# QRATPre+: Effective QBF Preprocessing via Strong Redundancy Properties

Florian Lonsing[1]     Uwe Egly[2]

[1]Department of Computer Science, Stanford University, Stanford, CA 94305, USA
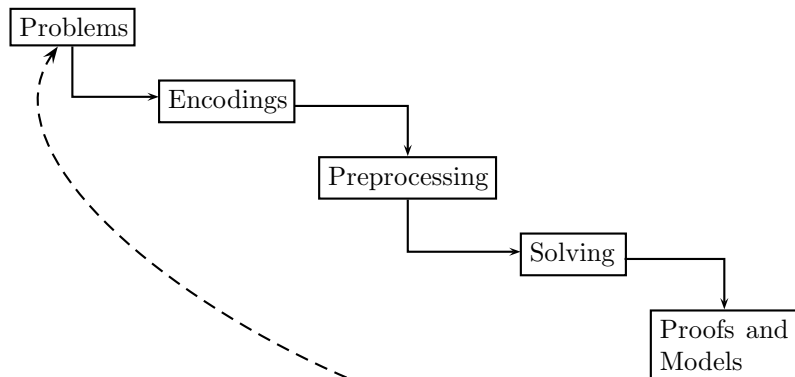
[2]Institute of Logic and Computation, TU Wien, Vienna, Austria

*22nd International Conference on Theory and Applications of Satisfiability Testing, July 7 - 12, 2019, Lisbon, Portugal*

# QBF Workflow

- Quantified Boolean formulas in prenex CNF: $\psi := \Pi.\phi$
- Prefix $\Pi := Q_1 B_1 \ldots Q_n B_n$, $Q_i \in \{\forall, \exists\}$, variable sets $B_i$.



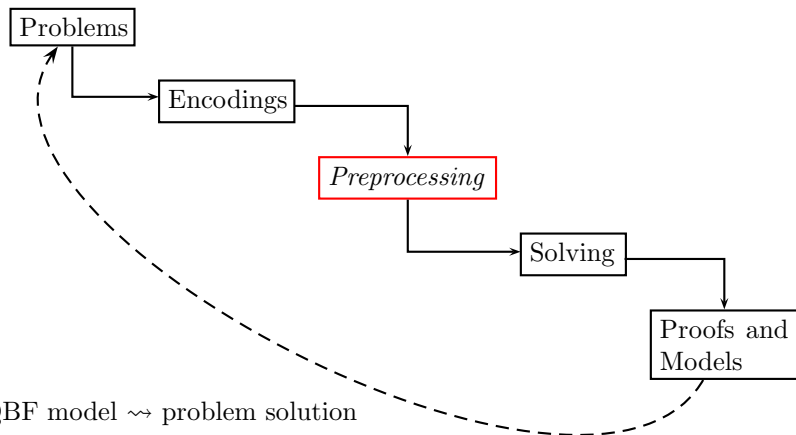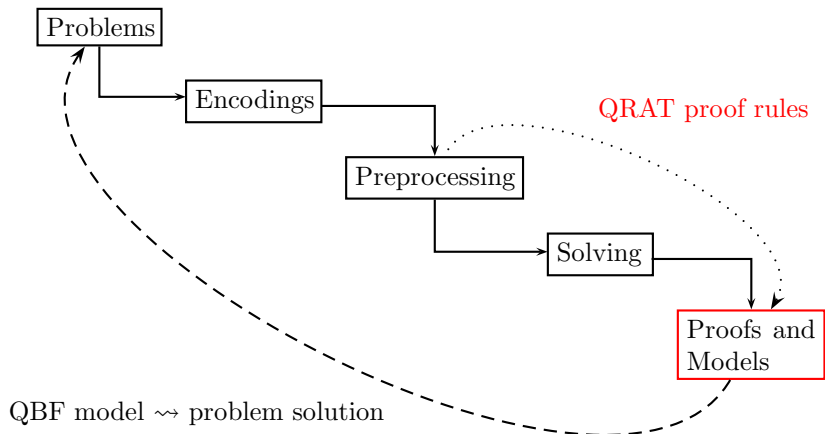QBF model $\rightsquigarrow$ problem solution

