A Resolution Proof System for Dependency Stochastic Boolean Satisfiability

Yun-Rong Luo [1] Advisor: Prof. Jie-Hong Roland Jiang [1,2]

Thesis Committee Members: Prof. Tony Tan [3] and Prof. Fang Yu [4]

[1]Graduate Institute of Electronics Engineering, National Taiwan University

[2]Department of Electrical Engineering, National Taiwan University

[3]Department of Computer Science and Information Engineering, National Taiwan University

[4]Department of Management Information Systems, National Chengchi University

Master's Thesis Oral Defense, July 18th, 2023

Outline

Outline

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].
- The success of SAT solvers has motivated research in more expressive logical formalisms that extend propositional logic with quantifiers.

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].

- The success of SAT solvers has motivated research in more expressive logical formalisms that extend propositional logic with quantifiers.

$\boxed{\text{QBF}}$

Figure 1: Relationships between (D)QBF and (D)SSAT

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].
- The success of SAT solvers has motivated research in more expressive logical formalisms that extend propositional logic with quantifiers.

$$\boxed{\text{QBF}} \xrightarrow{\text{nonlinear dependency}} \boxed{\text{DQBF}}$$

Figure 1: Relationships between (D)QBF and (D)SSAT

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].

- The success of SAT solvers has motivated research in more expressive logical formalisms that extend propositional logic with quantifiers.

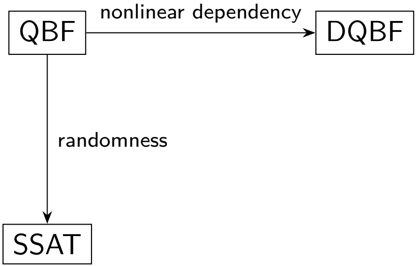

Figure 1: Relationships between (D)QBF and (D)SSAT

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].

- The success of SAT solvers has motivated research in more expressive logical formalisms that extend propositional logic with quantifiers.

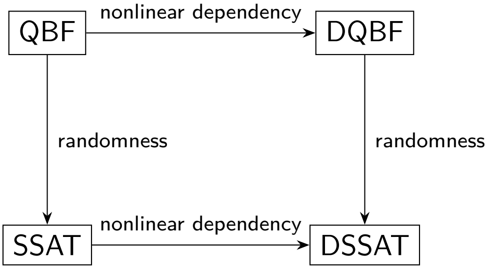

Figure 1: Relationships between (D)QBF and (D)SSAT

From SAT to Beyond-NP Formalisms

- Boolean satisfiability (SAT) solvers have been successfully applied in various areas such as artificial intelligence [Nil82], model checking [VWM15], electronic design automation [MS00], and combinatorics [HKM16].

- The success of SAT solvers has motivated research in more expressive logical formalisms that extend propositional logic with quantifiers.

- DSSAT [LJ21] is a new logical formalism that generalizes DQBF [SW18] and SSAT [Pap85]. With little related work [LJ21; CJ23], DSSAT is still in an early research stage.

$$
\begin{tikzcd}
\boxed{\text{QBF}} \arrow[r, "\text{nonlinear dependency}"] \arrow[d, "\text{randomness}"] & \boxed{\text{DQBF}} \arrow[d, "\text{randomness}"] \\
\boxed{\text{SSAT}} \arrow[r, "\text{nonlinear dependency}"] & \boxed{\text{DSSAT}}
\end{tikzcd}
$$

Figure 1: Relationships between (D)QBF and (D)SSAT

SSAT and DSSAT Applications

- SSAT (resp. DSSAT) can compactly encode PSPACE (resp. NEXPTIME) decision problems under uncertainty.

SSAT and DSSAT Applications

- SSAT (resp. DSSAT) can compactly encode PSPACE (resp. NEXPTIME) decision problems under uncertainty.
- Stochastic Boolean satisfiability (SSAT) [Pap85]
  - Game semantics: game against nature.
  - Applications: probabilistic planning [ML98; ML03; MB05; SP20], probabilistic equivalence checking [LJ18], probabilistic graphical models [HJ22], and verifying machine learning fairness [GBM21].

SSAT and DSSAT Applications

- SSAT (resp. DSSAT) can compactly encode PSPACE (resp. NEXPTIME) decision problems under uncertainty.
- Stochastic Boolean satisfiability (SSAT) [Pap85]
  - Game semantics: game against nature.
  - Applications: probabilistic planning [ML98; ML03; MB05; SP20], probabilistic equivalence checking [LJ18], probabilistic graphical models [HJ22], and verifying machine learning fairness [GBM21].
- Dependency stochastic Boolean satisfiability (DSSAT) [LJ21]
  - Game semantics: multi-player game under uncertainty and partial information.
  - Applications: decentralized partially observable Markov decision process (Dec-POMDP) and equivalence checking of probabilistic partial design [LJ21].

Example 1 (Equivalence Checking of Probabilistic Partial Design)

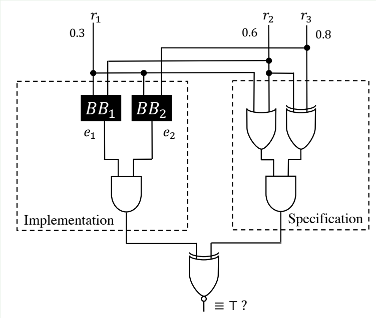

Figure 2: Example of probabilistic partial design.

Motivating Example of DSSAT Application

Example 1 (Equivalence Checking of Probabilistic Partial Design)

1. Probablistic inputs: r_1, r_2, and r_3 valuate to True with a probability of $0.3, 0.6$, and 0.8, respectively.

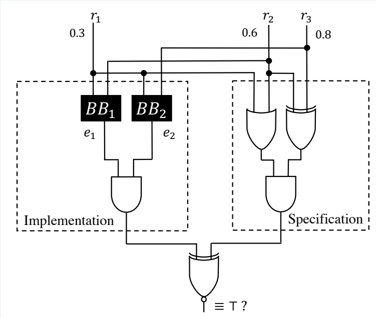

Figure 2: Example of probabilistic partial design.

Example 1 (Equivalence Checking of Probabilistic Partial Design)

1. Probablistic inputs: r_1, r_2, and r_3 valuate to True with a probability of $0.3, 0.6$, and 0.8, respectively.
2. Specification: $F_S \equiv (r_1 \vee r_2)(r_2 \oplus r_3)$.

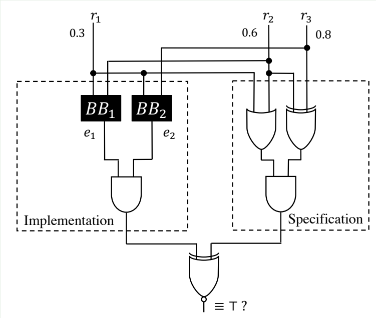

Figure 2: Example of probabilistic partial design.

Motivating Example of DSSAT Application

Example 1 (Equivalence Checking of Probabilistic Partial Design)

1. Probablistic inputs: r_1, r_2, and r_3 valuate to True with a probability of $0.3, 0.6$, and 0.8, respectively.
2. Specification: $F_S \equiv (r_1 \vee r_2)(r_2 \oplus r_3)$.
3. Implementation: a partial design $F_I \equiv \mathrm{BB}_1 \wedge \mathrm{BB}_2$, where BB_1 and BB_2 are black boxes with inputs $\{r_1, r_2\}$ and $\{r_1, r_3\}$, respectively.

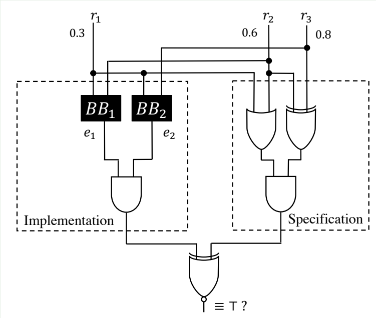

Figure 2: Example of probabilistic partial design.

Motivating Example of DSSAT Application

Example 1 (Equivalence Checking of Probabilistic Partial Design)

1. Probablistic inputs: r_1, r_2, and r_3 valuate to True with a probability of $0.3, 0.6$, and 0.8, respectively.
2. Specification: $F_S \equiv (r_1 \vee r_2)(r_2 \oplus r_3)$.
3. Implementation: a partial design $F_I \equiv \mathrm{BB}_1 \wedge \mathrm{BB}_2$, where BB_1 and BB_2 are black boxes with inputs $\{r_1, r_2\}$ and $\{r_1, r_3\}$, respectively.
4. Equivalence checking: aims to synthesize the black boxes $\mathrm{BB}_1 = (r_1 \oplus r_2)$ and $\mathrm{BB}_2 = (r_1 \equiv r_3)$ to maximize the equivalent probability $\Pr[F_S \equiv F_I] = 0.964$ between implementation and specification.

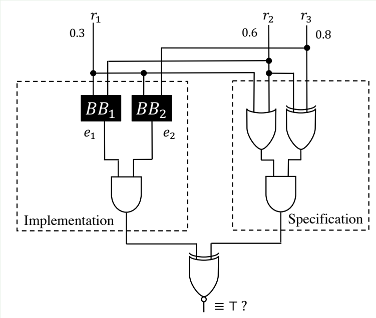

Figure 2: Example of probabilistic partial design.

Motivating Example of DSSAT Application

Problem Statement (Equivalence Checking of Probabilistic Partial Design)

Given a probabilistic partial design, the equivalence checking of probabilistic partial design (PPEC) finds realizations of black boxes making the implementation and specification behave the same to a maximum degree.

Motivating Example of DSSAT Application

Problem Statement (Equivalence Checking of Probabilistic Partial Design)

Given a probabilistic partial design, the equivalence checking of probabilistic partial design (PPEC) finds realizations of black boxes making the implementation and specification behave the same to a maximum degree.

- PPEC is NEXPTIME-complete and can be compactly encoded by DSSAT [LJ21]:

Motivating Example of DSSAT Application

Problem Statement (Equivalence Checking of Probabilistic Partial Design)

Given a probabilistic partial design, the equivalence checking of probabilistic partial design (PPEC) finds realizations of black boxes making the implementation and specification behave the same to a maximum degree.

- PPEC is NEXPTIME-complete and can be compactly encoded by DSSAT [LJ21]:
- DSSAT describes multi-agent systems with uncertainty and partial information, which can naturally encode the black boxes in PPEC with probabilistic inputs and incomparable supports.

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:
  1. generates polynomial-time verifiable proofs;

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:
  1. generates polynomial-time verifiable proofs;
  2. allows local pruning that reduces proof sizes;

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:
  1. generates polynomial-time verifiable proofs;
  2. allows local pruning that reduces proof sizes;
  3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:
  1. generates polynomial-time verifiable proofs;
  2. allows local pruning that reduces proof sizes;
  3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.

- Potential benefits of `DS-Res`:

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:
  1. generates polynomial-time verifiable proofs;
  2. allows local pruning that reduces proof sizes;
  3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.

- Potential benefits of `DS-Res`:
  1. provides theoretical insights for future DSSAT solver development; [1]

[1] The only existing DSSAT solver [CJ23] does not solve DSSAT instances directly, but instead converts a DSSAT instance into an SSAT one of equal solution and uses off-the-shelf SSAT solvers to compute the solution.

Contributions

Our contributions: developed the first sound and complete resolution proof system `DS-Res` for DSSAT and studied the properties of `DS-Res`.

- Favorable properties of `DS-Res`:
  1. generates polynomial-time verifiable proofs;
  2. allows local pruning that reduces proof sizes;
  3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.

- Potential benefits of `DS-Res`:
  1. provides theoretical insights for future DSSAT solver development; [1]
  2. serves as the underlying proof system for proof logging in future trustworthy DSSAT solvers. [2]

---

[1] The only existing DSSAT solver [CJ23] does not solve DSSAT instances directly, but instead converts a DSSAT instance into an SSAT one of equal solution and uses off-the-shelf SSAT solvers to compute the solution.

[2] SAT solvers [Heu21] and model counters [FHR22] now support proof logging to provide external certification of the solvers' results.

