Proof-Stitch: Proof Concatenation for Divide-and-Conquer SAT solvers

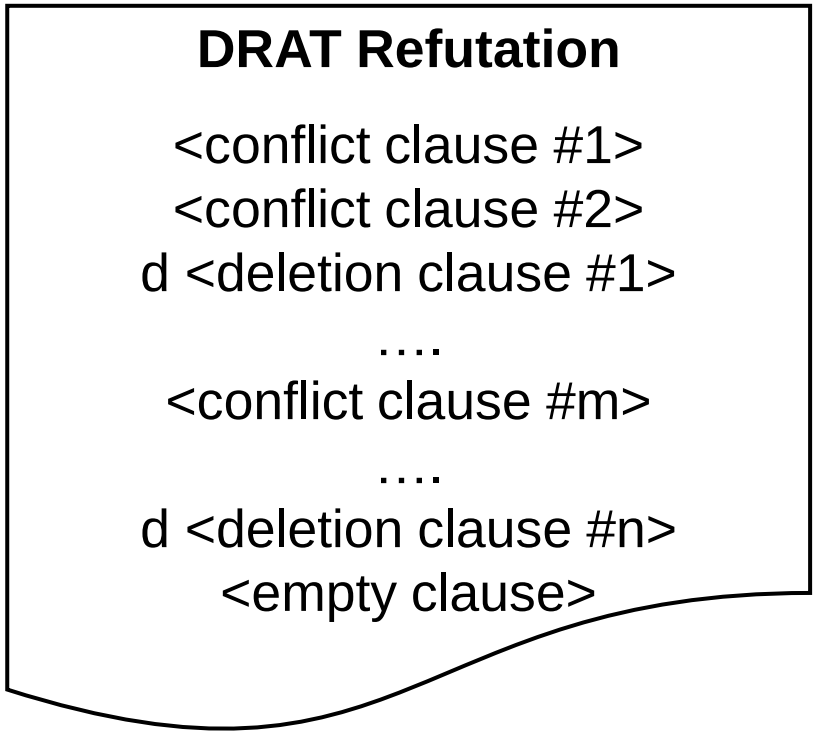
Abhishek Nair, Saranyu Chattopadhyay, Haoze Wu,
Alex Ozdemir and Clark Barrett

Stanford University

Background

Propositional Refutations

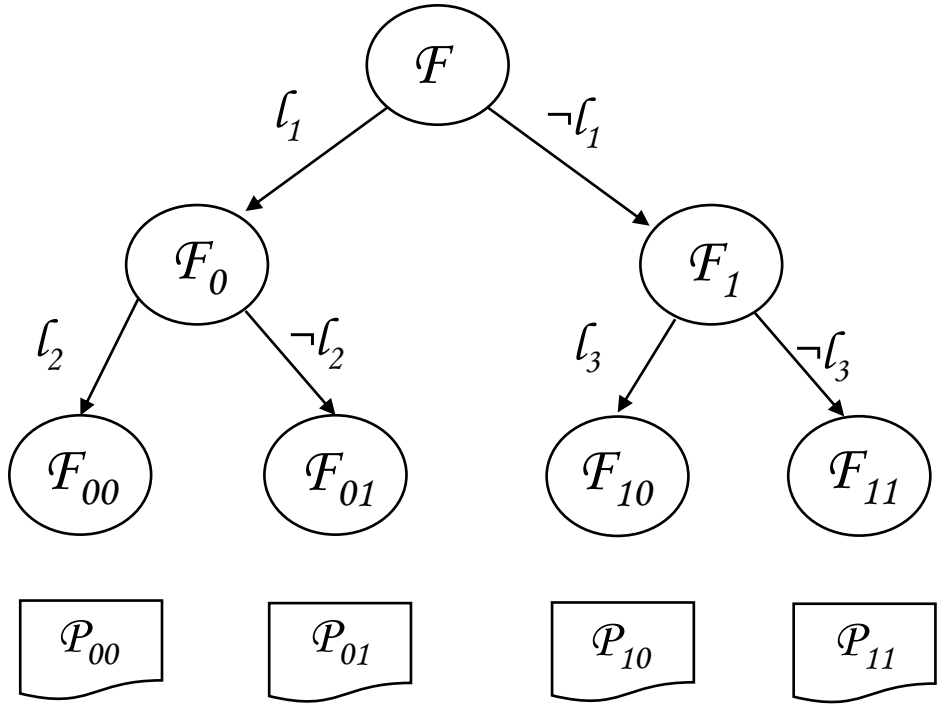
- Prove unsatisfiability of propositional formulae
- Popular format: DRAT



