On Knowledge Compilation Languages for QBFs

Sravanthi Chede[1] Vaibhav Krishan[1] Swagato Sanyal[2] **Anil Shukla**[3]

[1]The Institute of Mathematical Sciences, Chennai
[2]The University of Sheffield, United Kingdom
[3]Indian Institute of Technology Ropar

QBF Workshop 2026
Lisbon, Portugal
July 19, 2026

Quantified Boolean Formulas

QBF = Quantifier prefix + Propositional formula (CNF or logical statement)

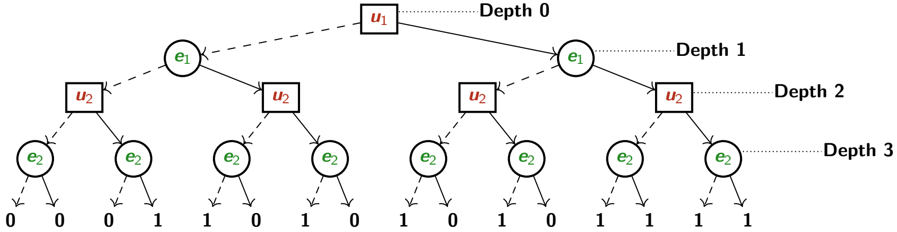
$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

Quantified Boolean Formulas

QBF = Quantifier prefix + Propositional formula (CNF or logical statement)

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

Quantified Tree: dashed edges for 0, solid edges for 1 (in this talk).

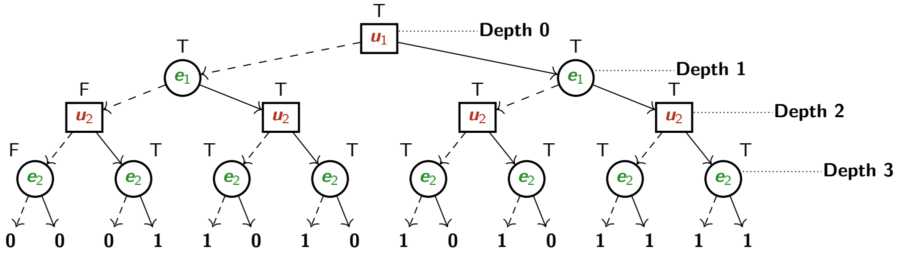


Quantified Boolean Formulas

QBF = Quantifier prefix + Propositional formula (CNF or logical statement)

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

Quantified Tree: dashed edges for 0, solid edges for 1 (in this talk).

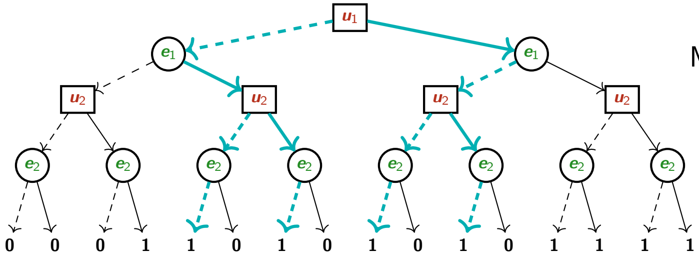


Satisfiability:

- existential node $\rightarrow \vee$
- universal node $\rightarrow \wedge$

Quantified Tree

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$



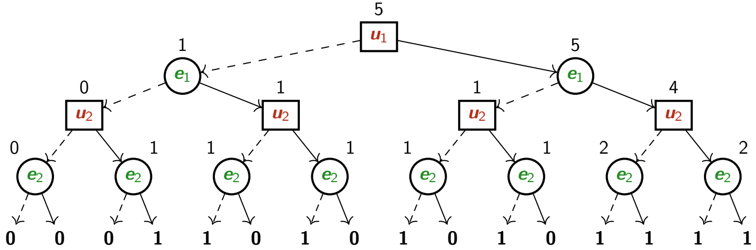
Model Tree:

- Existential node has 1 outgoing edge
- Universal node has both outgoing edges
- Contains the root node
- All leaves are 1

$\mathcal{S} = \{e_1 \leftarrow \overline{u_1}, e_2 \leftarrow 0\}$

Quantified Tree (contd.)

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$



#QBF = Number of model trees:

existential node $\rightarrow +$

universal node $\rightarrow \times$

$$\mathcal{S}_1 = \{e_1 \leftarrow \overline{u_1},\ e_2 \leftarrow 0\}$$
$$\mathcal{S}_2 = \{e_1 \leftarrow 1,\ e_2 \leftarrow 0\}$$
$$\mathcal{S}_3 = \{e_1 \leftarrow 1,\ e_2 \leftarrow u_1\}$$
$$\mathcal{S}_4 = \{e_1 \leftarrow 1,\ e_2 \leftarrow u_1 \wedge u_2\}$$
$$\mathcal{S}_5 = \{e_1 \leftarrow 1,\ e_2 \leftarrow u_1 \wedge \overline{u_2}\}$$

Folklore

For any quantified statement, having the corresponding Quantified Tree makes model counting easy w.r.t. its size.