Outline

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \land v_2) \implies e_1)(e_2 \equiv (v_1 \land v_2))$ and prefix $\mathcal{Q}$ be:

$\forall v_1, \exists e_1, \forall v_2, \exists e_2$

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$$\forall v_1, \exists e_1, \forall v_2, \exists e_2$$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$\forall v_1, \exists e_1, \forall v_2, \exists e_2$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$\reflectbox{R}^{0.7} v_1, \exists e_1, \reflectbox{R}^{0.6} v_2, \exists e_2$

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$$\forall v_1, \exists e_1, \forall v_2, \exists e_2$$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$$\reflectbox{R}^{0.7} v_1, \exists e_1, \reflectbox{R}^{0.6} v_2, \exists e_2$$

QBF/SSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_1, v_2\} \end{cases}$

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \land v_2) \implies e_1)(e_2 \equiv (v_1 \land v_2))$ and prefix $\mathcal{Q}$ be:

$\forall v_1, \exists e_1, \forall v_2, \exists e_2$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$\reflectbox{R}^{0.7} v_1, \exists e_1, \reflectbox{R}^{0.6} v_2, \exists e_2$

$$\begin{tikzcd} \text{QBF} \arrow[r] \arrow[d] & \text{DQBF} \arrow[d] \\ \text{SSAT} \arrow[r] & \text{DSSAT} \end{tikzcd}$$

QBF/SSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_1, v_2\} \end{cases}$ $\xrightarrow{\text{nonlinear dependency}}$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$$\forall v_1, \exists e_1, \forall v_2, \exists e_2 \qquad \forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})$$

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$$\reflectbox{R}^{0.7} v_1, \exists e_1, \reflectbox{R}^{0.6} v_2, \exists e_2 \qquad \reflectbox{R}^{0.7} v_1, \reflectbox{R}^{0.6} v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})$$

QBF/SSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_1, v_2\} \end{cases}$ $\xrightarrow{\text{nonlinear dependency}}$ DQBF/DSSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_2\} \end{cases}$

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$$\begin{tikzcd}
\forall v_1, \exists e_1, \forall v_2, \exists e_2 & \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] & \forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\}) \\
\rotatebox[origin=c]{180}{R}^{0.7} v_1, \exists e_1, \rotatebox[origin=c]{180}{R}^{0.6} v_2, \exists e_2 & \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} & \rotatebox[origin=c]{180}{R}^{0.7} v_1, \rotatebox[origin=c]{180}{R}^{0.6} v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})
\end{tikzcd}$$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

QBF/SSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_1, v_2\} \end{cases} \xrightarrow{\text{nonlinear dependency}}$ DQBF/DSSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_2\} \end{cases}$

- Semantics: search for existential strategy $\mathcal{F} = (f_{e_1}, f_{e_2})$, where f_{e_1} (resp. f_{e_2}) is Skolem function for e_1 (resp. e_2) that depends on D_{e_1} (resp. D_{e_2}).

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$\forall v_1, \exists e_1, \forall v_2, \exists e_2$ (QBF)

$\forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})$ (DQBF)

$$\begin{tikzcd} \boxed{\text{QBF}} \arrow[r] \arrow[d] & \boxed{\text{DQBF}} \arrow[d] \\ \boxed{\text{SSAT}} \arrow[r] & \boxed{\text{DSSAT}} \end{tikzcd}$$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$\reflectbox{\textsf{R}}^{0.7}\, v_1, \exists e_1, \reflectbox{\textsf{R}}^{0.6}\, v_2, \exists e_2$ (SSAT)

$\reflectbox{\textsf{R}}^{0.7}\, v_1, \reflectbox{\textsf{R}}^{0.6}\, v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})$ (DSSAT)

QBF/SSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_1, v_2\} \end{cases}$ $\xrightarrow{\text{nonlinear dependency}}$ DQBF/DSSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_2\} \end{cases}$

- Semantics: search for existential strategy $\mathcal{F} = (f_{e_1}, f_{e_2})$, where f_{e_1} (resp. f_{e_2}) is Skolem function for e_1 (resp. e_2) that depends on D_{e_1} (resp. D_{e_2}).
  - QBF/DQBF: decide whether there exists $\mathcal{F}$ such that $\Phi|_{\mathcal{F}} \equiv \top$ is tautology.

Beyond-NP Formalisms: (D)QBF and (D)SSAT

- Syntax: $\Phi = \mathcal{Q}.\ \phi$, let matrix $\phi = ((v_1 \wedge v_2) \implies e_1)(e_2 \equiv (v_1 \wedge v_2))$ and prefix $\mathcal{Q}$ be:

$\forall v_1, \exists e_1, \forall v_2, \exists e_2$ (QBF)

$\forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})$ (DQBF)

$$\begin{tikzcd} \text{QBF} \arrow[r] \arrow[d] & \text{DQBF} \arrow[d] \\ \text{SSAT} \arrow[r] & \text{DSSAT} \end{tikzcd}$$

randomness $\begin{cases} \Pr[v_1 \equiv \top] = 0.7, \\ \Pr[v_2 \equiv \top] = 0.6 \end{cases}$

$\reflectbox{R}^{0.7}\, v_1, \exists e_1, \reflectbox{R}^{0.6}\, v_2, \exists e_2$ (SSAT)

$\reflectbox{R}^{0.7}\, v_1, \reflectbox{R}^{0.6}\, v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})$ (DSSAT)

QBF/SSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_1, v_2\} \end{cases}$ $\xrightarrow{\text{nonlinear dependency}}$ DQBF/DSSAT $\begin{cases} D_{e_1} = \{v_1\}, \\ D_{e_2} = \{v_2\} \end{cases}$

- Semantics: search for existential strategy $\mathcal{F} = (f_{e_1}, f_{e_2})$, where f_{e_1} (resp. f_{e_2}) is Skolem function for e_1 (resp. e_2) that depends on D_{e_1} (resp. D_{e_2}).
  - QBF/DQBF: decide whether there exists $\mathcal{F}$ such that $\Phi|_{\mathcal{F}} \equiv \top$ is tautology.
  - SSAT/DSSAT: maximize satisfying probability $\Pr[\Phi|_{\mathcal{F}}]$.

Quantified Boolean Formula (QBF)

Example 2 (QBF)

The QBF formula

$$\Phi = \underbrace{\forall v_1, \exists e_1, \forall v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}} . \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is satisfiable, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes a tautology

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee (\bar{v}_1 \vee \bar{v}_2))(v_2 \vee (\bar{v}_1 \vee \bar{v}_2))(\bar{v}_1 \vee \bar{v}_2 \vee (v_1 \wedge v_2)) \equiv \top.$$

Quantified Boolean Formula (QBF)

Example 2 (QBF)

The QBF formula

$$\Phi = \underbrace{\forall v_1, \exists e_1, \forall v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}} . \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is satisfiable, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes a tautology

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee (\bar{v}_1 \vee \bar{v}_2))(v_2 \vee (\bar{v}_1 \vee \bar{v}_2))(\bar{v}_1 \vee \bar{v}_2 \vee (v_1 \wedge v_2)) \equiv \top.$$

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n . \phi$, each $Q_i \in \{\forall, \exists\}$

Quantified Boolean Formula (QBF)

Example 2 (QBF)

The QBF formula

$$\Phi = \underbrace{\forall v_1, \exists e_1, \forall v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}} . \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is satisfiable, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes a tautology

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee (\bar{v}_1 \vee \bar{v}_2))(v_2 \vee (\bar{v}_1 \vee \bar{v}_2))(\bar{v}_1 \vee \bar{v}_2 \vee (v_1 \wedge v_2)) \equiv \top.$$

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n . \phi$, each $Q_i \in \{\forall, \exists\}$
- Existential variable v_i can depend on the universal variables in $\{v_1, \ldots, v_{i-1}\}$[3]

[3] Hence, the dependency sets are linearly ordered, i.e. either $D_i \subseteq D_j$ or $D_i \supseteq D_j$ for any i, j.

Quantified Boolean Formula (QBF)

Example 2 (QBF)

The QBF formula

$$\Phi = \underbrace{\forall v_1, \exists e_1, \forall v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}} . \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is satisfiable, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes a tautology

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee (\bar{v}_1 \vee \bar{v}_2))(v_2 \vee (\bar{v}_1 \vee \bar{v}_2))(\bar{v}_1 \vee \bar{v}_2 \vee (v_1 \wedge v_2)) \equiv \top.$$

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\forall, \exists\}$
- Existential variable v_i can depend on the universal variables in $\{v_1, \ldots, v_{i-1}\}$[3]
- Semantics: Φ is *satisfiable* if there is a strategy $\mathcal{F}$ such that $\Phi|_{\mathcal{F}} \equiv \top$

[3] Hence, the dependency sets are linearly ordered, i.e. either $D_i \subseteq D_j$ or $D_i \supseteq D_j$ for any i, j.

Stochastic Boolean satisfiability (SSAT)

Example 3 (SSAT)

The SSAT formula

$$\Phi = \underbrace{\text{Я}^{0.7} v_1, \exists e_1, \text{Я}^{0.6} v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 1, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes tautological and has a weighted model count of 1.

Stochastic Boolean satisfiability (SSAT)

Example 3 (SSAT)

The SSAT formula

$$\Phi = \underbrace{\reflectbox{R}^{0.7} v_1, \exists e_1, \reflectbox{R}^{0.6} v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 1, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes tautological and has a weighted model count of 1.

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\reflectbox{R}^{p_i}, \exists\}$

Stochastic Boolean satisfiability (SSAT)

Example 3 (SSAT)

The SSAT formula

$$\Phi = \underbrace{\reflectbox{R}^{0.7} v_1, \exists e_1, \reflectbox{R}^{0.6} v_2, \exists e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}}. \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 1, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes tautological and has a weighted model count of 1.

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\reflectbox{R}^{p_i}, \exists\}$
- $\reflectbox{R}^{p_i} v_i$ denotes that v_i will be assigned to $\top$ with probability p_i

Stochastic Boolean satisfiability (SSAT)

Example 3 (SSAT)

The SSAT formula

$$\Phi = \underbrace{\text{Я}^{0.7}\, v_1, \exists\, e_1, \text{Я}^{0.6}\, v_2, \exists\, e_2}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 1, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes tautological and has a weighted model count of 1.

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\text{Я}^{p_i}, \exists\}$
- $\text{Я}^{p_i}\, v_i$ denotes that v_i will be assigned to $\top$ with probability p_i
- Existential variable v_i can depend on the randomized variables in $\{v_1, \ldots, v_{i-1}\}$

Stochastic Boolean satisfiability (SSAT)

Example 3 (SSAT)

The SSAT formula

$$\Phi = \underbrace{\text{Я}^{0.7}\, v_1, \exists\, e_1, \text{Я}^{0.6}\, v_2, \exists\, e_2.}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_1,v_2\}}\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 1, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_1, v_2) = v_1 \wedge v_2$ respectively, the matrix becomes tautological and has a weighted model count of 1.

- Syntax: $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\text{Я}^{p_i}, \exists\}$
- $\text{Я}^{p_i}\, v_i$ denotes that v_i will be assigned to $\top$ with probability p_i
- Existential variable v_i can depend on the randomized variables in $\{v_1, \ldots, v_{i-1}\}$
- Semantics: the *maximum satisfying probability* is $\Pr[\Phi] = p$ if $\max_{\mathcal{F}} \Pr[\Phi|_{\mathcal{F}}] = p$

Dependency Quantified Boolean Formula (DQBF)

Example 4 (DQBF)

The DQBF formula

$$\Phi = \underbrace{\forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}.\ \underbrace{(\bar{v}_1 \lor e_1 \lor \bar{v}_2)}_{(v_1 \land v_2) \implies e_1}\ \underbrace{(v_1 \lor \bar{e}_2)(v_2 \lor \bar{e}_2)(\bar{v}_1 \lor \bar{v}_2 \lor e_2)}_{e_2 \equiv (v_1 \land v_2)}$$

is unsatisfiable, since $v_1 \notin D_{e_2}$ and there is no Skolem function $f_{e_2}(v_2)$ for e_2 that satisfies the constraint $e_2 \equiv (v_1 \land v_2)$ imposed by the matrix.

Example 4 (DQBF)

The DQBF formula

$$\Phi = \underbrace{\forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is unsatisfiable, since $v_1 \notin D_{e_2}$ and there is no Skolem function $f_{e_2}(v_2)$ for e_2 that satisfies the constraint $e_2 \equiv (v_1 \wedge v_2)$ imposed by the matrix.

- Syntax: $\Phi = \forall u_1, \ldots \forall u_n, \exists e_1(D_{e_1}), \ldots, \exists e_m(D_{e_m}).\phi$

Dependency Quantified Boolean Formula (DQBF)

Example 4 (DQBF)

The DQBF formula

$$\Phi = \underbrace{\forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is unsatisfiable, since $v_1 \notin D_{e_2}$ and there is no Skolem function $f_{e_2}(v_2)$ for e_2 that satisfies the constraint $e_2 \equiv (v_1 \wedge v_2)$ imposed by the matrix.