Background

Divide and Conquer SAT solvers

- Partition search space of propositional formula
- Solve sub-problems using CDCL solvers



Contributions

Proof-Stitch Algorithm

Merges sub-problem refutations into single refutation

Validity of output refutation theoretically proven

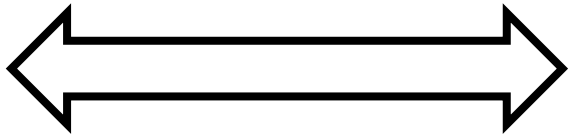
Proof-Stitch Tool

Open-source tool leveraging parallel processing

In-built optimization heuristic to improve runtime

Unsatisfiability using Divide-and-Conquer

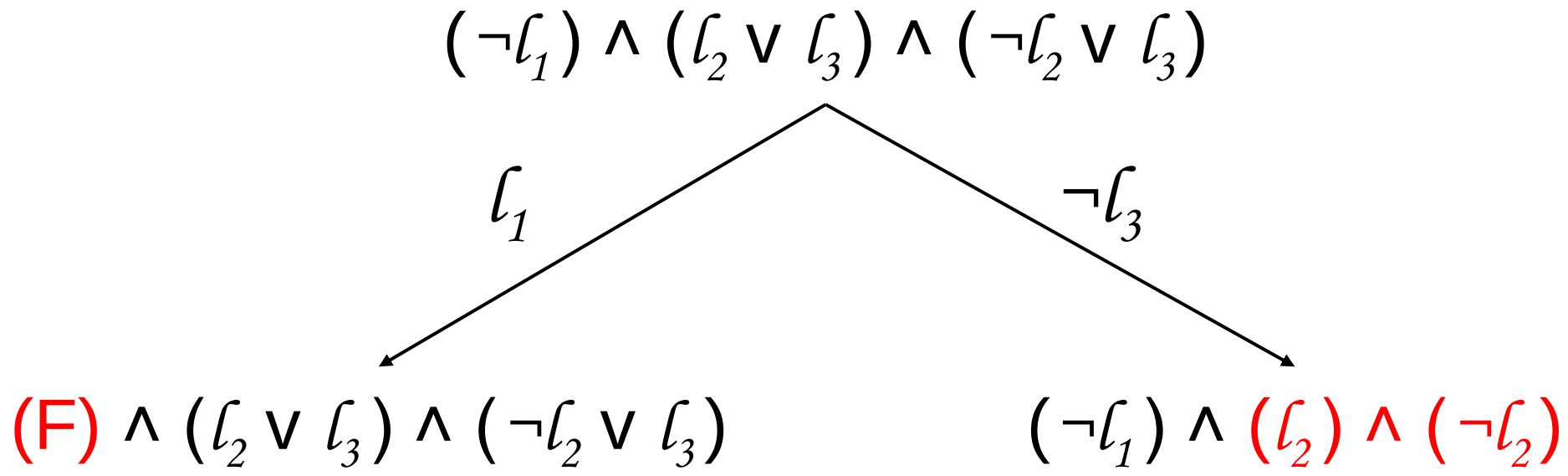
Original problem UNSAT



All sub-problems UNSAT*

*If partitioning strategy is error-free

Example: Erroneous Partition



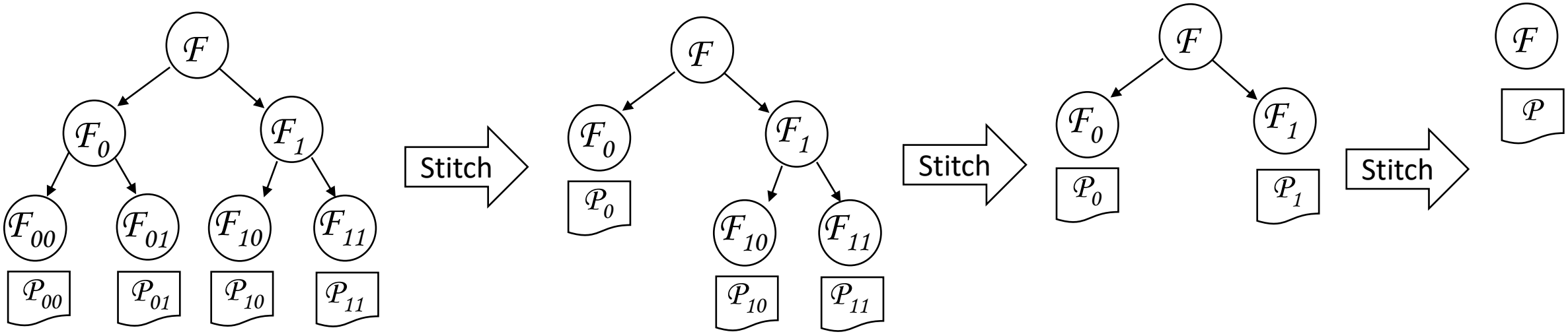
Original instance is satisfiable!

Proof-Stitch

Combine sub-problem refutations into single refutation recursively

Stitching Operation

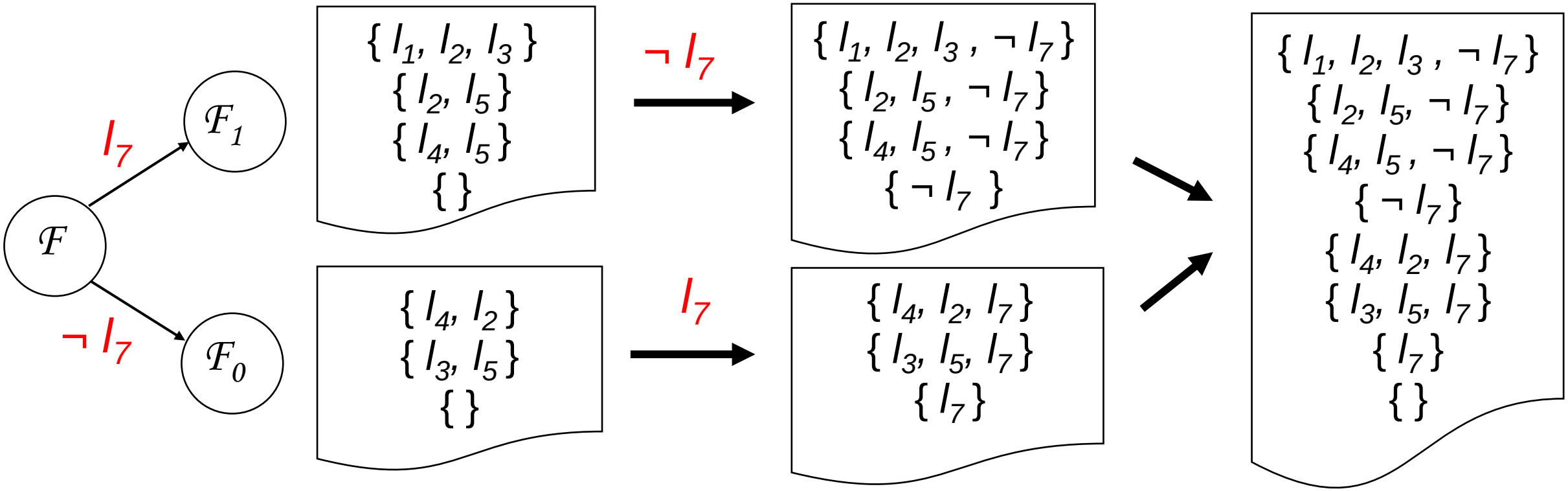
- Merge two sub-problem refutations which share a parent