However, the size of the Quantified Tree is exponential in the number of variables!

We try to truncate it without loosing the #QBF answer:

- If at any internal node v labelled by x_i, the subtree rooted at v has all 1-leaves:
  - Truncate the subtree
  - Find model count at v by running the model count algorithm on the Quantifier Prefix from x_n to x_i.
  - This value will be some 2^x, label v with x.

However, the size of the Quantified Tree is exponential in the number of variables!

We try to truncate it without loosing the #QBF answer:

- If at any internal node v labelled by x_i, the subtree rooted at v has all 1-leaves:
  - Truncate the subtree
  - Find model count at v by running the model count algorithm on the Quantifier Prefix from x_n to x_i.
  - This value will be some 2^x, label v with x.
- If at any node v, the subtree rooted at v corresponds to a False QBF:
  - Truncate the subtree
  - Label v with $0F$ (i.e. FQBF with model count 0)

However, the size of the Quantified Tree is exponential in the number of variables!

We try to truncate it without loosing the #QBF answer:

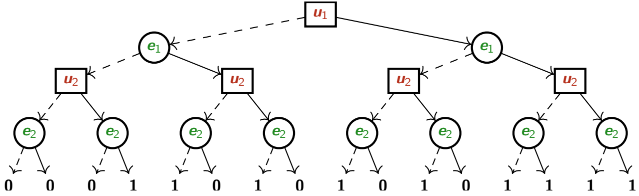
- If at any internal node v labelled by x_i, the subtree rooted at v has all 1-leaves:
  - Truncate the subtree
  - Find model count at v by running the model count algorithm on the Quantifier Prefix from x_n to x_i.
  - This value will be some 2^x, label v with x.
- If at any node v, the subtree rooted at v corresponds to a False QBF:
  - Truncate the subtree
  - Label v with $0F$ (i.e. FQBF with model count 0)
- For an internal node v and its child nodes v_0, v_1,
  if the subtrees rooted at v_0, v_1 correspond to QBFs having the same set of $\exists$-winning strategies.
  - Explore only one child and add double edge from v to that node.

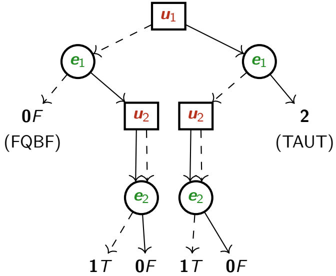
Truncated Quantified Tree (TQT)

$$\Phi = Q.F := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \;.\; (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

TQT: (Q, T)

Quantified Tree:





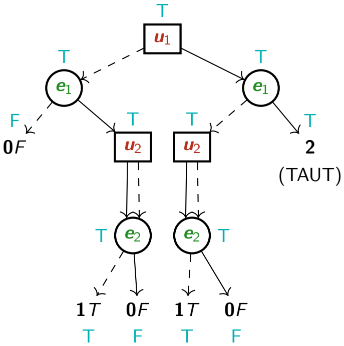
Satisfiability in TQT

A TQT structure $G = (Q, T)$ is true if the existential player can always land on a leaf with label other than '$0F$' of T when players assign values to their respective variables in the sequence of Q.

Satisfiability in TQT

A TQT structure $G = (Q, T)$ is true if the existential player can always land on a leaf with label other than '$0F$' of T when players assign values to their respective variables in the sequence of Q.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$



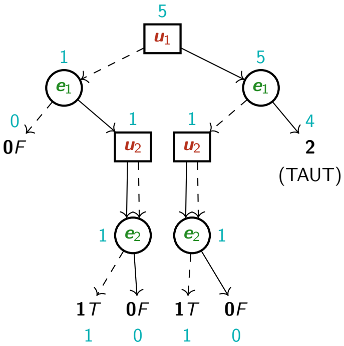
Satisfiability:

- Leaves labelled $0F \to F$
- Other leaves $\to T$
- Existential node $\to \vee$
- Universal node $\to \wedge$

Satisfiability in TQT

A TQT structure $G = (Q, T)$ is true if the existential player can always land on a leaf with label other than '$0F$' of T when players assign values to their respective variables in the sequence of Q.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$



Model count:
- Leaves labelled $0F \rightarrow 0$
- Leaves labelled $1T \rightarrow 1$
- Other leaves $\rightarrow 2^{\text{<node-label>}}$
- Existential node $\rightarrow +$
- Universal node $\rightarrow \times$

Model Tree of TQT: A subtree of TQT containing the root, two outgoing edges for an universal node , one outgoing edge for existential node and none of the leaves are labelled by $0F$.