- Syntax: $\Phi = \forall u_1, \ldots \forall u_n, \exists e_1(D_{e_1}), \ldots, \exists e_m(D_{e_m}).\phi$
- e_i can depend on the universal variables in its *dependency set* D_{e_i}

Dependency Quantified Boolean Formula (DQBF)

Example 4 (DQBF)

The DQBF formula

$$\Phi = \underbrace{\forall v_1, \forall v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

is unsatisfiable, since $v_1 \notin D_{e_2}$ and there is no Skolem function $f_{e_2}(v_2)$ for e_2 that satisfies the constraint $e_2 \equiv (v_1 \wedge v_2)$ imposed by the matrix.

- Syntax: $\Phi = \forall u_1, \ldots \forall u_n, \exists e_1(D_{e_1}), \ldots, \exists e_m(D_{e_m}).\phi$
- e_i can depend on the universal variables in its *dependency set* D_{e_i}
- Semantics: Φ is *satisfiable* if there is a strategy $\mathcal{F}$ such that $\Phi|_{\mathcal{F}} \equiv \top$

Dependency Stochastic Boolean Satisfiability (DSSAT)

Example 5 (DSSAT)

The DSSAT formula

$$\Phi = \underbrace{\text{Я}^{0.7}\, v_1, \text{Я}^{0.6}\, v_2, \exists\, e_1(\{v_1\}), \exists\, e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 0.82, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_2) = v_2$ respectively, the matrix becomes

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee \bar{v}_2)(v_2 \vee \bar{v}_2)(\bar{v}_1 \vee \bar{v}_2 \vee v_2) \equiv (v_1 \vee \bar{v}_2),$$

which has a weighted model count of 0.82, and no strategy can achieve a higher satisfying probability.

Dependency Stochastic Boolean Satisfiability (DSSAT)

Example 5 (DSSAT)

The DSSAT formula

$$\Phi = \underbrace{\text{Ⅎ}^{0.7} v_1, \text{Ⅎ}^{0.6} v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}. \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 0.82, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_2) = v_2$ respectively, the matrix becomes

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee \bar{v}_2)(v_2 \vee \bar{v}_2)(\bar{v}_1 \vee \bar{v}_2 \vee v_2) \equiv (v_1 \vee \bar{v}_2),$$

which has a weighted model count of 0.82, and no strategy can achieve a higher satisfying probability.

- Syntax: $\Phi = \text{Ⅎ}^{p_1} r_1, \ldots \text{Ⅎ}^{p_n} r_n, \exists e_1(D_{e_1}), \ldots, \exists e_m(D_{e_m}).\phi$

Dependency Stochastic Boolean Satisfiability (DSSAT)

Example 5 (DSSAT)

The DSSAT formula

$$\Phi = \underbrace{\text{Я}^{0.7} v_1, \text{Я}^{0.6} v_2, \exists e_1(\{v_1\}), \exists e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}. \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1} \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 0.82, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_2) = v_2$ respectively, the matrix becomes

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee \bar{v}_2)(v_2 \vee \bar{v}_2)(\bar{v}_1 \vee \bar{v}_2 \vee v_2) \equiv (v_1 \vee \bar{v}_2),$$

which has a weighted model count of 0.82, and no strategy can achieve a higher satisfying probability.

- Syntax: $\Phi = \text{Я}^{p_1} r_1, \ldots \text{Я}^{p_n} r_n, \exists e_1(D_{e_1}), \ldots, \exists e_m(D_{e_m}).\phi$
- Existential variable e_i can depend on the randomized variables in D_{e_i}

Dependency Stochastic Boolean Satisfiability (DSSAT)

Example 5 (DSSAT)

The DSSAT formula

$$\Phi = \underbrace{\reflectbox{R}^{0.7}\, v_1, \reflectbox{R}^{0.6}\, v_2, \exists\, e_1(\{v_1\}), \exists\, e_2(\{v_2\})}_{D_{e_1}=\{v_1\},\ D_{e_2}=\{v_2\}}.\ \underbrace{(\bar{v}_1 \vee e_1 \vee \bar{v}_2)}_{(v_1 \wedge v_2) \implies e_1}\ \underbrace{(v_1 \vee \bar{e}_2)(v_2 \vee \bar{e}_2)(\bar{v}_1 \vee \bar{v}_2 \vee e_2)}_{e_2 \equiv (v_1 \wedge v_2)}$$

has a maximum satisfying probability of 0.82, since after substituting e_1 and e_2 with $f_{e_1}(v_1) = \top$ and $f_{e_2}(v_2) = v_2$ respectively, the matrix becomes

$$(\bar{v}_1 \vee \top \vee \bar{v}_2)(v_1 \vee \bar{v}_2)(v_2 \vee \bar{v}_2)(\bar{v}_1 \vee \bar{v}_2 \vee v_2) \equiv (v_1 \vee \bar{v}_2),$$

which has a weighted model count of 0.82, and no strategy can achieve a higher satisfying probability.

- Syntax: $\Phi = \reflectbox{R}^{p_1}\, r_1, \ldots \reflectbox{R}^{p_n}\, r_n, \exists\, e_1(D_{e_1}), \ldots, \exists\, e_m(D_{e_m}).\phi$
- Existential variable e_i can depend on the randomized variables in D_{e_i}
- Semantics: the *maximum satisfying probability* is $\Pr[\Phi] = p$ if $\max_{\mathcal{F}} \Pr[\Phi|_{\mathcal{F}}] = p$.

Complexity of (D)QBF and (D)SSAT

PSPACE-complete NEXPTIME-complete

(D)QBF: decide satisfiability of Φ

(D)SSAT: decide whether $\Pr[\Phi] \geq t$

$$\begin{tikzcd}
\boxed{\text{QBF}} \arrow[r, "\text{nonlinear dependency}"] \arrow[d, "\text{randomness}"] & \boxed{\text{DQBF}} \arrow[d, "\text{randomness}"] \\
\boxed{\text{SSAT}} \arrow[r, "\text{nonlinear dependency}"] & \boxed{\text{DSSAT}}
\end{tikzcd}$$

Figure 3: Complexity of (D)QBF and (D)SSAT

Example 6 (DSSAT Encoding of PPEC)

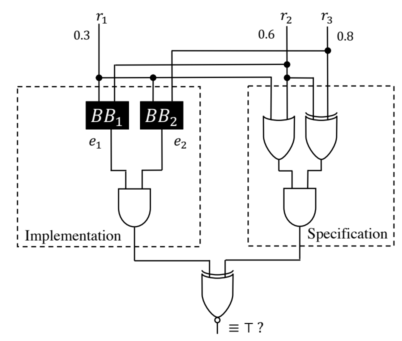

Figure 4: The probabilistic partial design in Example 1.

DSSAT Encoding of PPEC

Example 6 (DSSAT Encoding of PPEC)

1. The DSSAT encoding of PPEC in Example 1 is

$$\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$$

with $D_{e_1} = \{r_1, r_2\}$, $D_{e_2} = \{r_1, r_3\}$, and

$$\phi = ((r_1 \vee r_2)(r_2 \oplus r_3) \equiv (e_1 \wedge e_2)).$$

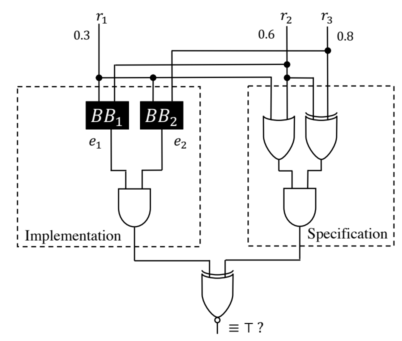

Figure 4: The probabilistic partial design in Example 1.

Example 6 (DSSAT Encoding of PPEC)

1. The DSSAT encoding of PPEC in Example 1 is

$$\Phi = \text{Ⴏ}^{0.3} r_1, \text{Ⴏ}^{0.6} r_2, \text{Ⴏ}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$$

with $D_{e_1} = \{r_1, r_2\}$, $D_{e_2} = \{r_1, r_3\}$, and

$$\phi = ((r_1 \vee r_2)(r_2 \oplus r_3) \equiv (e_1 \wedge e_2)).$$

2. The maximum satisfying probability $\Pr[\Phi] = 0.964$ is witnessed by the strategy $\mathcal{F} = (f_{e_1}, f_{e_2})$ for Φ, where

$$f_{e_1}(r_1, r_2) = (r_1 \oplus r_2), \ f_{e_2}(r_1, r_3) = (r_1 \equiv r_3),$$

which are optimal realizations of the black boxes.

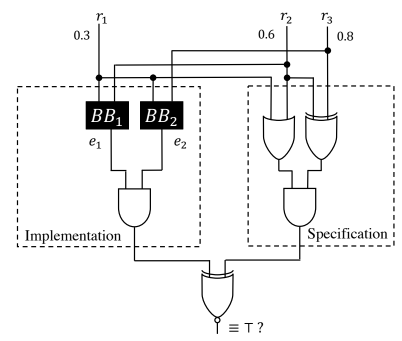

Figure 4: The probabilistic partial design in Example 1.

Towards a DSSAT Proof System

- In Example 6, given the DSSAT-encoded PPEC problem, we not only want to derive the maximum probability 0.964 of matching the implementation and the specification, we also want to synthesize an optimal implementation of the black boxes: $\mathrm{BB}_1 = (r_1 \oplus r_2)$, $\mathrm{BB}_2 = (r_1 \equiv r_3)$.

Towards a DSSAT Proof System

- In Example 6, given the DSSAT-encoded PPEC problem, we not only want to derive the maximum probability 0.964 of matching the implementation and the specification, we also want to synthesize an optimal implementation of the black boxes: $\mathrm{BB}_1 = (r_1 \oplus r_2)$, $\mathrm{BB}_2 = (r_1 \equiv r_3)$.
- Given a DSSAT formula Φ, we are interested in deriving its maximum satisfying probability $\Pr[\Phi] = p$ and a strategy $\mathcal{F}$ for Φ that witnesses $\Pr[\Phi|_{\mathcal{F}}] = p$.

Towards a DSSAT Proof System

- In Example 6, given the DSSAT-encoded PPEC problem, we not only want to derive the maximum probability 0.964 of matching the implementation and the specification, we also want to synthesize an optimal implementation of the black boxes: $\mathrm{BB}_1 = (r_1 \oplus r_2)$, $\mathrm{BB}_2 = (r_1 \equiv r_3)$.
- Given a DSSAT formula Φ, we are interested in deriving its maximum satisfying probability $\Pr[\Phi] = p$ and a strategy $\mathcal{F}$ for Φ that witnesses $\Pr[\Phi|_{\mathcal{F}}] = p$.
- Our goal: develop a proof system such that given a DSSAT formula Φ, it is able to gradually synthesize an optimal strategy $\mathcal{F}$ and derive the maximum satisfying probability $\Pr[\Phi]$ along the proof.

Outline

Proof System

- A *proof system* f for a language L is a system of *inference rules* that derives a sequence of *proof lines* from an input x.

Proof System

- A *proof system* f for a language L is a system of *inference rules* that derives a sequence of *proof lines* from an input x.
- Three required properties:

Proof System

- A *proof system* f for a language L is a system of *inference rules* that derives a sequence of *proof lines* from an input x.
- Three required properties:
  - *sound*: if there exists an f-proof for x, then $x \in L$

Proof System

- A *proof system* f for a language L is a system of *inference rules* that derives a sequence of *proof lines* from an input x.
- Three required properties:
  - *sound*: if there exists an f-proof for x, then $x \in L$
  - *complete*: for every $x \in L$, there exists an f-proof for x

Proof System

- A *proof system* f for a language L is a system of *inference rules* that derives a sequence of *proof lines* from an input x.
- Three required properties:
  - *sound*: if there exists an f-proof for x, then $x \in L$
  - *complete*: for every $x \in L$, there exists an f-proof for x
  - *polynomial-time verifiable*: every line in an f-proof can be checked whether it is derived by a valid rule in polynomial-time with respect to the previous lines

Proof System

- A proof system f for a language L is a system of inference rules that derives a sequence of proof lines from an input x.
- Three required properties:
  - sound: if there exists an f-proof for x, then $x \in L$
  - complete: for every $x \in L$, there exists an f-proof for x
  - polynomial-time verifiable: every line in an f-proof can be checked whether it is derived by a valid rule in polynomial-time with respect to the previous lines
- The later slides will present our proposed DSSAT proof system `DS-Res`, and will start with formalizing the `DS-Res` proof lines syntax and inference rules. Then, an explanation of how `DS-Res` is sound, complete, and polynomial-time verifiable will be given.