# QBF Workflow

- Quantified Boolean formulas in prenex CNF: $\psi := \Pi.\phi$
- Prefix $\Pi := Q_1 B_1 \ldots Q_n B_n$, $Q_i \in \{\forall, \exists\}$, variable sets $B_i$.
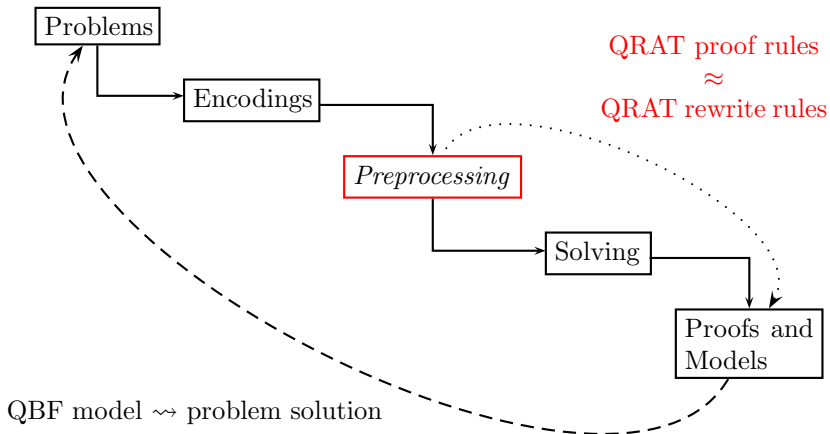


QBF model $\rightsquigarrow$ problem solution

# QRAT: Quantified RAT

- Proof system, generalization of (D)RAT from SAT to QBF .
- Proof/model generation in workflows with preprocessing .



QBF model $\rightsquigarrow$ problem solution

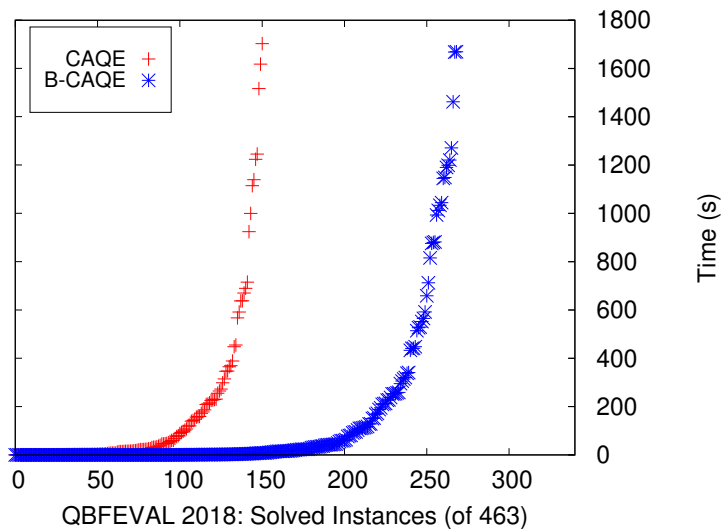# QRAT: Quantified RAT

- QRAT proof rules: based on QBF redundancy property.
- Idea: apply QRAT redundancy property to simplify QBFs.
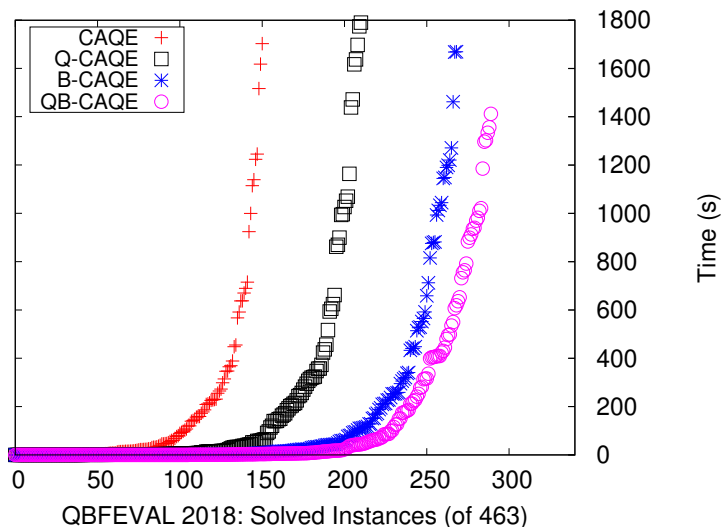


QBF model $\rightsquigarrow$ problem solution
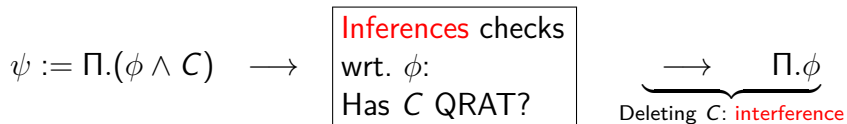
# QRATPre+: Improving the Power of Preprocessing



- 2018 winner CAQE with Bloqqer (B): +78% solved.