Every such tree has an associated weightage = Model count (similar to above)

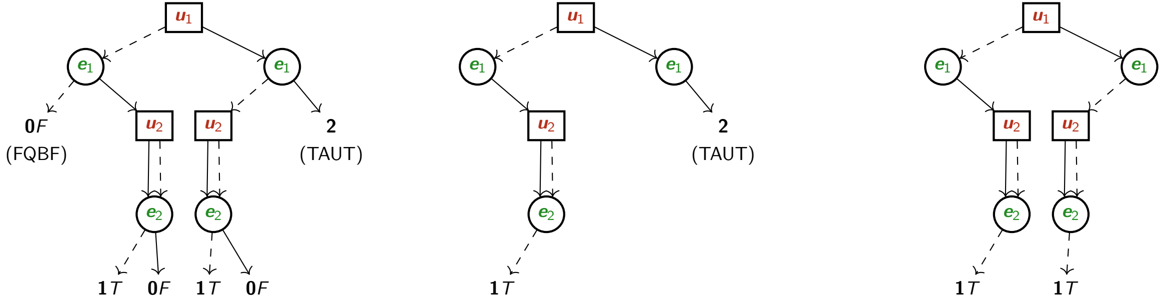
#QBF answer is the sum of all weightages of all model trees of the TQT.

Model Tree of TQT: A subtree of TQT containing the root, two outgoing edges for an universal node , one outgoing edge for existential node and none of the leaves are labelled by $0F$.

Every such tree has an associated weightage = Model count (similar to above)

#QBF answer is the sum of all weightages of all model trees of the TQT.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

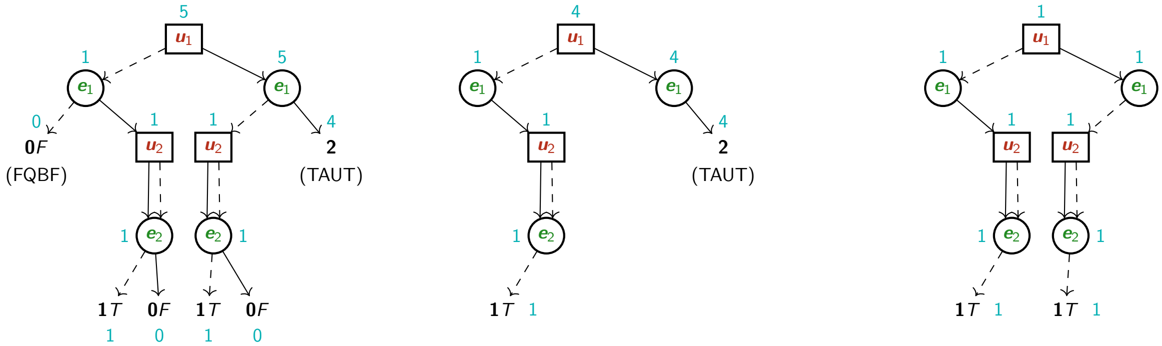


Model Tree of TQT: A subtree of TQT containing the root, two outgoing edges for an universal node , one outgoing edge for existential node and none of the leaves are labelled by $0F$.

Every such tree has an associated weightage = Model count (similar to above)

#QBF answer is the sum of all weightages of all model trees of the TQT.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$



Certified Truncated Quantified Tree

TQT doesn't include certificates for the Tautology, the False QBF nodes and the equivalence nodes.

cert-TQT includes these proofs also.

- To certify that two QBFs are 'Skolem Equivalent', QBF-proof systems are enough (see [Pfeiffer, Große, and Seidl, 2025])

Therefore, given a cert-TQT and the corresponding QBF, it is easy to verify correctness of cert-TQT.

cert-TQT can be seen as:

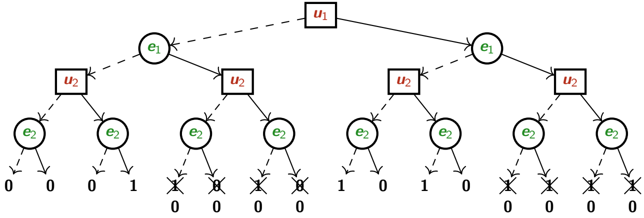
- a knowledge compilation language for QBFs
- a static polynomial time verfiable proof system for #QBF (for QBFs with at most exponential model counts, as opposed to doubly exponential).