Outline

1. Introduction
2. Preliminaries
3. Proof System for DSSAT
   - Proof Lines
   - Inference Rules
   - Soundness, Completeness, and Polynomial-time Verifiability
4. Properties of DSSAT Proof System
5. Conclusion and Future Work

DS-Res Proof Lines

- Syntax:

$$(\underbrace{C}_{\text{clause}}, \underbrace{\mathcal{F}}_{\text{strategy}}, \underbrace{p}_{\text{probability}}, \underbrace{i}_{\text{iteration index}})$$ [4]

[4] $i \in \{0, 1, \ldots, n+m\}$, where $n = \#$ randomized variables, $m = \#$ existential variable

DS-Res Proof Lines

- Syntax:

$$\Big(\underset{\substack{\downarrow \\ \text{clause}}}{C},\quad \underset{\substack{\downarrow \\ \text{strategy}}}{\mathcal{F}},\quad \underset{\substack{\downarrow \\ \text{probability}}}{p},\quad \underset{\substack{\downarrow \\ \text{iteration index}^{4}}}{i}\Big)$$

  - Consider the DSSAT formula $\Phi = \text{ᴚ}^{0.3} r_1, \text{ᴚ}^{0.6} r_2, \text{ᴚ}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, the final proof line derived by `DS-Res` is:

$$\eta = (\{\bot\}, \{f_{e_1} = (r_1 \oplus r_2), f_{e_2} = (r_1 \equiv r_3)\}, 0.964, 5).$$

[4] $i \in \{0, 1, \ldots, n+m\}$, where $n = \#$ randomized variables, $m = \#$ existential variable

DS-Res Proof Lines

- Syntax:

$$\Big(\underset{\substack{\big\downarrow \\ \text{clause}}}{C},\quad \underset{\substack{\big\downarrow \\ \text{strategy}}}{\mathcal{F}},\quad \underset{\substack{\big\downarrow \\ \text{probability}}}{p},\quad \underset{\substack{\big\downarrow \\ \text{iteration index}}}{i}\Big)$$ [4]

  - Consider the DSSAT formula $\Phi = \reflectbox{\textsf{R}}^{0.3} r_1, \reflectbox{\textsf{R}}^{0.6} r_2, \reflectbox{\textsf{R}}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, the final proof line derived by `DS-Res` is:

$$\eta = (\{\perp\}, \{f_{e_1} = (r_1 \oplus r_2), f_{e_2} = (r_1 \equiv r_3)\}, 0.964, 5).$$

- Semantics:

[4] $i \in \{0, 1, \ldots, n+m\}$, where $n = \#$ randomized variables, $m = \#$ existential variable

DS-Res Proof Lines

- Syntax:

$$\underbrace{(C,}_{\text{clause}} \quad \underbrace{\mathcal{F},}_{\text{strategy}} \quad \underbrace{p,}_{\text{probability}} \quad \underbrace{i)}_{\text{iteration index [4]}}$$

  - Consider the DSSAT formula $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, the final proof line derived by DS-Res is:

$$\eta = (\{\bot\}, \{f_{e_1} = (r_1 \oplus r_2), f_{e_2} = (r_1 \equiv r_3)\}, 0.964, 5).$$

- Semantics:
  - Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

[4] $i \in \{0, 1, \ldots, n+m\}$, where $n = \#$ randomized variables, $m = \#$ existential variable

DS-Res Proof Lines

- Syntax:

$$\underset{\text{clause}}{(C,} \quad \underset{\text{strategy}}{\mathcal{F},} \quad \underset{\text{probability}}{p,} \quad \underset{\text{iteration index [4]}}{i)}$$

  - Consider the DSSAT formula $\Phi = \text{Ⅎ}^{0.3} r_1, \text{Ⅎ}^{0.6} r_2, \text{Ⅎ}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, the final proof line derived by DS-Res is:

$$\eta = (\{\perp\}, \{f_{e_1} = (r_1 \oplus r_2), f_{e_2} = (r_1 \equiv r_3)\}, 0.964, 5).$$

- Semantics:
  - Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.
  - Conisder η, the empty clause $(\perp)$ is automatically falsified, which means that $\Pr[\Phi|_{(f_{e_1}, f_{e_2})}] = 0.964$ and (f_{e_1}, f_{e_2}) is the optimal realization of the black boxes BB_1 and BB_2.

[4] $i \in \{0, 1, \ldots, n+m\}$, where $n = \#$ randomized variables, $m = \#$ existential variable

Outline

DS-Res Inference Rules: Overview

- Consider $\Phi = \reflectbox{R}^{0.3}\, r_1, \reflectbox{R}^{0.6}\, r_2, \reflectbox{R}^{0.8}\, r_3, \exists\, e_1(D_{e_1}), \exists\, e_2(D_{e_2}).\phi$ in Example 6, assume:

DS-Res Inference Rules: Overview

- Consider $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists\, e_1(D_{e_1}), \exists\, e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;

DS-Res Inference Rules: Overview

- Consider $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;

DS-Res Inference Rules: Overview

- Consider $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.

DS-Res Inference Rules: Overview

- Consider $\Phi = \text{Ʀ}^{0.3} r_1, \text{Ʀ}^{0.6} r_2, \text{Ʀ}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

DS-Res Inference Rules: Overview

- Consider $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

CNF

$(r_1 \vee r_2 \vee e_2) \in \psi$

$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi$

DS-Res Inference Rules: Overview

- Consider $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \lor r_2 \lor \bar{r}_3 \lor e_1)(\bar{r}_1 \lor r_2 \lor \bar{r}_3 \lor e_2)(\bar{r}_2 \lor r_3 \lor e_1)(\bar{r}_2 \lor r_3 \lor e_2)(r_1 \lor r_2 \lor \bar{e}_1 \lor \bar{e}_2)(r_2 \lor r_3 \lor \bar{e}_1 \lor \bar{e}_2)(\bar{r}_2 \lor \bar{r}_3 \lor \bar{e}_1 \lor \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \lor r_2 \lor e_2)(r_1 \lor r_2 \lor e_1)(\bar{r}_2 \lor r_3 \lor \bar{e}_1 \lor \bar{e}_2)(\bar{r}_2 \lor \bar{r}_3 \lor e_1)(\bar{r}_2 \lor \bar{r}_3 \lor e_2)(\bar{r}_1 \lor r_2 \lor \bar{r}_3 \lor \bar{e}_1 \lor \bar{e}_2)(r_2 \lor r_3 \lor e_2)(r_2 \lor r_3 \lor e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3, \; r_2 \mapsto 4, \; r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

$\boxed{\text{CNF}}$ $\downarrow$

$(r_1 \lor r_2 \lor e_2) \in \psi$

$\downarrow$ Axiom-1

$(r_2 \lor r_3 \lor \bar{e}_1 \lor \bar{e}_2) \in \phi$

$\downarrow$ Axiom-0

DS-Res Inference Rules: Overview

- Consider $\Phi = \text{ꓤ}^{0.3} r_1, \text{ꓤ}^{0.6} r_2, \text{ꓤ}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

CNF $\rightarrow$ iter 0

$$(r_1 \vee r_2 \vee e_2) \in \psi \xrightarrow{\textsc{Axiom-1}} (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$$

$$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi \xrightarrow{\textsc{Axiom-0}} (\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0)$$

DS-Res Inference Rules: Overview

- Consider $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

$\boxed{\text{CNF}} \rightarrow \boxed{\text{iter 0}} \xrightarrow{\text{pivot 1: } e_2,\ \text{pivot 2: } e_1}$

$(r_1 \vee r_2 \vee e_2) \in \psi \xrightarrow{\textsc{Axiom-1}} (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0) \rightarrow$

$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi \xrightarrow{\textsc{Axiom-0}} (\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) \xrightarrow{\exists\textsc{-Reduction}}$

DS-Res Inference Rules: Overview

- Consider $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3, \ r_2 \mapsto 4, \ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

CNF

↓

iter 0

↓ pivot 1: e_2 pivot 2: e_1

iter 2

$$(r_1 \vee r_2 \vee e_2) \in \psi \xrightarrow{\textsc{Axiom-1}} (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0) \longrightarrow (\{r_1, r_2\}, \{f_{e_1} = *, f_{e_2} = \bot\}, 1, 2)$$

$$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi \xrightarrow{\textsc{Axiom-0}} (\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) \xrightarrow{\exists\text{-}\textsc{Reduction}} (\{r_2, r_3\}, \{f_{e_1} = \top, f_{e_2} = \top\}, 0, 2)$$

- Consider $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

CNF → iter 0 (pivot 1: e_2, pivot 2: e_1) → iter 2 (pivot 3: r_3) →

$(r_1 \vee r_2 \vee e_2) \in \psi$ $\xrightarrow{\text{Axiom-1}}$ $(\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$ $\rightarrow$ $(\{r_1, r_2\}, \{f_{e_1} = *, f_{e_2} = \bot\}, 1, 2)$

$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi$ $\xrightarrow{\text{Axiom-0}}$ $(\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0)$ $\xrightarrow{\exists\text{-Reduction}}$ $(\{r_2, r_3\}, \{f_{e_1} = \top, f_{e_2} = \top\}, 0, 2)$

$\rotatebox[origin=c]{180}{R}$-Resolution

DS-Res Inference Rules: Overview

- Consider $\Phi = Я^{0.3} r_1, Я^{0.6} r_2, Я^{0.8} r_3, \exists e_1(D_{e_1}), \exists e_2(D_{e_2}).\phi$ in Example 6, assume:
  - ϕ in CNF: $\phi = (\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_2 \vee r_3 \vee e_1)(\bar{r}_2 \vee r_3 \vee e_2)(r_1 \vee r_2 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)$;
  - $\psi \equiv \neg\phi$ in CNF: $\psi = (r_1 \vee r_2 \vee e_2)(r_1 \vee r_2 \vee e_1)(\bar{r}_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2)(\bar{r}_2 \vee \bar{r}_3 \vee e_1)(\bar{r}_2 \vee \bar{r}_3 \vee e_2)(\bar{r}_1 \vee r_2 \vee \bar{r}_3 \vee \bar{e}_1 \vee \bar{e}_2)(r_2 \vee r_3 \vee e_2)(r_2 \vee r_3 \vee e_1)$;
  - variables indexed as: $\{e_2 \mapsto 1, e_1 \mapsto 2, r_3 \mapsto 3,\ r_2 \mapsto 4,\ r_1 \mapsto 5\}$.
- DS-Res proof schematic: a variable's index is its order of being selected as pivot.

CNF $\rightarrow$ iter 0 $\xrightarrow{\text{pivot 1: } e_2,\ \text{pivot 2: } e_1}$ iter 2 $\xrightarrow{\text{pivot 3: } r_3}$ iter 3

$(r_1 \vee r_2 \vee e_2) \in \psi \xrightarrow{\text{AXIOM-1}} (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0) \rightarrow (\{r_1, r_2\}, \{f_{e_1} = *, f_{e_2} = \bot\}, 1, 2)$

$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi \xrightarrow{\text{AXIOM-0}} (\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) \xrightarrow{\exists\text{-REDUCTION}} (\{r_2, r_3\}, \{f_{e_1} = \top, f_{e_2} = \top\}, 0, 2)$

$Я$-RESOLUTION: $(\{r_1, r_2\}, \{f_{e_1} = \top, f_{e_2} = \bar{r}_3\}, 0.8, 3)$

DS-Res Inference Rules: Overview

- Given a DSSAT formula $\Phi = \mathcal{Q}.\phi$ over the variables $\{v_1, \ldots, v_{n+m}\}$.[5] Assume that a quantifier-free formula $\psi \equiv \neg\phi$ is given, and both ϕ and ψ are CNF.

[5] For every $r \in D_e$, e has smaller index than r. Note that $n = \#$ randomized variables, $m = \#$ existential variables.

DS-Res Inference Rules: Overview

- Given a DSSAT formula $\Phi = \mathcal{Q}.\phi$ over the variables $\{v_1, \ldots, v_{n+m}\}$.[5] Assume that a quantifier-free formula $\psi \equiv \neg\phi$ is given, and both ϕ and ψ are CNF.
- `DS-Res` iterates through the indices $\{0, 1, \ldots, n+m\}$ and derives i-*proof lines* $(C, \mathcal{F}, p, i)$ in each i-iteration.

[5] For every $r \in D_e$, e has smaller index than r. Note that $n = \#$ randomized variables, $m = \#$ existential variables.

DS-Res Inference Rules: Overview

- Given a DSSAT formula $\Phi = \mathcal{Q}.\phi$ over the variables $\{v_1, \ldots, v_{n+m}\}$.[5] Assume that a quantifier-free formula $\psi \equiv \neg\phi$ is given, and both ϕ and ψ are CNF.
- `DS-Res` iterates through the indices $\{0, 1, \ldots, n+m\}$ and derives i-*proof lines* $(C, \mathcal{F}, p, i)$ in each i-iteration.
  - The base-case 0-proof lines are derived from ϕ and ψ by the *axiom* rule.

[5] For every $r \in D_e$, e has smaller index than r. Note that $n = \#$ randomized variables, $m = \#$ existential variables.