Stitching Operation

Two step process:

- Clauses appended with negation of decision literal
- Refutations concatenated; empty clause added



Validity of generated refutation

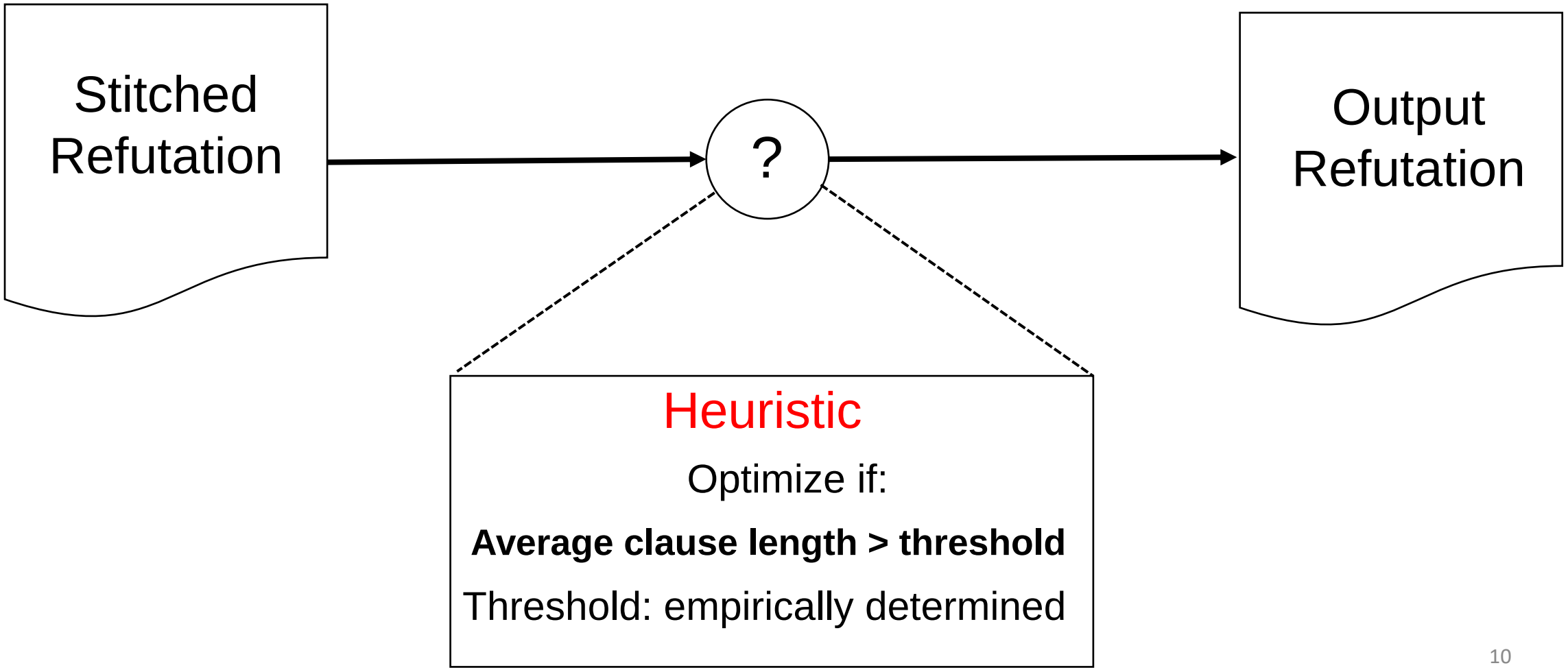
- Each stitching operation creates a valid refutation*