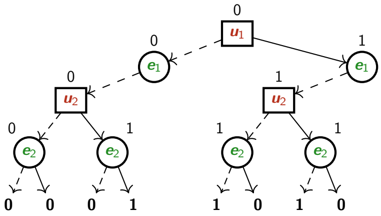
Subtrees and QBF restrictions

$$\begin{aligned}\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \;.\; &(u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \\ &\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})\end{aligned}$$

Restricting existential variables:

$$\begin{aligned}\Phi|_{\{e_1=0\}} := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \;.\; &(u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \\ &\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{e_1})\end{aligned}$$





$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \; . \; (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

Restricting universal variables:

$$\Phi|_{\{u_1=1\}} \neq \forall u_1 \exists e_1 \forall u_2 \exists e_2 \; . \; (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2}) \wedge (u_1) \text{ (False QBF!)}$$

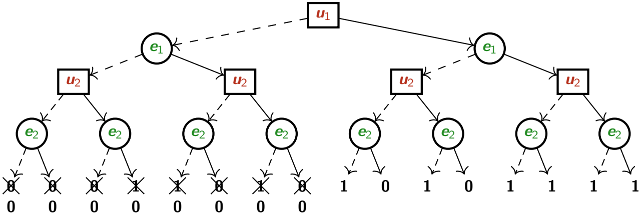
$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

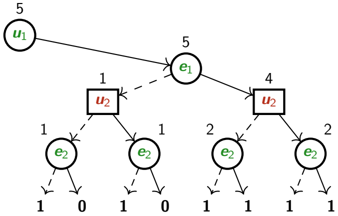
Restricting universal variables:

$$\Phi|_{\{u_1=1\}} \neq \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2}) \wedge (u_1) \text{ (False QBF!)}$$

Making it existential will work:

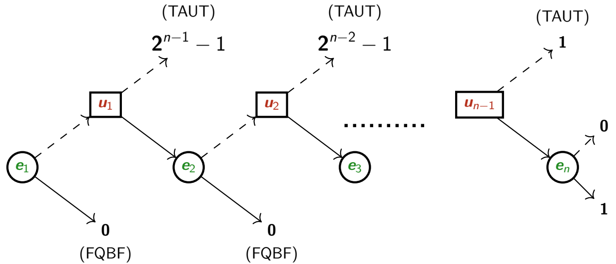
$$\Phi|_{\{u_1=1\}} = \exists u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2) \wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2}) \wedge (u_1)$$





Upper bound for cert-TQT

$$\Phi := \exists e_1 \forall u_1 \exists e_2 \ldots \forall u_{n-1} \exists e_n \,.\, (\overline{e_1} \vee \overline{u_1}) \wedge \bigwedge_{i\in[2,n-1]} (\overline{u_1} \vee \ldots \vee \overline{u_{i-1}} \vee \overline{e_i} \vee \overline{u_i}) \;\wedge (\overline{u_1} \vee \ldots \vee \overline{u_{n-1}} \vee e_n)$$



Model count $= 2^{2^n - n - 1}$

All FQBF nodes: have a universal unit clause in the formula (easy)
All TAUT nodes: the root-to-node path satisfies all clauses

We provide QBF Benchmarks based on these formulas: 'Rec_Alt_Taut_False' instances

Lower bound for TQT

$$\text{Quant-Parity}_n := \exists e_1 \forall u_1 \ldots \forall u_{n-1} \exists e_n \,.\, (e_1 \oplus u_1 \oplus \ldots \oplus u_{n-1} \oplus e_n)$$

Theorem

Any TQT *representing Quant-Parity$_n$ requires size exponential in n.*

Proof: Truncation of Quantified Tree not possible until depth $2n-2$.

Certified Truncated Quantified DAG

TQD is a generalized TQT with DAG-edges of the following form:

- two or more outgoing edges from nodes in the same depth can go to the same child node.

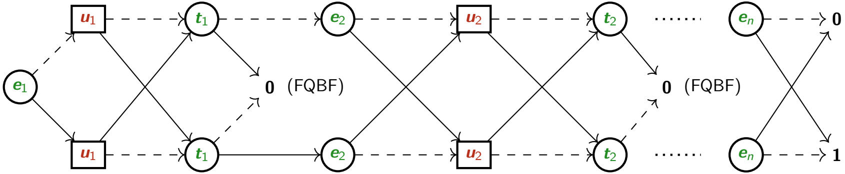
cert-TQD is a generalized cert-TQT as it includes:

- proofs that all root-to-node paths on the DAG-edges are Skolem equivalent.