DS-Res Inference Rules: Overview

- Given a DSSAT formula $\Phi = \mathcal{Q}.\phi$ over the variables $\{v_1, \ldots, v_{n+m}\}$.[5] Assume that a quantifier-free formula $\psi \equiv \neg\phi$ is given, and both ϕ and ψ are CNF.
- `DS-Res` iterates through the indices $\{0, 1, \ldots, n+m\}$ and derives i*-proof lines* $(C, \mathcal{F}, p, i)$ in each i-iteration.
  - The base-case 0-proof lines are derived from ϕ and ψ by the *axiom* rule.
  - The inductive-case i-proof lines (with $i > 0$) are derived from $(i-1)$-proof lines by either the *existential reduction* rule or the *random resolution* rule, depending on whether v_i is existential or randomized.

[5] For every $r \in D_e$, e has smaller index than r. Note that $n = \#$ randomized variables, $m = \#$ existential variables.

- Given a DSSAT formula $\Phi = \mathcal{Q}.\phi$ over the variables $\{v_1, \ldots, v_{n+m}\}$.[5] Assume that a quantifier-free formula $\psi \equiv \neg\phi$ is given, and both ϕ and ψ are CNF.
- DS-Res iterates through the indices $\{0, 1, \ldots, n+m\}$ and derives i-*proof lines* $(C, \mathcal{F}, p, i)$ in each i-iteration.
  - The base-case 0-proof lines are derived from ϕ and ψ by the *axiom* rule.
  - The inductive-case i-proof lines (with $i > 0$) are derived from $(i-1)$-proof lines by either the *existential reduction* rule or the *random resolution* rule, depending on whether v_i is existential or randomized.
  - In the i-iteration, DS-Res eliminates the variable v_i from clauses and refines the learned strategies by considering v_i.

[5] For every $r \in D_e$, e has smaller index than r. Note that $n = \#$ randomized variables, $m = \#$ existential variables.

DS-Res Inference Rules: Overview

- Given a DSSAT formula $\Phi = \mathcal{Q}.\phi$ over the variables $\{v_1, \ldots, v_{n+m}\}$.[5] Assume that a quantifier-free formula $\psi \equiv \neg\phi$ is given, and both ϕ and ψ are CNF.
- `DS-Res` iterates through the indices $\{0, 1, \ldots, n+m\}$ and derives *i-proof lines* $(C, \mathcal{F}, p, i)$ in each i-iteration.
  - The base-case 0-proof lines are derived from ϕ and ψ by the *axiom* rule.
  - The inductive-case i-proof lines (with $i > 0$) are derived from $(i-1)$-proof lines by either the *existential reduction* rule or the *random resolution* rule, depending on whether v_i is existential or randomized.
  - In the i-iteration, `DS-Res` eliminates the variable v_i from clauses and refines the learned strategies by considering v_i.
  - In the terminal $(n+m)$-iteration, terminal proof lines $(\perp, \mathcal{F}, p, n+m)$ are derived, entailing the semantics $\Pr[\Phi|_{\mathcal{F}}] = p$.

[5] For every $r \in D_e$, e has smaller index than r. Note that $n = \#$ randomized variables, $m = \#$ existential variables.

DS-Res Inference Rules: Axiom

- Example:

$$(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi \quad\quad (r_1 \vee r_2 \vee e_2) \in \psi$$

$$\downarrow \textsc{Axiom-0} \quad\quad \downarrow \textsc{Axiom-1}$$

$$(\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) \quad (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$$

DS-Res Inference Rules: Axiom

- Example:

$$\begin{array}{cc} (r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi & (r_1 \vee r_2 \vee e_2) \in \psi \\ \downarrow \textsc{Axiom-0} & \downarrow \textsc{Axiom-1} \\ (\underset{\sim\sim\sim}{\{r_2, r_3, \bar{e}_1, \bar{e}_2\}}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) & (\underset{\sim\sim\sim}{\{r_1, r_2, e_2\}}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0) \end{array}$$

- Intuitively, the *no-assignment* $*$ represents an arbitrary value of $\top$ or $\bot$ and $f_e = *$ represents an arbitrary function that assigns e to an arbitrary value.

DS-Res Inference Rules: Axiom

- Example:

$$\begin{array}{cc}
(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi & (r_1 \vee r_2 \vee e_2) \in \psi \\
\downarrow \textsc{Axiom-0} & \downarrow \textsc{Axiom-1} \\
(\underset{\sim}{\{r_2, r_3, \bar{e}_1, \bar{e}_2\}}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) & (\underset{\sim}{\{r_1, r_2, e_2\}}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)
\end{array}$$

- Intuitively, the *no-assignment* $*$ represents an arbitrary value of $\top$ or $\bot$ and $f_e = *$ represents an arbitrary function that assigns e to an arbitrary value.
- Recall the semantics of DS-Res proof lines.[6]

[6] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

DS-Res Inference Rules: Axiom

- Example:

$$\begin{array}{cc} (r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi & (r_1 \vee r_2 \vee e_2) \in \psi \\ \downarrow \text{AXIOM-0} & \downarrow \text{AXIOM-1} \\ (\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0) & (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0) \end{array}$$

- Intuitively, the *no-assignment* $*$ represents an arbitrary value of $\top$ or $\bot$ and $f_e = *$ represents an arbitrary function that assigns e to an arbitrary value.
- Recall the semantics of DS-Res proof lines.[6]
  - For AXIOM-0, when $(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi$ is falsified, then ϕ is also falsified, so an arbitrary strategy cannot satisfy ϕ, thus resulting in the satisfying probability 0.

[6]Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

DS-Res Inference Rules: Axiom

- Example:

$$\frac{(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi}{(\{r_2, r_3, \bar{e}_1, \bar{e}_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 0, 0)}\ \textsc{Axiom-0} \qquad \frac{(r_1 \vee r_2 \vee e_2) \in \psi}{(\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)}\ \textsc{Axiom-1}$$

- Intuitively, the *no-assignment* $*$ represents an arbitrary value of $\top$ or $\perp$ and $f_e = *$ represents an arbitrary function that assigns e to an arbitrary value.
- Recall the semantics of DS-Res proof lines.[6]
  - For Axiom-0, when $(r_2 \vee r_3 \vee \bar{e}_1 \vee \bar{e}_2) \in \phi$ is falsified, then ϕ is also falsified, so an arbitrary strategy cannot satisfy ϕ, thus resulting in the satisfying probability 0.
  - For Axiom-1, when $(r_1 \vee r_2 \vee e_2) \in \psi$ is falsified, then ψ is also falsified (and $\phi \equiv \neg\psi$ is satisfied), so an arbitrary strategy satisfies ϕ, thus resulting in the satisfying probability 1.

[6] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

DS-Res Inference Rules: Axiom

$$\textsc{Axiom-0}\ \frac{C \in \phi}{(C, \mathcal{F}^*, 0, 0)} \qquad \textsc{Axiom-1}\ \frac{C \in \psi}{(C, \mathcal{F}^*, 1, 0)}$$

DS-Res Inference Rules: Axiom

$$\textsc{Axiom-0}\ \frac{C \in \phi}{(C, \mathcal{F}^*, {\color{blue}0}, 0)} \qquad \textsc{Axiom-1}\ \frac{C \in \psi}{(C, \mathcal{F}^*, {\color{blue}1}, 0)}$$

- $\mathcal{F}^* := (f_e^*)_{e \in E}$, where each existential Skolem function $f_e^* = *$ assigns e to the *no-assignment* $*$.

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and e_2 has index 1.

- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and e_2 has index 1.

$\boxed{\text{iter } 0}$ $\qquad \eta_0 = (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and e_2 has index 1.

$$\boxed{\text{iter } 0} \qquad \eta_0 = (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$$

$\downarrow$ pivot: e_2 $\qquad\qquad$ $\downarrow$ $\exists$-REDUCTION

DS-Res Inference Rules: Existential Reduction

- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and e_2 has index 1.

$$\boxed{\text{iter } 0} \qquad \eta_0 = (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$$

$$\Big\downarrow \text{pivot: } e_2 \qquad\qquad \Big\downarrow \exists\text{-REDUCTION}$$

$$\boxed{\text{iter } 1} \qquad \eta_1 = (\{r_1, r_2\}, \{f_{e_1} = *, f_{e_2} = \perp\}, 1, 1)$$

DS-Res Inference Rules: Existential Reduction

- Example: $\Phi = \text{ꓤ}^{0.3} r_1, \text{ꓤ}^{0.6} r_2, \text{ꓤ}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and e_2 has index 1.

$$\boxed{\text{iter } 0} \qquad \eta_0 = (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0)$$

$$\Big\downarrow \text{pivot: } e_2 \qquad\qquad \Big\downarrow \exists\text{-Reduction}$$

$$\boxed{\text{iter } 1} \qquad \eta_1 = (\{r_1, r_2\}, \{f_{e_1} = *, f_{e_2} = \bot\}, 1, 1)$$

- $\exists$-Reduction eliminates e_2 from the clause of η_0 and refines the strategy $f_{e_2} = *$ in η_0 into the one $f_{e_2} = \bot$ in η_1 that *fasifies* the literal e_2.

DS-Res Inference Rules: Existential Reduction

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and e_2 has index 1.

$$
\begin{array}{ccc}
\boxed{\text{iter } 0} & & \eta_0 = (\{r_1, r_2, e_2\}, \{f_{e_1} = *, f_{e_2} = *\}, 1, 0) \\
\Big\downarrow \text{pivot: } e_2 & & \Big\downarrow \exists\text{-}\textsc{Reduction} \\
\boxed{\text{iter } 1} & & \eta_1 = (\{r_1, r_2\}, \{f_{e_1} = *, f_{e_2} = \bot\}, 1, 1)
\end{array}
$$

- $\exists$-Reduction eliminates e_2 from the clause of η_0 and refines the strategy $f_{e_2} = *$ in η_0 into the one $f_{e_2} = \bot$ in η_1 that *fasifies* the literal e_2.
- Recall the semantics of DS-Res proof lines,[7] the condition that $(r_1 \vee r_2 \vee e_2)$ is falsified and $(f_{e_1} = *, f_{e_2} = *)$ is adopted is equivalent to the condition that $(r_1 \vee r_2)$ is falsified and $(f_{e_1} = *, f_{e_2} = \bot)$ is adopted, so both conditions result in the same satisfying probability 1.

[7] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

DS-Res Inference Rules: Existential Reduction

$$\exists\text{-}\textsc{Reduction}\ \frac{(C, \mathcal{F}, p, i-1) : \begin{cases} \neg l \notin C, \\ \mathrm{var}(l) = v_i \text{ is existential} \end{cases}}{(C \setminus \{l\}, \mathcal{F}^l, p, i)}$$

DS-Res Inference Rules: Existential Reduction

$$\exists\text{-}\textsc{Reduction}\ \frac{(C, \mathcal{F}, p, i-1) : \begin{cases} \neg l \notin C, \\ \mathrm{var}(l) = v_i \text{ is existential} \end{cases}}{(C \setminus \{l\}, \mathcal{F}^l, p, i)}$$

- $\mathcal{F}^l$ replaces the Skolem function $f_{v_i} = *$ in $\mathcal{F}$ with $f^l_{v_i} := \begin{cases} \bot, & \text{if } l = v_i \in C, \\ \top, & \text{if } l = \neg v_i \in C, \\ *, & \text{if } v_i, \neg v_i \notin C. \end{cases}$

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \reflectbox{R}^{0.3}\, r_1, \reflectbox{R}^{0.6}\, r_2, \reflectbox{R}^{0.8}\, r_3, \exists\, e_1(\{r_1, r_2\}), \exists\, e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter 3}}$ $\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

iter 3

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

pivot: r_2

Я-Resolution

- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3}\, r_1, \rotatebox[origin=c]{180}{R}^{0.6}\, r_2, \rotatebox[origin=c]{180}{R}^{0.8}\, r_3, \exists\, e_1(\{r_1, r_2\}), \exists\, e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$$\begin{array}{c} \boxed{\text{iter } 3} \\ \Big\downarrow \text{ pivot: } r_2 \\ \boxed{\text{iter } 4} \end{array}$$

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\rotatebox[origin=c]{180}{R}$-Resolution

$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists\, e_1(\{r_1, r_2\}), \exists\, e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter 3}} \xrightarrow{\text{pivot: } r_2} \boxed{\text{iter 4}}$

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3) \quad \eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\rotatebox[origin=c]{180}{R}$-Resolution

$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- $\rotatebox[origin=c]{180}{R}$-Resolution derives η by

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \text{ꓤ}^{0.3} r_1, \text{ꓤ}^{0.6} r_2, \text{ꓤ}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

iter 3 $\qquad \eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3) \quad \eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\downarrow$ pivot: r_2 $\qquad$ ꓤ-Resolution

iter 4 $\qquad \eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- ꓤ-Resolution derives η by
  1. performing weakened-resolution: $\{\neg r_1\} = (\{\neg r_2\} \setminus \{\neg r_2\}) \cup (\{\neg r_1, r_2\} \setminus \{r_2\})$,[8]

[8] Applying the weakened-resolution on the C_0, C_1 is equivalent to first weakening C_0, C_1 into $(C_0 \cup \{\neg r\}), (C_1 \cup \{r\})$ respectively and then performing the standard resolution on the weakened clauses [JHB08].