# QRATPre+: Improving the Power of Preprocessing



- Applying QRATPre+ (Q) and Bloqqer (B): +92% solved.

# QRAT$^+$: Generalization of QRAT

$$\psi := \Pi.(\phi \wedge C) \quad \longrightarrow \quad \boxed{\begin{array}{l} \text{Inferences checks} \\ \text{wrt. } \phi\text{:} \\ \text{Has } C \text{ QRAT?} \end{array}} \qquad \underbrace{\longrightarrow \quad \Pi.\phi}_{\text{Deleting } C\text{: interference}}$$

**Simplifications in Preprocessing:** cf. [HK17]

- Equivalence-preserving inference steps: $\Pi.(\phi \wedge D) \equiv \Pi.\phi$.
- Satisfiability-preserving interference steps: $\Pi.(\phi \wedge D) \equiv_{sat} \Pi.\phi$.

# QRAT$^+$: Generalization of QRAT
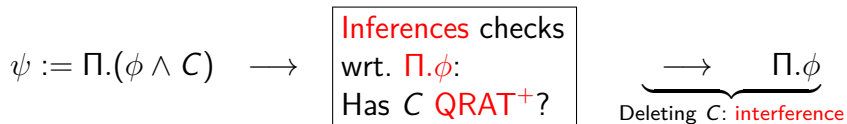
$$\psi := \Pi.(\phi \wedge C) \quad \longrightarrow \quad \boxed{\begin{array}{l} \text{Inferences checks} \\ \text{wrt. } \Pi.\phi\text{:} \\ \text{Has } C \text{ QRAT}^+? \end{array}} \qquad \underbrace{\longrightarrow}_{\text{Deleting } C: \text{ interference}} \Pi.\phi$$

**Simplifications in Preprocessing:** cf. [HK17]

- Equivalence-preserving inference steps: $\quad \Pi.(\phi \wedge D) \;\equiv\; \Pi.\phi.$
- Satisfiability-preserving interference steps: $\Pi.(\phi \wedge D) \;\equiv_{sat}\; \Pi.\phi.$

QRAT$^+$-**based QBF Preprocessing:** cf. IJCAR 2018 [LE18]

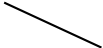- Generalizes inference checking, leverages structure of prefix $\Pi$.
- Poly-time inferences checks by QBF unit propagation.
- More powerful interferences by stronger redundancy property.

# Interference Checking in QRAT$^+$

$D_1 = D_1' \cup \{\bar{l}\} \ldots$ $\qquad\qquad$ $D_j = D_j' \cup \{\bar{l}\} \ldots$ $\qquad\qquad$ $D_n = D_n' \cup \{\bar{l}\}$

$$\mathsf{OR}_1 := \mathsf{OR}(C, D_1, l) \subset (C \cup D_1)$$
$$i = max(levels(\Pi, \mathsf{OR}_1)) = 3$$
$$\Pi = \exists B_1 \forall B_2 \exists B_3 \forall B_4 \exists B_5$$
$$\exists\text{-}Abs(\Pi.\phi, i) := \exists B_1 \exists B_2 \exists B_3 \forall B_4 \exists B_5.\phi$$
$$\text{Check:} \quad Abs(\Pi.(\phi \wedge \overline{\mathsf{OR}_1}), i) \vDash_{\overline{1\forall}} \emptyset?$$

$$C = C' \cup \{l\}$$

## Checking the Resolution Neighborhood (RN):

- RN of $C = C' \cup \{l\}$: all possible resolution partners.
- Outer resolvent ($\mathsf{OR}_i$): subsets of actual resolvent of $C$ and $D_i$.
- QBF unit propagation ($\vDash_{\overline{1\forall}}$) on $\exists$-abstraction: $\Pi.\phi \stackrel{?}{\equiv} \Pi.(\phi \wedge \mathsf{OR}_i)$.