Upper bounds for cert-TQD

Quant-Parity$_n$ can be encoded as a QBF using Tseitin variables:

$$\text{TQBF-Parity}_n = \exists e_1 \forall u_1 \exists t_1 \exists e_2 \forall u_2 \exists t_2 \ldots \exists t_{n-1} \exists e_n \; . \; (t_1 \leftrightarrow e_1 \oplus u_1) \wedge \bigwedge_{i \in [2, n-1]} (t_i \leftrightarrow t_{i-1} \oplus e_i \oplus u_i) \; \wedge (t_{n-1} \oplus e_n)$$

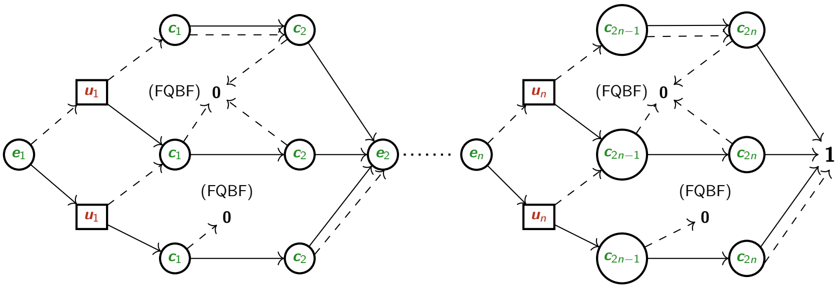


Theorem

The languages of TQD *and* cert-TQD *are more succinct than* TQT *and* cert-TQT*.*

Upper bounds for cert-TQD (contd.)

$$\text{TQBF-}\phi_n := \exists e_1 \forall u_1 \exists c_1, c_2 \ldots \exists e_n \forall u_n \exists c_{2n-1}, c_{2n} \,.\, \bigwedge_{i\in[n]} (\overline{e_i} \vee c_{2i-1}) \wedge (\overline{u_i} \vee c_{2i-1}) \wedge (e_i \vee c_{2i}) \wedge (u_i \vee c_{2i})$$



Model count $= 2^{2^{n+1}-2}$

All FQBF nodes: have a Falsified clause in the formula (easy)
All Equivalent edges: corresponding QBF formulas are identical

We provide QBF Benchmarks based on these formulas: ‘Phi_True’ instances

Lower bounds for TQD

QXOR-pairs are known to be easy for the #QBF proof system Q-MICE [Chede, Chew, Krishan, S, SAT-2026].

$$\text{QXOR-pairs} := \forall u_1, \ldots, u_n \underset{i \neq j \in [n]}{\exists} x_{i,j} \cdot \bigwedge_{i \neq j \in [n]} (x_{i,j} \leftrightarrow u_i \oplus u_j)$$

Theorem

Any TQD *representing QXOR-pairs requires size exponential in n.*

Proof: Need to branch individually on all $u_1, \ldots, u_n$ variables in the starting of the TQD.

Q-MICE p-simulates cert-TQD

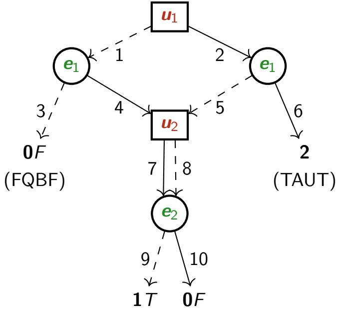
A proof in Q-MICE consists of claims in the form $(Q.F|_\rho,\ A,\ c)$ where:

- $Q.F$ is the input QBF
- ρ is partial assignment to universal variables
- A is a partial strategy i.e. some existential variables are assigned
- c is then the correct number of models for QBF $Q.F|_{\rho \cup A}$

However, here we just need A to be partial assignment to (some) existential variables.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

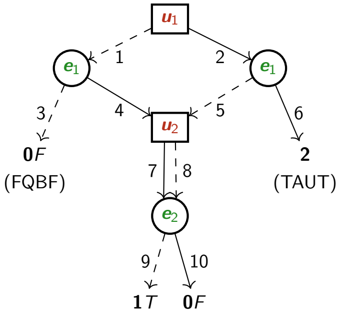
Claim no.	Rule	Claim	TQD Edge no.
C_1	Axiom	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=0\},\ 1)$	9



Axiom-Rule: $\frac{}{(Q.F|_\rho,\ \alpha,\ c)}$, when $F|_{\rho\cup\alpha}$ is a Tautology and c is the correct model count of the corresponding subtree.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

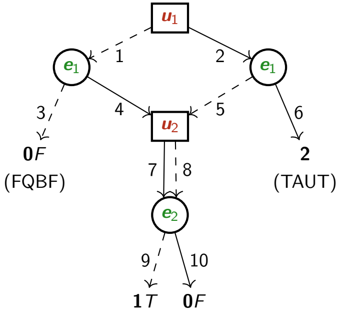
Claim no.	Rule	Claim	TQD Edge no.
C_1	Axiom	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1 = 1, e_2 = 0\},\ 1)$	9
C_2	Absence of models	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1 = 1, e_2 = 1\},\ 0)$	10