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter 3}} \xrightarrow{\text{pivot: } r_2} \boxed{\text{iter 4}}$

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\reflectbox{R}$-Resolution

$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- $\reflectbox{R}$-Resolution derives η by
  1. performing weakened-resolution: $\{\neg r_1\} = (\{\neg r_2\} \setminus \{\neg r_2\}) \cup (\{\neg r_1, r_2\} \setminus \{r_2\})$,[8]
  2. merging strategies: $\begin{cases} r_2 \in D_{e_1} \implies f_{e_1} = g_{e_1} \overset{r_2}{\bowtie} h_{e_1} = \top \overset{r_2}{\bowtie} \bot = r_2 & \text{(if-then-else)},^{9} \\ r_2 \notin D_{e_2} \implies f_{e_2} = g_{e_2} \circ h_{e_2} = \neg r_3 \circ * = \neg r_3 & \text{(composition)},^{10} \end{cases}$

---

[8] Applying the weakened-resolution on the C_0, C_1 is equivalent to first weakening C_0, C_1 into $(C_0 \cup \{\neg r\}), (C_1 \cup \{r\})$ respectively and then performing the standard resolution on the weakened clauses [JHB08].

[9] $f_{e_1} = g_{e_1} \overset{r_2}{\bowtie} h_{e_1}$ is the if-then-else function: if assignment to r_2 is $\top$ then follows g_{e_1} else h_{e_1}.

[10] $f_{e_2} = g_{e_2} \circ h_{e_2}$ is the composite function: for consistent values $s, t \in \{*, \top, \bot\}$, the composite value $s \circ t$ equals s if $t = *$, else equals t if $s = *$, otherwise equals $s = t$.

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

iter 3 $\xrightarrow{\text{pivot: } r_2}$ iter 4

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\quad$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\reflectbox{R}$-Resolution

$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- $\reflectbox{R}$-Resolution derives η by
  1. performing weakened-resolution: $\{\neg r_1\} = (\{\neg r_2\} \setminus \{\neg r_2\}) \cup (\{\neg r_1, r_2\} \setminus \{r_2\})$,[8]
  2. merging strategies: $\begin{cases} r_2 \in D_{e_1} \implies f_{e_1} = g_{e_1} \overset{r_2}{\bowtie} h_{e_1} = \top \overset{r_2}{\bowtie} \bot = r_2 \text{ (if-then-else)},^{9} \\ r_2 \notin D_{e_2} \implies f_{e_2} = g_{e_2} \circ h_{e_2} = \neg r_3 \circ * = \neg r_3 \text{ (composition)},^{10} \end{cases}$
  3. taking weighted average: $0.68 = 0.6 \times 1 + (1 - 0.6) \times 0.2$.

[8] Applying the weakened-resolution on the C_0, C_1 is equivalent to first weakening C_0, C_1 into $(C_0 \cup \{\neg r\}), (C_1 \cup \{r\})$ respectively and then performing the standard resolution on the weakened clauses [JHB08].

[9] $f_{e_1} = g_{e_1} \overset{r_2}{\bowtie} h_{e_1}$ is the if-then-else function: if assignment to r_2 is $\top$ then follows g_{e_1} else h_{e_1}.

[10] $f_{e_2} = g_{e_2} \circ h_{e_2}$ is the composite function: for consistent values $s, t \in \{*, \top, \bot\}$, the composite value $s \circ t$ equals s if $t = *$, else equals t if $s = *$, otherwise equals $s = t$.

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter 3}}$ $\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \text{ꓤ}^{0.3}\, r_1, \text{ꓤ}^{0.6}\, r_2, \text{ꓤ}^{0.8}\, r_3, \exists\, e_1(\{r_1, r_2\}), \exists\, e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

iter 3

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

pivot: r_2

ꓤ-Resolution

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \reflectbox{R}^{0.3}\, r_1, \reflectbox{R}^{0.6}\, r_2, \reflectbox{R}^{0.8}\, r_3, \exists\, e_1(\{r_1, r_2\}), \exists\, e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter 3}}$ $\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

pivot: r_2

$\reflectbox{R}$-Resolution

$\boxed{\text{iter 4}}$ $\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \text{Ⅎ}^{0.3} r_1, \text{Ⅎ}^{0.6} r_2, \text{Ⅎ}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$$\boxed{\text{iter 3}} \xrightarrow{\text{pivot: } r_2} \boxed{\text{iter 4}}$$

$$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3) \quad \eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$$

Ⅎ-Resolution

$$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$$

- Semantically,[11] since the clause $(\neg r_1) \in \eta$ is falsified by an assignment $\alpha : r_1 \mapsto \top$, we can construct the assignments

[11]Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

- Example: $\Phi = \text{Ʉ}^{0.3} r_1, \text{Ʉ}^{0.6} r_2, \text{Ʉ}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter } 3}$ $\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

pivot: r_2

Ʉ-Resolution

$\boxed{\text{iter } 4}$ $\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- Semantically,[11] since the clause $(\neg r_1) \in \eta$ is falsified by an assignment $\alpha : r_1 \mapsto \top$, we can construct the assignments
  - $\alpha^0 := \alpha \cup \{r_2 \mapsto \top\}$ falsifies the clause in η_0 with $\Pr[$satisfy ϕ given α^0 and $(g_{e_1}, g_{e_2})] = 1$

[11] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[$ satisfy $\phi \mid C$ is falsified, $\mathcal{F}$ is adopted $] = p$.

- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

$\boxed{\text{iter 3}}$ $\xrightarrow{\text{pivot: } r_2}$ $\boxed{\text{iter 4}}$

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\quad$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\rotatebox[origin=c]{180}{R}$-Resolution

$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- Semantically,[11] since the clause $(\neg r_1) \in \eta$ is falsified by an assignment $\alpha : r_1 \mapsto \top$, we can construct the assignments
  - $\alpha^0 := \alpha \cup \{r_2 \mapsto \top\}$ falsifies the clause in η_0 with $\Pr[\text{satisfy } \phi \text{ given } \alpha^0 \text{ and } (g_{e_1}, g_{e_2})] = 1$
  - $\alpha^1 := \alpha \cup \{r_2 \mapsto \bot\}$ falsifies the clause in η_1 with $\Pr[\text{satisfy } \phi \text{ given } \alpha^1 \text{ and } (h_{e_1}, h_{e_2})] = 0.2$

[11] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[\text{ satisfy } \phi \mid C \text{ is falsified, } \mathcal{F} \text{ is adopted }] = p$.

DS-Res Inference Rules: Random Resolution

- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2 has index 4.

iter 3 $\xrightarrow{\text{pivot: } r_2}$ iter 4

$\eta_0 = (\{\neg r_2\}, \{g_{e_1} = \top, g_{e_2} = \neg r_3\}, 1, 3)$ $\eta_1 = (\{\neg r_1, r_2\}, \{h_{e_1} = \bot, h_{e_2} = *\}, 0.2, 3)$

$\reflectbox{R}$-Resolution

$\eta = (\{\neg r_1\}, \{f_{e_1} = r_2, f_{e_2} = \neg r_3\}, 0.68, 4)$

- Semantically,[11] since the clause $(\neg r_1) \in \eta$ is falsified by an assignment $\alpha : r_1 \mapsto \top$, we can construct the assignments
  - $\alpha^0 := \alpha \cup \{r_2 \mapsto \top\}$ falsifies the clause in η_0 with $\Pr[\text{satisfy } \phi \text{ given } \alpha^0 \text{ and } (g_{e_1}, g_{e_2})] = 1$
  - $\alpha^1 := \alpha \cup \{r_2 \mapsto \bot\}$ falsifies the clause in η_1 with $\Pr[\text{satisfy } \phi \text{ given } \alpha^1 \text{ and } (h_{e_1}, h_{e_2})] = 0.2$
  - Hence,

$$\begin{aligned}\Pr[\text{satisfy } \phi \text{ given } \alpha \text{ and } (f_{e_1}, f_{e_2})] &= \Pr[r_2 \mapsto \top] \cdot 1 + \Pr[r_2 \mapsto \bot] \cdot 0.2 \\ &= 0.6 \times 1 + (1 - 0.6) \times 0.2 = 0.68\end{aligned}$$

[11] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[\text{ satisfy } \phi \mid C \text{ is falsified, } \mathcal{F} \text{ is adopted }] = p$.

DS-Res Inference Rules: Random Resolution

$$\text{Я-Resolution}\ \frac{\begin{array}{l}(C_0, \mathcal{F}^0, p_0, i-1), (C_1, \mathcal{F}^1, p_1, i-1) : \\ \begin{cases} v_i \text{ is randomized}, v_i \notin C_0,\ \neg v_i \notin C_1,\ \text{and } C_0 \dot{\cup}_{v_i} C_1 \text{ is not a tautology}, \\ f_e^0 \in \mathcal{F}^0 \text{ is } \textit{consistent} \text{ with } f_e^1 \in \mathcal{F}^1,\ \text{for each } e \text{ with } v_i \notin D_e.^{12} \end{cases}\end{array}}{(C_0 \dot{\cup}_{v_i} C_1, \mathcal{F}^{0,1}, p^{0,1}, i)}$$

The random resolution rule takes two input $(i-1)$-proof lines and derives an i-proof line by

[12] For $s, t \in \{*, \top, \bot\}$, s is consistent with t if $s = t$ or one of s, t is $*$.

DS-Res Inference Rules: Random Resolution

$$\text{Я-Resolution}\ \frac{\begin{array}{l}(C_0, \mathcal{F}^0, p_0, i-1), (C_1, \mathcal{F}^1, p_1, i-1) : \\ \begin{cases} v_i \text{ is randomized}, v_i \notin C_0,\ \neg v_i \notin C_1,\ \text{and } C_0 \mathbin{\dot\cup}_{v_i} C_1 \text{ is not a tautology}, \\ f_e^0 \in \mathcal{F}^0 \text{ is } \textit{consistent} \text{ with } f_e^1 \in \mathcal{F}^1,\ \text{for each } e \text{ with } v_i \notin D_e.^{12} \end{cases}\end{array}}{(C_0 \mathbin{\dot\cup}_{v_i} C_1, \mathcal{F}^{0,1}, p^{0,1}, i)}$$

The random resolution rule takes two input $(i-1)$-proof lines and derives an i-proof line by

1. performing weakened-resolution: $C_0 \mathbin{\dot\cup}_{v_i} C_1 \coloneqq (C_0 \setminus \{\neg v_i\}) \cup (C_1 \setminus \{v_i\})$,

[12] For $s, t \in \{*, \top, \bot\}$, s is consistent with t if $s = t$ or one of s, t is $*$.

DS-Res Inference Rules: Random Resolution

$$\text{Ы-Resolution}\ \frac{\begin{array}{c}(C_0, \mathcal{F}^0, p_0, i-1), (C_1, \mathcal{F}^1, p_1, i-1) : \\ \begin{cases} v_i \text{ is randomized}, v_i \notin C_0,\ \neg v_i \notin C_1,\ \text{and } C_0 \dot{\cup}_{v_i} C_1 \text{ is not a tautology}, \\ f_e^0 \in \mathcal{F}^0 \text{ is } \textit{consistent} \text{ with } f_e^1 \in \mathcal{F}^1,\ \text{for each } e \text{ with } v_i \notin D_e.^{12} \end{cases}\end{array}}{(C_0 \dot{\cup}_{v_i} C_1, \mathcal{F}^{0,1}, p^{0,1}, i)}$$

The random resolution rule takes two input $(i-1)$-proof lines and derives an i-proof line by

1. performing weakened-resolution: $C_0 \dot{\cup}_{v_i} C_1 := (C_0 \setminus \{\neg v_i\}) \cup (C_1 \setminus \{v_i\})$,
2. merging stratgies: $\mathcal{F}^{0,1}$ is the list of functions $f_e^{0,1} := \begin{cases} f_e^0 \overset{v_i}{\bowtie} f_e^1 & \text{if } v_i \in D_e,^{13} \\ f_e^0 \circ f_e^1 & \text{if } v_i \notin D_e,^{14} \end{cases}$ and

[12] For $s, t \in \{*, \top, \bot\}$, s is consistent with t if $s = t$ or one of s, t is $*$.