# Interference Checking in QRAT$^+$

$$D_1 = D'_1 \cup \{\bar{l}\} \dots \qquad D_j = D'_j \cup \{\bar{l}\} \dots \qquad D_n = D'_n \cup \{\bar{l}\}$$

$$\mathsf{OR}_j := \mathsf{OR}(C, D_j, l) \subset (C \cup D_j)$$
$$i = max(levels(\Pi, \mathsf{OR}_j)) = 1$$
$$\Pi = \exists B_1 \forall B_2 \exists B_3 \forall B_4 \exists B_5$$
$$\exists\text{-}Abs(\Pi.\phi, i) := \exists B_1 \forall B_2 \exists B_3 \forall B_4 \exists B_5.\phi$$
$$\text{Check:} \quad Abs(\Pi.(\phi \wedge \overline{\mathsf{OR}_j}), i) \models_{\overline{1}\forall} \emptyset?$$

$$C = C' \cup \{l\}$$

## Checking the Resolution Neighborhood (RN):

- RN of $C = C' \cup \{l\}$: all possible resolution partners.
- Outer resolvent ($\mathsf{OR}_i$): subsets of actual resolvent of $C$ and $D_i$.
- QBF unit propagation ($\models_{\overline{1}\forall}$) on $\exists$-abstraction: $\Pi.\phi \overset{?}{\equiv} \Pi.(\phi \wedge \mathsf{OR}_i)$.

# Interference Checking in QRAT$^+$

$$D_1 = D'_1 \cup \{\bar{l}\} \ldots \qquad\qquad D_j = D'_j \cup \{\bar{l}\} \ldots \qquad\qquad D_n = D'_n \cup \{\bar{l}\}$$

$$\mathsf{OR}_n := \mathsf{OR}(C, D_n, l) \subset (C \cup D_n)$$
$$i = max(levels(\Pi, \mathsf{OR}_n)) = 5$$
$$\Pi = \exists B_1 \forall B_2 \exists B_3 \forall B_4 \exists B_5$$
$$\exists\text{-}Abs(\Pi.\phi, i) := \exists B_1 \exists B_2 \exists B_3 \exists B_4 \exists B_5.\phi$$
$$\text{Check:} \quad Abs(\Pi.(\phi \wedge \overline{\mathsf{OR}_n}), i) \vDash_{\overline{1}\forall} \emptyset?$$

$$C = C' \cup \{l\}$$

## Checking the Resolution Neighborhood (RN):

- RN of $C = C' \cup \{l\}$: all possible resolution partners.
- Outer resolvent ($\mathsf{OR}_i$): subsets of actual resolvent of $C$ and $D_i$.
- QBF unit propagation ($\vDash_{\overline{1}\forall}$) on $\exists$-abstraction: $\Pi.\phi \overset{?}{\equiv} \Pi.(\phi \wedge \mathsf{OR}_i)$.

# Interference Checking in QRAT$^+$

$$D_1 = D_1' \cup \{\bar{l}\} \ldots \qquad D_j = D_j' \cup \{\bar{l}\} \ldots \qquad D_n = D_n' \cup \{\bar{l}\}$$

$$\mathsf{OR}_n := \mathsf{OR}(C, D_n, l) \subset (C \cup D_n)$$
$$i = max(levels(\Pi, \mathsf{OR}_n)) = 5$$
$$\Pi = \exists B_1 \forall B_2 \exists B_3 \forall B_4 \exists B_5$$
$$\exists\text{-}Abs(\Pi.\phi, i) := \exists B_1 \exists B_2 \exists B_3 \exists B_4 \exists B_5.\phi$$
$$\text{Check:} \quad Abs(\Pi.(\phi \wedge \overline{\mathsf{OR}_n}), i) \vDash_{\mathbb{1}\forall} \emptyset?$$

$$C = C' \cup \{l\}$$

$$\psi := \Pi.(\phi \wedge C) \longrightarrow \boxed{\begin{array}{l} \textit{Inference check for all } \mathsf{OR}_i: \\ \Pi.\phi \equiv \Pi.(\phi \wedge \mathsf{OR}_i) \text{ if} \\ Abs(\Pi.(\phi \wedge \overline{\mathsf{OR}_i}), i) \vDash_{\mathbb{1}\forall} \emptyset \end{array}} \longrightarrow \underbrace{\Pi.\phi}_{C \text{ has QRAT}^+}$$