Absence of models rule: $\overline{(Q.F|_\rho,\ \alpha,\ 0)}$ when there is explicit proof that $Q.F|_{\rho\cup\alpha}$ is a False QBF.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

Claim no.	Rule	Claim	TQD Edge no.
C_1	Axiom	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=0\},\ 1)$	9
C_2	Absence of models	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=1\},\ 0)$	10
C_3	Composition (C_1, C_2)	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1\},\ 1)$	8

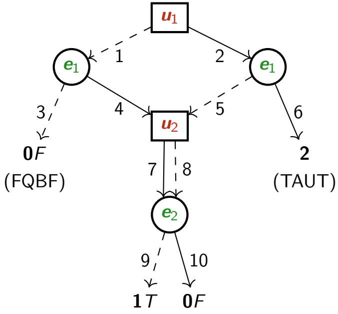


Composition rule: $\dfrac{(Q.F|_\rho,\ \alpha \cup \{y{=}0\},\ c_1) \quad (Q.F|_\rho,\ \alpha \cup \{y{=}1\},\ c_2)}{(Q.F|_\rho,\ \alpha,\ c_1{+}c_2)}$ can be used when,

$\rho \cup \alpha$ assigns all and only variables to the left of y in the prefix Q.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

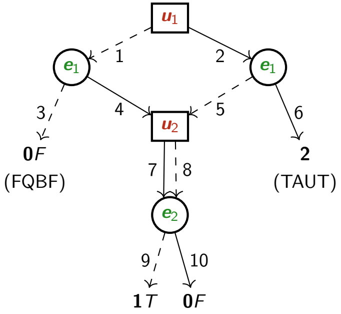
Claim no.	Rule	Claim	TQD Edge no.
C_1	Axiom	$(\Phi\vert_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=0\},\ 1)$	9
C_2	Absence of models	$(\Phi\vert_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=1\},\ 0)$	10
C_3	Composition (C_1, C_2)	$(\Phi\vert_{\{u_1=0,u_2=0\}},\ \{e_1=1\},\ 1)$	8
C_4	$\simeq C_3$	$(\Phi\vert_{\{u_1=0,u_2=1\}},\ \{e_1=1\},\ 1)$	7



Equivalence rule: $\dfrac{(Q.F|_{\rho_1},\ \alpha_1,\ c)}{(Q.F|_{\rho_2},\ \alpha_2,\ c)}$ when there is explicit proof that the QBFs are Skolem equivalent

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

Claim no.	Rule	Claim	TQD Edge no.
C_1	Axiom	$(\Phi\vert_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=0\},\ 1)$	9
C_2	Absence of models	$(\Phi\vert_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=1\},\ 0)$	10
C_3	Composition (C_1, C_2)	$(\Phi\vert_{\{u_1=0,u_2=0\}},\ \{e_1=1\},\ 1)$	8
C_4	$\simeq C_3$	$(\Phi\vert_{\{u_1=0,u_2=1\}},\ \{e_1=1\},\ 1)$	7
C_5	Join (C_3, C_4)	$(\Phi\vert_{\{u_1=0\}},\ \{e_1=1\},\ 1)$	4

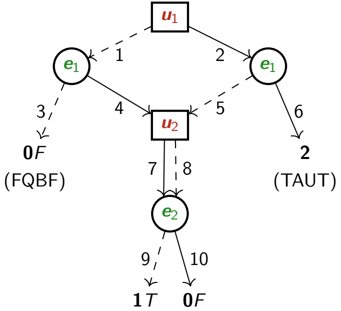


Join rule: $\dfrac{(Q.F\vert_{\rho\cup y=0},\ \alpha,\ c_1) \quad (Q.F\vert_{\rho\cup y=1},\ \alpha,\ c_2)}{(Q.F\vert_{\rho},\ \alpha,\ c_1 \cdot c_2)}$ can be used when,

$\rho \cup \alpha$ assign all and only variables to the left of y in the prefix Q.

$$\Phi := \forall u_1 \exists e_1 \forall u_2 \exists e_2 \,.\, (u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee e_1 \vee \overline{u_2} \vee e_2)$$
$$\wedge (u_1 \vee \overline{e_1} \vee u_2 \vee \overline{e_2}) \wedge (u_1 \vee \overline{e_1} \vee \overline{u_2} \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee u_2 \vee \overline{e_2}) \wedge (\overline{u_1} \vee e_1 \vee \overline{u_2} \vee \overline{e_2})$$