[13] $f_e^0 \overset{v_i}{\bowtie} f_e^1$ is the if-then-else function: if assignment to v_i is $\top$ then follows f_e^0 else f_e^1

[14] $f_e^0 \circ f_e^1$ is the composite function: for consistent values $s, t \in \{*, \top, \bot\}$, the composite value $s \circ t$ equals s if $t = *$, else equals t if $s = *$, otherwise equals $s = t$.

DS-Res Inference Rules: Random Resolution

$$
\text{Я-RESOLUTION}\ \frac{\begin{array}{l}(C_0, \mathcal{F}^0, p_0, i-1), (C_1, \mathcal{F}^1, p_1, i-1) : \\ \begin{cases} v_i \text{ is randomized}, v_i \notin C_0,\ \neg v_i \notin C_1,\ \text{and } C_0 \dot{\cup}_{v_i} C_1 \text{ is not a tautology}, \\ f_e^0 \in \mathcal{F}^0 \text{ is } \textit{consistent} \text{ with } f_e^1 \in \mathcal{F}^1,\ \text{for each } e \text{ with } v_i \notin D_e.^{12} \end{cases}\end{array}}{(C_0 \dot{\cup}_{v_i} C_1, \mathcal{F}^{0,1}, p^{0,1}, i)}
$$

The random resolution rule takes two input $(i-1)$-proof lines and derives an i-proof line by

1. performing weakened-resolution: $C_0 \dot{\cup}_{v_i} C_1 := (C_0 \setminus \{\neg v_i\}) \cup (C_1 \setminus \{v_i\})$,
2. merging stratgies: $\mathcal{F}^{0,1}$ is the list of functions $f_e^{0,1} := \begin{cases} f_e^0 \overset{v_i}{\bowtie} f_e^1 & \text{if } v_i \in D_e,^{13} \\ f_e^0 \circ f_e^1 & \text{if } v_i \notin D_e,^{14} \end{cases}$ and
3. taking weighted average: $p^{0,1} := p_{v_i} \cdot p_0 + (1 - p_{v_i}) \cdot p_1$.

---

[12] For $s, t \in \{*, \top, \bot\}$, s is consistent with t if $s = t$ or one of s, t is $*$.

[13] $f_e^0 \overset{v_i}{\bowtie} f_e^1$ is the if-then-else function: if assignment to v_i is $\top$ then follows f_e^0 else f_e^1

[14] $f_e^0 \circ f_e^1$ is the composite function: for consistent values $s, t \in \{*, \top, \bot\}$, the composite value $s \circ t$ equals s if $t = *$, else equals t if $s = *$, otherwise equals $s = t$.

Outline

Soundness of DS-Res

Theorem 7 (Soundness of DS-Res)

Given a DSSAT formula Φ, if some proof line $(\perp, \mathcal{F}, p, n+m)$ is derivable by DS-Res from Φ, then $\Pr[\Phi] \geq p$, and the strategy $\mathcal{F}$ is a certificate witnessing the lower bound p of $\Pr[\Phi]$.

Soundness of DS-Res

Theorem 7 (Soundness of DS-Res)

Given a DSSAT formula Φ, if some proof line $(\perp, \mathcal{F}, p, n+m)$ is derivable by DS-Res from Φ, then $\Pr[\Phi] \geq p$, and the strategy $\mathcal{F}$ is a certificate witnessing the lower bound p of $\Pr[\Phi]$.

Theorem 7 is a corollary that follows directly from Lemma 8, which formally states the previously mentioned semantics of proof lines [15] and can be proved by structural induction over the applications of DS-Res rules.

Lemma 8 (Semantics of DS-Res Proof Lines)

Given a DSSAT formula Φ and a proof line $(C, \mathcal{F}, p, i)$ derived by DS-Res from Φ, let α be an assignment to the variables $\{v_{i+1}, \ldots, v_{n+m}\}$ falsifying C, and let $\Phi[\alpha]$ denotes the DSSAT formula obtained through constraining Φ by the assignment α. Then, we have $\Pr[\Phi[\alpha]|_{\mathcal{F}}] = p$.

[15] Intuition of deriving $(C, \mathcal{F}, p, i)$: $\Pr[\text{ satisfy } \phi \mid C \text{ is falsified, } \mathcal{F} \text{ is adopted }] = p$.

Completeness of DS-Res

Theorem 9 (Completeness of DS-Res)

Given a DSSAT formula Φ with $\Pr[\Phi] = p$, then some proof line $(\bot, \mathcal{F}, p, n+m)$ is derivable by DS-Res from Φ.

Completeness of DS-Res

Theorem 9 (Completeness of DS-Res)

Given a DSSAT formula Φ with $\Pr[\Phi] = p$, then some proof line $(\bot, \mathcal{F}, p, n+m)$ is derivable by DS-Res from Φ.

Theorem 9 is a corollary that follows directly from Lemma 10, which intuitively states that every *possible* strategy is derivable in each iteration, and can be proved by structural induction over the applications of DS-Res rules.

Lemma 10

Given a DSSAT formula Φ, for each index $i \in \{0, \dots, m+n\}$ and for each assignment α to the variables $\{v_{i+1}, \dots . v_{n+m}\}$, let $\mathcal{G}$ be a complete strategy [a] for $\Phi[\alpha]$. Then, some proof line $(C, \mathcal{F}, p, i)$ is derivable by DS-Res from Φ such that

1. *α falsifies C, and*
2. *$\mathcal{G}$ is consistent with $\mathcal{F}$.*

[a] A complete strategy consists of functions that only outputs $\{\top, \bot\}$, i.e., has no $*$-output.

Completeness of DS-Res

- The proof for completeness of DS-Res is more subtle and less intuitive than that for soundness.

Completeness of DS-Res

- The proof for completeness of DS-Res is more subtle and less intuitive than that for soundness.
- Ideas of showing some $(\perp, \mathcal{F}, \Pr[\Phi], n+m)$ is derivable by DS-Res from the DSSAT formula Φ:

Completeness of DS-Res

- The proof for completeness of DS-Res is more subtle and less intuitive than that for soundness.
- Ideas of showing some $(\bot, \mathcal{F}, \Pr[\Phi], n+m)$ is derivable by DS-Res from the DSSAT formula Φ:
    1. $(\bot)$ is derivable: DS-Res resolves on both ϕ and $\psi \equiv \neg\phi$, and $\phi \wedge \psi \equiv \bot$;

Completeness of DS-Res

- The proof for completeness of DS-Res is more subtle and less intuitive than that for soundness.
- Ideas of showing some $(\bot, \mathcal{F}, \Pr[\Phi], n+m)$ is derivable by DS-Res from the DSSAT formula Φ:
  1. $(\bot)$ is derivable: DS-Res resolves on both ϕ and $\psi \equiv \neg\phi$, and $\phi \wedge \psi \equiv \bot$;
  2. all *possible* strategy $\mathcal{F}$ is derivable: weakened-resolution [16] and building strategies into proof lines are crucial to ensure all possible strategies can be derived.

[16] Weakening is also essential for the completeness of some DQBF resolution schemes [BBM21; BPS21].

Polynomial-time Verifiability of DS-Res

- Given a DS-Res proof, each derivation can be polynomial-time (w.r.t. proof size) verified.[17] To achieve polynomial-time verifiability, the *data-structures* for strategies is critical.

[17] The proof size itself may be a doubly exponential blow up w.r.t. the input DSSAT formula.

Polynomial-time Verifiability of `DS-Res`

- Given a `DS-Res` proof, each derivation can be polynomial-time (w.r.t. proof size) verified.[17] To achieve polynomial-time verifiability, the *data-structures* for strategies is critical.
- Recall that a function $f_e \in \mathcal{F}$ can output $\{*, \top, \bot\}$. According to [BPS21], f_e can be represented by a pair of Boolean functions $(f_e^\top, f_e^\bot)$, defined as follows: for each truth-assignment α to D_e,

$$f_e^\top(\alpha) = \begin{cases} \top, & \text{if } f_e(\alpha) = \top, \\ \bot, & \text{otherwise,} \end{cases} \qquad f_e^\bot(\alpha) = \begin{cases} \top, & \text{if } f_e(\alpha) = \bot, \\ \bot, & \text{otherwise.} \end{cases}$$

[17] The proof size itself may be a doubly exponential blow up w.r.t. the input DSSAT formula.

Polynomial-time Verifiability of DS-Res

- Given a DS-Res proof, each derivation can be polynomial-time (w.r.t. proof size) verified.[17] To achieve polynomial-time verifiability, the *data-structures* for strategies is critical.

- Recall that a function $f_e \in \mathcal{F}$ can output $\{*, \top, \bot\}$. According to [BPS21], f_e can be represented by a pair of Boolean functions $(f_e^\top, f_e^\bot)$, defined as follows: for each truth-assignment α to D_e,

$$f_e^\top(\alpha) = \begin{cases} \top, & \text{if } f_e(\alpha) = \top, \\ \bot, & \text{otherwise}, \end{cases} \qquad f_e^\bot(\alpha) = \begin{cases} \top, & \text{if } f_e(\alpha) = \bot, \\ \bot, & \text{otherwise}. \end{cases}$$

- If all Boolean functions $(f_e^\top, f_e^\bot)$ are represented by ordered binary decision diagrams (OBDD [DM02]), then the strategy operations: if-then-else, composition, and consistency check can be done in polynomial time. [BPS21]

[17] The proof size itself may be a doubly exponential blow up w.r.t. the input DSSAT formula.

Outline

Outline

Local Pruning in DS-Res.

- If we are only concerned with proof lines that lead to terminal proof lines with maximum satisfying probability, then some intermediate proof lines may be redundant and can be removed from the current proof.

Local Pruning in DS-Res.

- If we are only concerned with proof lines that lead to terminal proof lines with maximum satisfying probability, then some intermediate proof lines may be redundant and can be removed from the current proof.
- Example: $\Phi = \reflectbox{R}^{0.3} r_1, \reflectbox{R}^{0.6} r_2, \reflectbox{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2, r_1 has index 4,5, respectively.

$$\begin{array}{c} \eta_1 = (\{r_1, r_2\}, \{f_{e_1} = \bot, f_{e_2} = *\}, 1, 3) \\ \Big\downarrow \text{dominate} \\ \eta_2 = (\{r_1, r_2\}, \{g_{e_1} = \top, g_{e_2} = r_3\}, 0.2, 3) \end{array}$$

Local Pruning in DS-Res.

- If we are only concerned with proof lines that lead to terminal proof lines with maximum satisfying probability, then some intermediate proof lines may be redundant and can be removed from the current proof.
- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2, r_1 has index 4,5, respectively.

$$\eta_1 = (\{r_1, r_2\}, \{f_{e_1} = \bot, \color{red}{f_{e_2} = *}\}, \color{red}{1}, 3)$$

$$\Big\downarrow \text{dominate}$$

$$\eta_2 = (\{r_1, r_2\}, \{g_{e_1} = \top, \color{red}{g_{e_2} = r_3}\}, \color{red}{0.2}, 3)$$

  - Only r_1, r_2 are unresolved pivots.

Local Pruning in DS-Res.

- If we are only concerned with proof lines that lead to terminal proof lines with maximum satisfying probability, then some intermediate proof lines may be redundant and can be removed from the current proof.
- Example: $\Phi = \text{Я}^{0.3} r_1, \text{Я}^{0.6} r_2, \text{Я}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2, r_1 has index 4,5, respectively.

$$\eta_1 = (\{r_1, r_2\}, \{f_{e_1} = \bot, f_{e_2} = *\}, 1, 3)$$

$$\Big\downarrow \text{dominate}$$

$$\eta_2 = (\{r_1, r_2\}, \{g_{e_1} = \top, g_{e_2} = r_3\}, 0.2, 3)$$

  - Only r_1, r_2 are unresolved pivots.
  - Since $r_1, r_2 \in D_{e_1}$ and $f_{e_2} = * \in \eta_1$ is more general than $g_{e_2} = r_3 \in \eta_2$, any η resolvable with η_2 is also resolvable with η_1.

Local Pruning in DS-Res.

