repeat the following steps for a specified number of integration steps:

Step 1: Compute the set of predecessor schemata for each schema mapping to the given output.

The set of predecessor schemata of a given schema is the intersection of the sets of predecessor schemata of the atomic schemata associated with its individual literals.

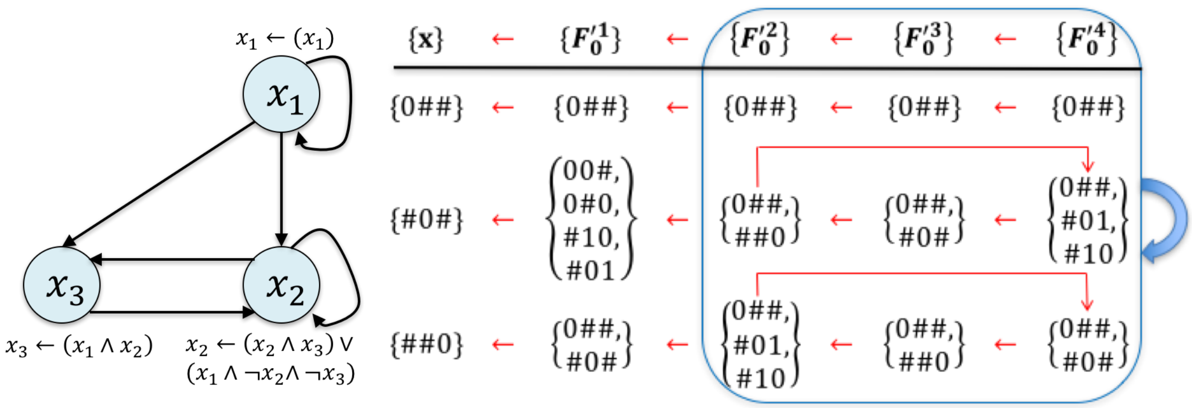
Step 2: Compress the set of all predecessor schemata sets obtained in step 1.

This set is just the union of the set of all predecessor schemata sets obtained in step 1.

B.3 EXAMPLE

In this section, we apply the procedure described above to integrate the example BN described in Chapter 4, and describe the steps involved in detail. For clarity, we repeat the BN and the associated schema causation chain (SCC) for output 0 in Fig.B.1.

For simplicity, we only describe the steps involved in the first step of integration for the atomic schemata containing a '0'. That is, we only show how to compute $F_0^{\prime 2}$. Note that $F_0^{\prime 1}$ comprises the set of input schemata of each node corresponding to output 0, which we list below (note that



(a) BN

(b) SCC

Figure B.1: An example BN and the SCC corresponding to atomic schemata containing a '0'.

all schemata have $n = 3$ symbols; symbols corresponding to nodes that are not inputs are just wildcards):

1. Node x_1: $\{0\#\#\} \mapsto 0$; $\{1\#\#\} \mapsto 1$.
2. Node x_2: $\{00\#, 0\#0, \#10, \#01\} \mapsto 0$; $\{100, \#11\} \mapsto 1$.
3. Node x_3: $\{0\#\#, \#0\#\} \mapsto 0$; $\{11\#\} \mapsto 1$.

We now proceed to calculating $F_0^{'2}$.

Node $= x_1$, output $= 0$:

Step 1. Compute the set of predecessor schemata for each schema in the set $\{0\#\#\}$ and intersect the resulting sets.

> Step 1a. Compute the predecessor schemata of $\{0\#\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$. Intersection is not necessary since there are no more predecessor schemata to retrieve. Therefore, the result of intersection is: $\{0\#\#\}$.

Step 2. Compress the set of all predecessor schemata obtained in step 1: $\{0\#\#\}$. Since there is only one schema in the set, the result of union is: $\{0\#\#\}$.

Therefore, $F_0^{'2}$ for node x_1 is: $\{0\#\#\}$.

Node = x_2, output = 0:

Step 1. Compute the set of predecessor schemata for each schema in the set $\{00\#, 0\#0, \#10, \#01\}$ and intersect the resulting sets.

Step 1a. Compute the predecessor schemata of $\{00\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$; predecessor schemata of $x_2 = 0$: $\{00\#, 0\#0, \#10, \#01\}$. Then, intersect the sets $\{0\#\#\}$ and $\{00\#, 0\#0, \#10, \#01\}$. The result of intersection is: $\{00\#, 0\#0\}$.

Step 1b. Compute the predecessor schemata of $\{0\#0\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$; predecessor schemata of $x_3 = 0$: $\{0\#\#, \#0\#\}$. Then, intersect the sets $\{0\#\#\}$ and $\{0\#\#, \#0\#\}$. The result of intersection is: $\{0\#\#\}$.

Step 1c. Compute the predecessor schemata of $\{\#10\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_2 = 1$: $\{100, \#11\}$; predecessor schemata of $x_3 = 0$: $\{0\#\#, \#0\#\}$. Then, intersect the sets $\{100, \#11\}$ and $\{0\#\#, \#0\#\}$. The result of intersection is: $\{100, 011\}$.

Step 1d. Compute the predecessor schemata of $\{\#01\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_2 = 0$: $\{00\#, 0\#0, \#10, \#01\}$; predecessor schemata of $x_3 = 1$: $\{11\#\}$. Then, intersect the sets $\{00\#, 0\#0, \#10, \#01\}$ and $\{11\#\}$. The result of intersection is: $\{110\}$.

Step 2. Compress the set of all predecessor schemata obtained in step 1: $\{00\#, 0\#0, 0\#\#, 100, 011,$

$110\}$. The result of the union is: $\{\#\#0, 0\#\#\}$.

Therefore, $F_0^{'2}$ for node x_2 is: $\{\#\#0, 0\#\#\}$.

Node = x_3, output = 0:

Step 1. Compute the set of predecessor schemata for each schema in the set $\{0\#\#, \#0\#\}$ and intersect the resulting sets.

Step 1a. Compute the predecessor schemata of $\{0\#\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$. Intersection is not necessary since there are no more predecessor schemata to retrieve. Therefore, the result of intersection is: $\{0\#\#\}$.

Step 1b. Compute the predecessor schemata of $\{\#0\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_2 = 0$: $\{00\#, 0\#0, \#10, \#01\}$. Intersection is not necessary since there are no more predecessor schemata to retrieve. Therefore, the result of intersection is: $\{00\#, 0\#0, \#10, \#01\}$.

Step 2. Compress the set of all predecessor schemata obtained in step 1: $\{0\#\#, 00\#, 0\#0, \#10, \#01\}$. The result of the union is: $\{0\#\#, \#10, \#01\}$.

Therefore, $F_0^{'2}$ for node x_3 is: $\{0\#\#, \#10, \#01\}$.

This completes the computation of $F_0^{'2}$ for all three nodes in the BN.

B.4 SOURCE CODE

An implementation of the above is available in **R**; the link to the *GitHub* repository is listed in Ref. [59]. The main files are:

1. *ComputeIntegrateBoolNet.R*: The main file containing an implementation of the Boolean network integration procedure.
2. *ComputeSchemaSetOperations.R*: A supporting file containing implementations of the union and intersection procedures.
3. *ComputeIntegratedKeff.R*: Contains an implementation of computing k_e of $F^{'t}$ using the output of the integration procedure.