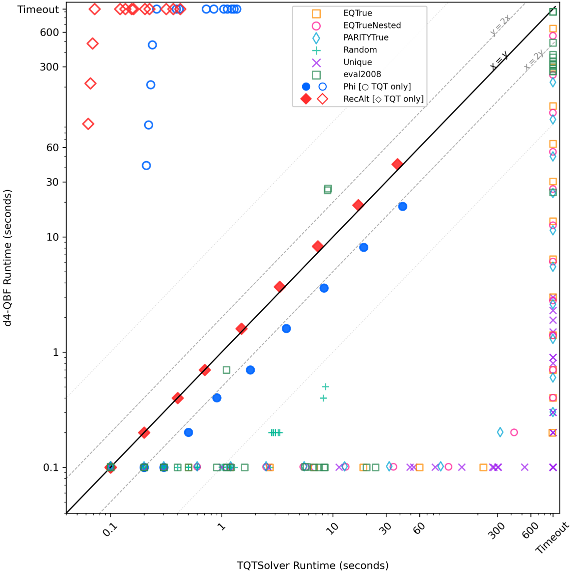
Claim no.	Rule	Claim	TQD Edge no.
C_1	Axiom	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=0\},\ 1)$	9
C_2	Absence of models	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1, e_2=1\},\ 0)$	10
C_3	Composition (C_1, C_2)	$(\Phi\|_{\{u_1=0,u_2=0\}},\ \{e_1=1\},\ 1)$	8
C_4	$\simeq C_3$	$(\Phi\|_{\{u_1=0,u_2=1\}},\ \{e_1=1\},\ 1)$	7
C_5	Join (C_3, C_4)	$(\Phi\|_{\{u_1=0\}},\ \{e_1=1\},\ 1)$	4
C_6	Absence of models	$(\Phi\|_{\{u_1=0\}},\ \{e_1=0\},\ 0)$	3
C_7	Composition (C_5, C_6)	$(\Phi\|_{\{u_1=0\}},\ \emptyset,\ 1)$	1
C_8	Axiom	$(\Phi\|_{\{u_1=1\}},\ \{e_1=1\},\ 2^2)$	6
C_9	$\simeq C_5$	$(\Phi\|_{\{u_1=1\}},\ \{e_1=0\},\ 1)$	5
C_{10}	Composition (C_8, C_9)	$(\Phi\|_{\{u_1=1\}},\ \emptyset,\ 5)$	2
C_{11}	Join (C_7, C_{10})	$(\Phi,\ \emptyset,\ 5)$	-

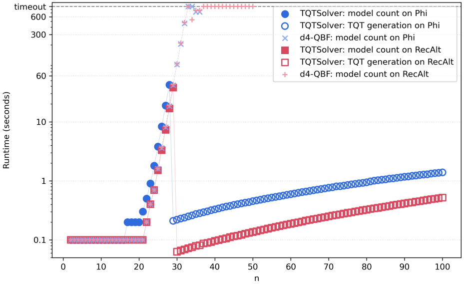


Experimental Setup

- We have implemented a solver denoted as TQTSolver.
- For an input QBF, TQTSolver first builds a TQT representation. Followed by computing the model count of TQT.
- We run the TQTSolver on new benchmarks and the benchmarks provided in [Capelli, Lagniez, Plank, and Seidl, 2024].
- Compare the performance of TQTSolver and d4-QBF [Capelli, Lagniez, Plank, and Seidl, 2024].
- Experimental setup:
  - The experimental evaluation is performed on an Amazon Web Services (AWS) EC2 'r7a.2xlarge' instance running Ubuntu.
  - Run on single core: 900 seconds timeout and a 32 GB memory limit per instance.
  - instance unsolved: either timeout or memory not sufficient.

Experimental Results





Benchmarks \ Solvers	d4-QBF	TQTSolver
Phi_True (99 files)	31	99
Rec_Alt_Taut_False (99 files)	31	99
Random_generated (1800 files)	1800	1800
Unique (666 files)	666	101
PARITYTrue (24 files)	24	14
EQTrueNested (24 files)	24	13
EQTrue (24 files)	23	11
eval2008 (458 files)	93	36

Future Directions

- Theoretical p-simulation results between cert-TQD and d4-QBF?
- To improve our solver by incorporating more engineering and optimization techniques.
- Extends the TQTSolver to TQD.

Thank you.

Bibliography I

[1] Peter Pfeiffer, Daniel Große, and Martina Seidl. "Refined Notions of QBF Equivalences". In: *Logics in Artificial Intelligence - 19th European Conference, JELIA 2025*. Lecture Notes in Computer Science. Springer, 2025, pp. 159–165. DOI: 10.1007/978-3-032-04590-4_11.

[2] Florent Capelli et al. "A Top-Down Tree Model Counter for Quantified Boolean Formulas". In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024*. ijcai.org, 2024, pp. 1853–1861. DOI: 10.24963/ijcai.2024/205.

Algorithm 1. Explorer

Require: A QBF $Q.\phi$ and a partial assignment α.
Ensure: A TQT structure (Q, T).