- If we are only concerned with proof lines that lead to terminal proof lines with maximum satisfying probability, then some intermediate proof lines may be redundant and can be removed from the current proof.
- Example: $\Phi = \rotatebox[origin=c]{180}{R}^{0.3} r_1, \rotatebox[origin=c]{180}{R}^{0.6} r_2, \rotatebox[origin=c]{180}{R}^{0.8} r_3, \exists e_1(\{r_1, r_2\}), \exists e_2(\{r_1, r_3\}).\phi$ and r_2, r_1 has index 4,5, respectively.

$$\eta_1 = (\{r_1, r_2\}, \{f_{e_1} = \bot, f_{e_2} = *\}, 1, 3)$$

$$\Big\downarrow \text{dominate}$$

$$\eta_2 = (\{r_1, r_2\}, \{g_{e_1} = \top, g_{e_2} = r_3\}, 0.2, 3)$$

  - Only r_1, r_2 are unresolved pivots.
  - Since $r_1, r_2 \in D_{e_1}$ and $f_{e_2} = * \in \eta_1$ is more general than $g_{e_2} = r_3 \in \eta_2$, any η resolvable with η_2 is also resolvable with η_1.
  - Resolving with η_1 yields proof lines with higher satisfying probability. Therefore, η_2 is dominated by η_1 and is redundant.

Local Pruning in DS-Res.

Definition 11 (Dominance)

Let Φ be a DSSAT formula. Given two proof lines $\eta_1 = (C_1, \mathcal{F}, p_1, i)$, $\eta_2 = (C_2, \mathcal{G}, p_2, i)$, we say that η_1 *dominates* η_2 (written $\eta_1 \succeq \eta_2$) if

1. $C_1 \subseteq C_2$,
2. $p_1 \geq p_2$, and
3. for each existential variable e, either
    1. $f_e \in \mathcal{F}$ is more general than $g_e \in \mathcal{G}$, or
    2. e depends on all randomized variables that are not yet resolved.

Local Pruning in DS-Res.

Definition 11 (Dominance)

Let Φ be a DSSAT formula. Given two proof lines $\eta_1 = (C_1, \mathcal{F}, p_1, i)$, $\eta_2 = (C_2, \mathcal{G}, p_2, i)$, we say that η_1 *dominates* η_2 (written $\eta_1 \succeq \eta_2$) if

1. $C_1 \subseteq C_2$,
2. $p_1 \geq p_2$, and
3. for each existential variable e, either
    1. $f_e \in \mathcal{F}$ is more general than $g_e \in \mathcal{G}$, or
    2. e depends on all randomized variables that are not yet resolved.

Proposition 1

Given a DSSAT formula Φ, soundness and completeness still hold for Φ if we iteratively remove dominated $(i-1)$-proof lines until no $(i-1)$-proof line is dominated before deriving the i-proof lines, for each index $i \in \{1, \ldots, m+n\}$.

Outline

SSAT Resolution

- Given
  - an SSAT formula $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\text{Ⅎ}^{p_i}, \exists\}$ and
  - a CNF $\psi \equiv \neg\phi$.

SSAT Resolution

- Given
  - an SSAT formula $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\text{Ǝ}^{p_i}, \exists\}$ and
  - a CNF $\psi \equiv \neg\phi$.
- The SSAT resolution `S-resolution` [TF12]:

SSAT Resolution

- Given
  - an SSAT formula $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\reflectbox{R}^{p_i}, \exists\}$ and
  - a CNF $\psi \equiv \neg\phi$.
- The SSAT resolution `S-resolution` [TF12]:
  - proof lines syntax: (C, p) where C is a clause and p is a probability value;

SSAT Resolution

- Given
  - an SSAT formula $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\reflectbox{R}^{p_i}, \exists\}$ and
  - a CNF $\psi \equiv \neg\phi$.
- The SSAT resolution `S-resolution` [TF12]:
  - proof lines syntax: (C, p) where C is a clause and p is a probability value;
  - proof line semantics: $\Pr[\text{ satisfy } \phi \mid C \text{ is falsified }] = p$.

SSAT Resolution

- Given
  - an SSAT formula $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, each $Q_i \in \{\reflectbox{R}^{p_i}, \exists\}$ and
  - a CNF $\psi \equiv \neg\phi$.
- The SSAT resolution `S-resolution` [TF12]:
  - proof lines syntax: (C, p) where C is a clause and p is a probability value;
  - proof line semantics: $\Pr[$ satisfy $\phi \mid C$ is falsified $] = p$.
- Note that `S-resolution` do not need to maintain strategies in proof lines because the SSAT prefix is totally ordered and allows local decisions.

SSAT Resolution

`S-resolution` consists of the following three rules:

$$\text{R.1}\ \frac{C \in \phi}{(C, 0)} \qquad \text{R.2}\ \frac{C \in \psi}{(C, 1)}$$

$$\text{R.3}\ \frac{(C_0 \cup \{\neg v_i\}, p_0), (C_1 \cup \{v_i\}, p_1) : \text{pivot } v_i \text{ has larger variable index than every literal in } C_0 \cup C_1}{(C_0 \cup C_1, p^{0,1})}$$

$$\text{where } p^{0,1} = \begin{cases} \max(p_0, p_1)\ ^{18} & \text{if } v_i \text{ is } \exists\text{-quantified,} \\ p \cdot p_0 + (1-p) \cdot p_1 & \text{if } v_i \text{ is } \text{Я}^p\text{-quantified.} \end{cases}$$

[18] The $\max$-operation is essentially adopting a locally optimal decision in `S-resolution`.

P-simulation of S-resolution

Lemma 12

Given an SSAT formula $\Phi = Q_1 v_1, \ldots, Q_n v_n.\phi$, let P be an S-resolution *proof of Φ. For each* S-resolution *proof line $(C, p) \in P$, we can derive some* DS-Res *proof line $(C', \mathcal{F}, p, i)$ in polynomial time such that*

1. $C' \subseteq C$, *and*
2. $i = \begin{cases} 0 & \text{if } (C, p) \text{ is derived from } \mathrm{R}.1 \text{ or } \mathrm{R}.2 \\ n + 1 - j & \text{if } (C, p) \text{ is derived from } \mathrm{R}.3 \text{ over the pivot variable } v_j. \end{cases}$

Corollary 13 (P-simulation of S-resolution)

Given an S-resolution *proof P, a* DS-Res *proof P' can be derived in polynomial time.*

Note that the DS-Res proof P' derives not only the maximum satisfying probability, but also the witnessing strategy. Hence, the simulation may also serve as a polynomial-time strategy extraction for S-resolution.

Outline

Conclusion and Future Work

- Conclusion: We developed the first sound and complete resolution proof system `DS-Res` for DSSAT.

Conclusion and Future Work

- Conclusion: We developed the first sound and complete resolution proof system `DS-Res` for DSSAT.
  - `DS-Res` has favorable properties:
    1. generates polynomial-time verifiable proofs;
    2. allows local pruning that reduces proof sizes;
    3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.

Conclusion and Future Work

- Conclusion: We developed the first sound and complete resolution proof system `DS-Res` for DSSAT.
  - `DS-Res` has favorable properties:
    1. generates polynomial-time verifiable proofs;
    2. allows local pruning that reduces proof sizes;
    3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.
  - The potential benefits of `DS-Res`:
    1. provides theoretical insights for future DSSAT solver development;
    2. serves as the underlying proof system for proof logging in future trustworthy DSSAT solvers.

Conclusion and Future Work

- Conclusion: We developed the first sound and complete resolution proof system `DS-Res` for DSSAT.
  - `DS-Res` has favorable properties:
    1. generates polynomial-time verifiable proofs;
    2. allows local pruning that reduces proof sizes;
    3. p-simulates a known SSAT resolution scheme given the special case of SSAT instances.
  - The potential benefits of `DS-Res`:
    1. provides theoretical insights for future DSSAT solver development;
    2. serves as the underlying proof system for proof logging in future trustworthy DSSAT solvers.
- Future work: develop a DSSAT decision procedure exploiting the DS-Res calculus.

Acknowledgement

- Thanks to Prof. Jie-Hong Roland Jiang for guiding me all the way throughout my research journey, introducing me to the wonderful world of logic and formal methods, and supporting my Ph.D. application.
- Thanks to Prof. Tony Tan for teaching me mathematical logic and complexity, which fostered my interest in mathematics and theoretical computer science and provided me with solid training in rigorous reasoning and writing.
- Thanks to Prof. Fang Yu for teaching me a lecture during Flolac 2019 and joining my thesis committee even though he does not know me.
- Thanks to Che Cheng for his significant contribution to this work and our discussion on almost every research topic of mine.
- Thanks to all ALCom colleagues, my dearest family and friends, and God for supporting and sustaining me throughout this journey.

References I

[BBM21] Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. "Building Strategies into QBF Proofs". In: *Journal of Automated Reasoning* 65 (2021), pp. 125–154.

[BPS21] Joshua Blinkhorn, Tomas Peitl, and Friedrich Slivovsky. "Davis and Putnam Meet Henkin: Solving DQBF with Resolution". In: *International Conference on Theory and Applications of Satisfiability Testing*. 2021, pp. 30–46.

[CJ23] Che Cheng and Jie-Hong R. Jiang. "Lifting (D)QBF Preprocessing and Solving Techniques to (D)SSAT". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023.

[DM02] Adnan Darwiche and Pierre Marquis. "A Knowledge Compilation Map". In: *Journal of Artificial Intelligence Research* 17 (2002), pp. 229–264.

[FHR22] Johannes K Fichte, Markus Hecher, and Valentin Roland. "Proofs for propositional model counting". In: *25th International Conference on Theory and Applications of Satisfiability Testing (SAT 2022)*. 2022.

References II

[GBM21] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel. "Justicia: A Stochastic SAT Approach to Formally Verify Fairness". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.9 (May 2021), pp. 7554–7563.

[Heu21] Marijn JH Heule. "Proofs of unsatisfiability". In: *Handbook of Satisfiability*. Amsterdam, Netherlands: IOS Press, 2021, pp. 635–668.

[HJ22] Cheng-Han Hsieh and Jie-Hong R. Jiang. "Encoding Probabilistic Graphical Models into Stochastic Boolean Satisfiability". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 2022, pp. 1834–1842.

[HKM16] Marijn JH Heule, Oliver Kullmann, and Victor W Marek. "Solving and verifying the boolean pythagorean triples problem via cube-and-conquer". In: *International Conference on Theory and Applications of Satisfiability Testing*. 2016, pp. 228–245.

References III

[JHB08] Jan Johannsen, Jan Hoffmann, and Samuel R Buss. "Resolution Trees with Lemmas: Resolution Refinements that Characterize DLL Algorithms with Clause Learning". In: *Logical Methods in Computer Science* 4 (2008).

[LJ18] Nian-Ze Lee and Jie-Hong R. Jiang. "Towards Formal Evaluation and Verification of Probabilistic Design". In: *IEEE Transactions on Computers* 67.8 (2018), pp. 1202–1216.

[LJ21] Nian-Ze Lee and Jie-Hong R. Jiang. "Dependency Stochastic Boolean Satisfiability: A Logical Formalism for NEXPTIME Decision Problems with Uncertainty". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2021, pp. 3877–3885.

[MB05] Stephen M. Majercik and Byron Boots. "DC-SSAT: A Divide-and-Conquer Approach to Solving Stochastic Satisfiability Problems Efficiently". In: *Proceedings of National Conference on Artificial Intelligence*. 2005, pp. 416–422.

References IV

[ML03] Stephen M Majercik and Michael L Littman. "Contingent Planning under Uncertainty via Stochastic Satisfiability". In: *Artificial Intelligence* 147.1-2 (2003), pp. 119–162.

[ML98] Stephen M. Majercik and Michael L. Littman. "MAXPLAN: A New Approach to Probabilistic Planning". In: *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems*. 1998, pp. 86–93.

[MS00] João P Marques-Silva and Karem A Sakallah. "Boolean satisfiability in electronic design automation". In: *Proceedings of the 37th Annual Design Automation Conference*. 2000, pp. 675–680.

[Nil82] Nils J Nilsson. *Principles of Artificial Intelligence*. Springer Science & Business Media, 1982.

[Pap85] Christos H. Papadimitriou. "Games against Nature". In: *Journal of Computer and System Sciences* 31.2 (1985), pp. 288–301.

References V

[SP20] Ricardo Salmon and Pascal Poupart. “On the Relationship Between Satisfiability and Markov Decision Processes”. In: *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. Vol. 115. 2020, pp. 1105–1115.

[SW18] Christoph Scholl and Ralf Wimmer. “Dependency Quantified Boolean Formulas: An Overview of Solution Methods and Applications”. In: *Proceedings of International Conference on Theory and Applications of Satisfiability Testing (SAT)*. 2018, pp. 3–16.

[TF12] Tino Teige and Martin Fränzle. “Generalized Craig Interpolation for Stochastic Boolean Satisfiability Problems with Applications to Probabilistic State Reachability and Region Stability”. In: *Log. Methods Comput. Sci.* 8.2 (2012).

[VWM15] Yakir Vizel, Georg Weissenbacher, and Sharad Malik. “Boolean Satisfiability Solvers and Their Applications in Model Checking”. In: *Proceedings of the IEEE* 103.11 (2015), pp. 2021–2035.