# Soundness of Preprocessing by QRAT and QRAT$^+$

> ## Theorem ([LE18], cf. [HSB17])
>
> *Given a PCNF $\psi := \Pi.(\phi \wedge C)$ with a clause $C = C' \cup \{l\}$.*
>
> *If clause $C$ has QRAT$^+$ on literal $l$ with respect to $\Pi.\phi$ and*
>
> 1. *$q(l) = \exists$, then $\psi \equiv_{sat} \Pi.\phi$.* $\qquad\qquad$ *(add/delete clauses)*
> 2. *$q(l) = \forall$, then $\psi \equiv_{sat} \Pi.(\phi \wedge C')$.* $\qquad$ *(add/delete literals)*

- Implementation in QRATPre+: only deletion of clauses/literals.

# High-Level Workflow of QRATPre+

**Input:** PCNF $\psi$.
**Output:** Simplified PCNF $\psi'$.

1   $\psi' = \psi$;
2   **do**
     // clause elimination
3      $\psi' := \text{QBCE}(\psi')$;
4      $\psi' := \text{QAT}(\psi')$;
5      $\psi' := \text{QRATE}^+(\psi')$;
     // literal elimination
6      $\psi' := \text{BLE}(\psi')$;
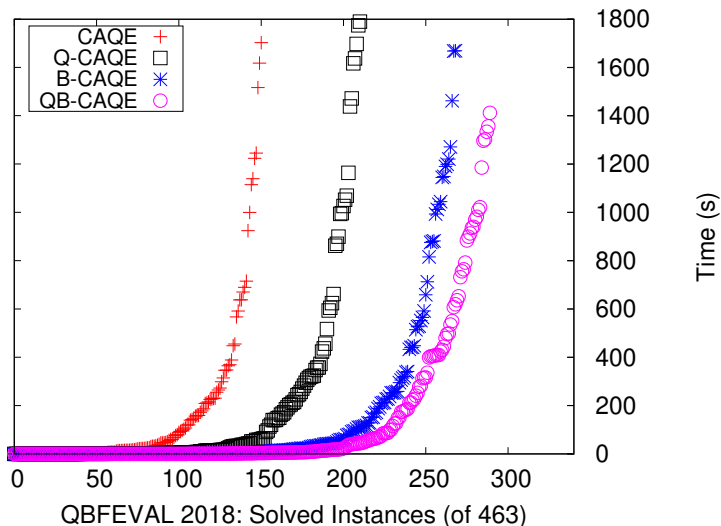7      $\psi' := \text{QRATU}^+(\psi')$;
8   **while** $\psi'$ *changed*;

- Checking $\text{QRAT}^+$ property: poly-time, but potentially expensive.
- Reduce formula by applying "cheap" (weaker) techniques first.
- Watched data structures to avoid re-checking clauses.
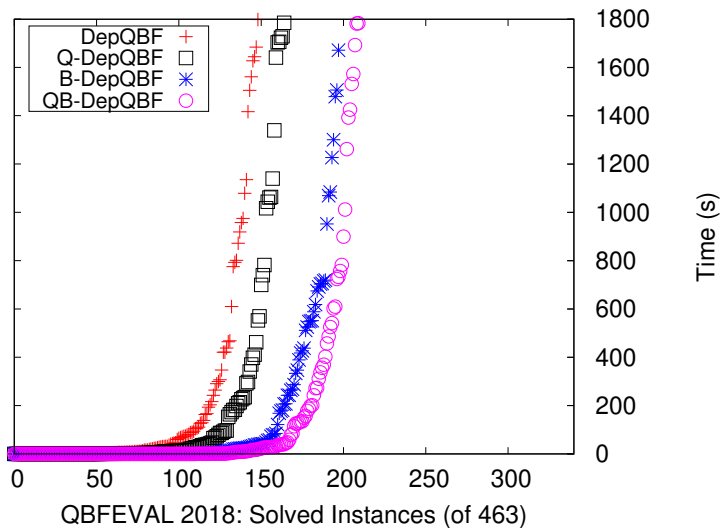
# Experiments: CNF Size Reduction
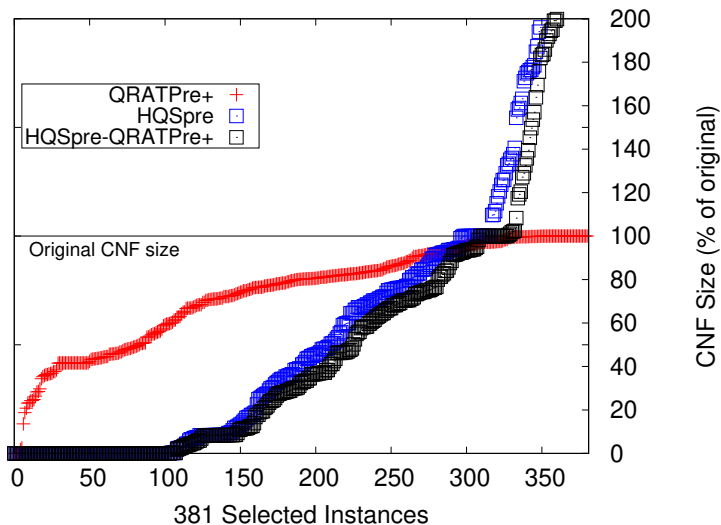