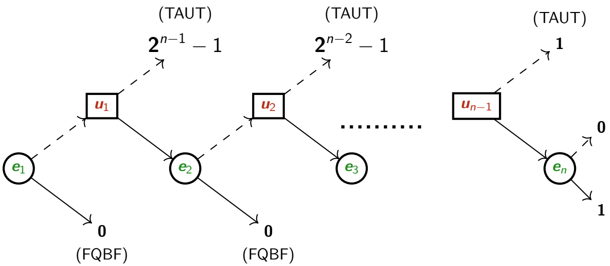
$N \leftarrow \text{Node}(F = Q.\phi, e = 0, v = 0)$.
$Q'.\phi', N.e \leftarrow \mathit{Simplify}(Q.\phi)$.
$k \leftarrow \mathit{ComputeHash}(Q'.\phi')$.
if $\mathit{Cache}(k)$ **then**
  $N.eq \leftarrow \mathit{Cache}(k)$.
  Output N as SEEN.
end if
$\mathit{Cache}(k) \leftarrow N$.
if $\mathit{FQBF}(Q'.\phi')$ **then**
  Output N as FQBF_LEAF.
else if $\mathit{TAUT}(\phi)$ **then**
  $N.v \leftarrow \mathit{TautCountStrategies}(Q'.\phi')$.
  Output N as TAUT_LEAF.
end if
Denote the leftmost variable in $Q'.\phi'$ as a.
Output N as EXPANDED.
$N_1 \leftarrow \mathit{Explorer}(Q'.\phi' |_{a=0}, \alpha \cup \{a = 0\})$.
if $\mathit{Equiv}(Q'.\phi' |_{a=0}, Q'.\phi' |_{a=1})$ **then**
  $N_2 \leftarrow N_1$.
else
  $N_2 \leftarrow \mathit{Explorer}(Q'.\phi' |_{a=1}, \alpha \cup \{a = 1\})$.
end if
Output an edge from N to N_1 and N_2 each.

Algorithm 2. EvaluateTreeHelper

Require: A TQT structure (Q, T) and a node N in T.
Ensure: An integer c denoting the value calculated at N.

if memo(N) **then**
  Return memo(N).
end if
if $N.eq$ is set **then**
  $c \leftarrow \mathit{EvaluateTree}((Q, T), N.eq)$.
  $c \leftarrow 2^{N.e - N.eq.e} \times c$.
  memo$(N) \leftarrow c$.
else if N is an FQBF leaf **then**
  memo$(N) \leftarrow 0$.
else if N is a TAUT leaf **then**
  memo$(N) \leftarrow 2^{N.e} \times N.v$.
else
  $c_1 \leftarrow \mathit{EvaluateTreeHelper}((Q, T), \mathit{left}(N))$.
  $c_2 \leftarrow \mathit{EvaluateTreeHelper}((Q, T), \mathit{right}(N))$.
  if N is universal node **then**
    $c \leftarrow c_1 \times c_2$.
  else
    $c \leftarrow c_1 + c_2$.
  end if
  memo$(N) \leftarrow 2^{N.e} \times c$.
end if
Return memo(N).

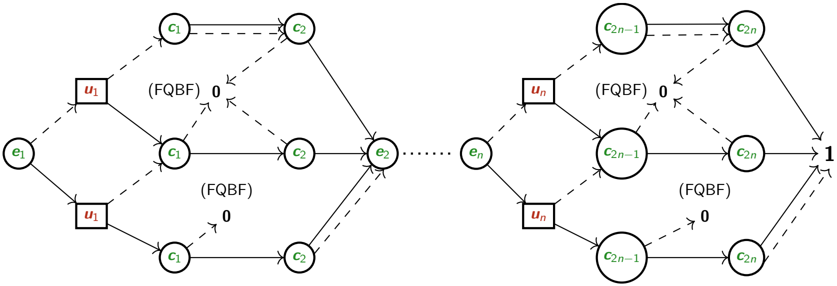
Model count of 'Rec_Alt_Taut_False' instances ($n \geq 2$):



Taut_count $\leftarrow 1$
models $\leftarrow 1$
for $i \leftarrow 1$ to $n-1$ **do**
  Taut_count $\leftarrow 2 \times (\text{Taut\_count})^2$
  models $\leftarrow$ models $\times$ Taut_count
end for
return models

Model count $= 2^{2^n - n - 1}$

Model count of ‘Phi_True’ instances ($n \geq 2$):



Prev_count $\leftarrow$ 1
models $\leftarrow$ 1
for $i \leftarrow 1$ to n **do**
  Prev_count $\leftarrow$ models
  models $\leftarrow 4\times$ Prev_count^2
end for
return models

Model count $= 2^{2^{n+1}-2}$