- HQSpre might increase CNF size, QRATPre+ still effective.
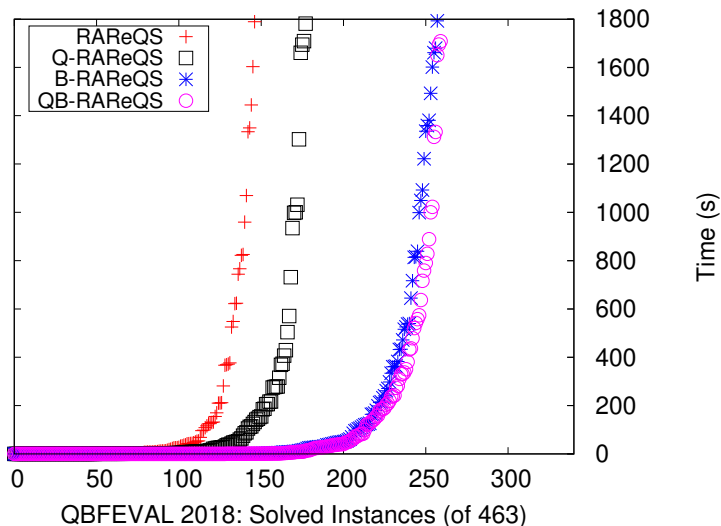
# Experiments: Different Solving Paradigms



Only Bloqqer (B)/also QRATPre+ (QB): +78% / +92% solved.

# Experiments: Different Solving Paradigms



■ Only Bloqqer (B)/also QRATPre+ (QB): +32% / +40% solved.

# Experiments: Different Solving Paradigms



Only Bloqqer (B)/also QRATPre+ (QB): +75% / +76% solved.

# Summary

**QRATPre+: Effective QBF Preprocessing**

- Strong redundancy property based on $QRAT^+$.
- Orthogonal to existing preprocessors.
- Improvement of state-of-the-art preprocessing and solving.
- Implementation in C, API for integration in QBF workflows.
- Future work: handling non-confluence, $QRAT^+$ proof checking and Skolem function extraction, tentatively adding redundancy.

QRATPre+ source code (GPLv3):
`https://lonsing.github.io/qratpreplus/`

# *References*

# References I

[FHSB17]  Katalin Fazekas, Marijn Heule, Martina Seidl, and Armin
          Biere.
          Skolem Function Continuation for Quantified Boolean
          Formulas.
          In *TAP*, volume 10375 of *LNCS*, pages 129–138. Springer,
          2017.

[HK17]    Marijn Heule and Benjamin Kiesl.
          The Potential of Interference-Based Proof Systems.
          In *ARCADE*, volume 51 of *EPiC Series in Computing*,
          pages 51–54. EasyChair, 2017.

# References II

[HSB14]   Marijn Heule, Martina Seidl, and Armin Biere.
          Efficient extraction of Skolem functions from QRAT
          proofs.
          In *FMCAD*, pages 107–114. IEEE, 2014.

[HSB17]   Marijn Heule, Martina Seidl, and Armin Biere.
          Solution Validation and Extraction for QBF Preprocessing.

          *J. Autom. Reasoning*, 58(1):97–125, 2017.

[LE18]    Florian Lonsing and Uwe Egly.
          QRAT$^{+}$: Generalizing QRAT by a More Powerful QBF
          Redundancy Property.
          In *IJCAR*, volume 10900 of *Lecture Notes in Computer
          Science*, pages 161–177. Springer, 2018.