*if the input refutations are “preserving”

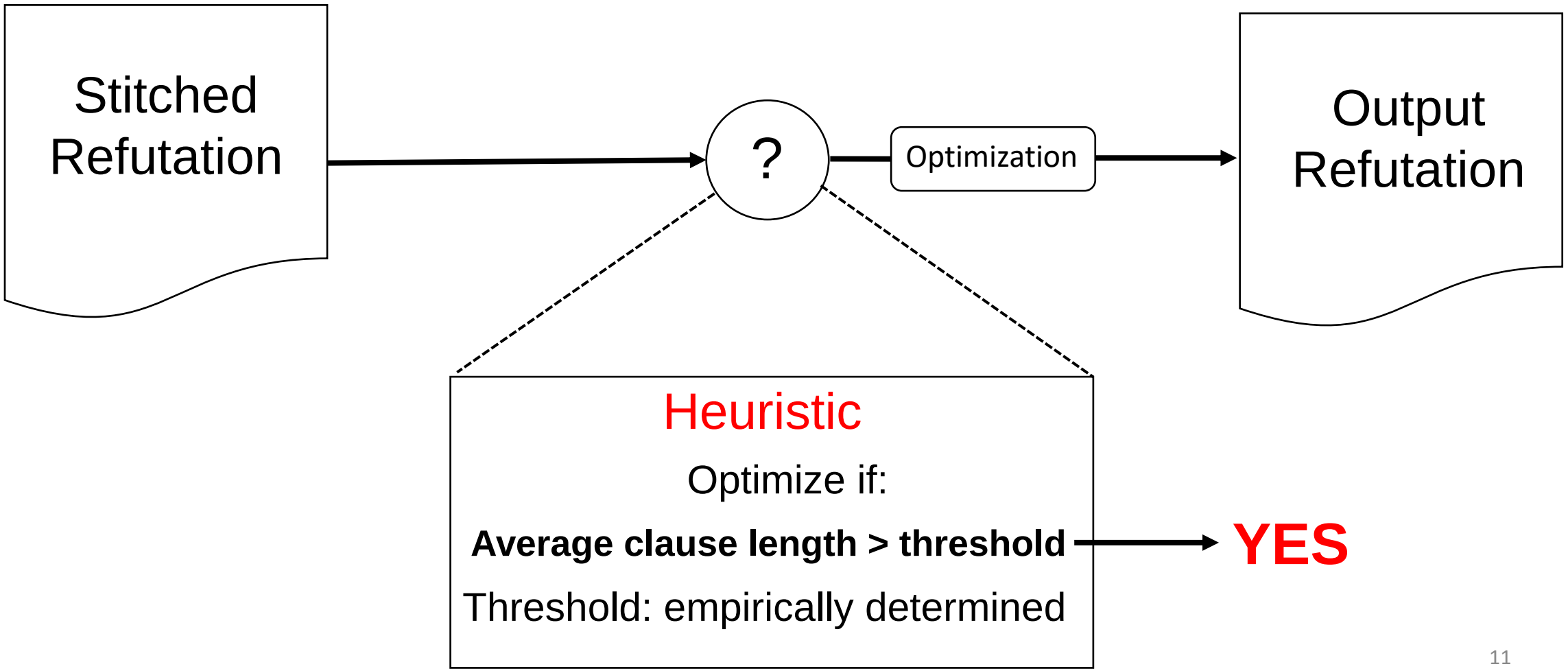
Preserving refutation:

- No clauses from formula are deleted

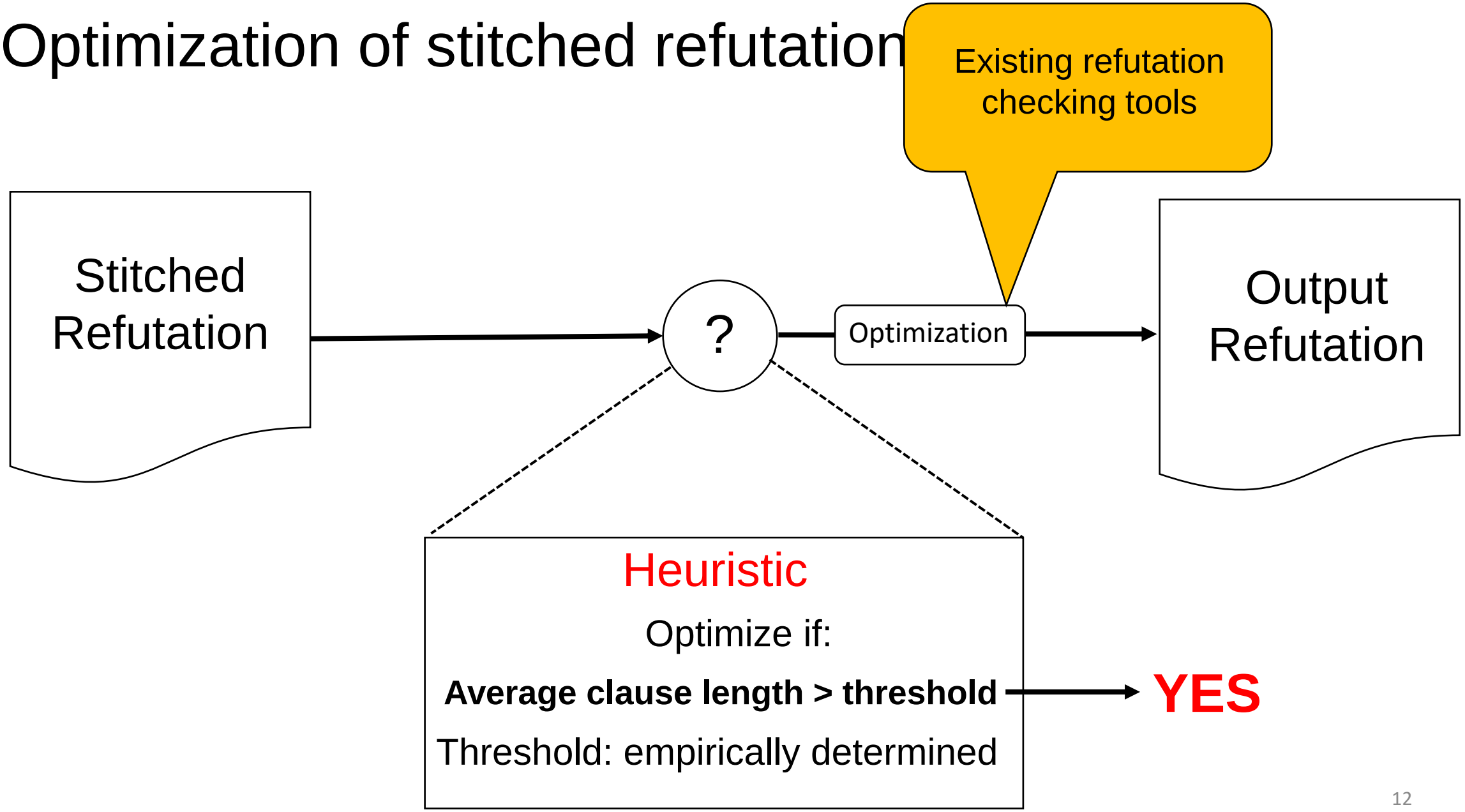
Optimization of stitched refutations



Optimization of stitched refutations



Optimization of stitched refutation



Proof-Stitch Tool Interface

Inputs:

- Original formula in DIMACS format
- Sub-problem refutations in DRAT format
- Refutations file names contain decision literals

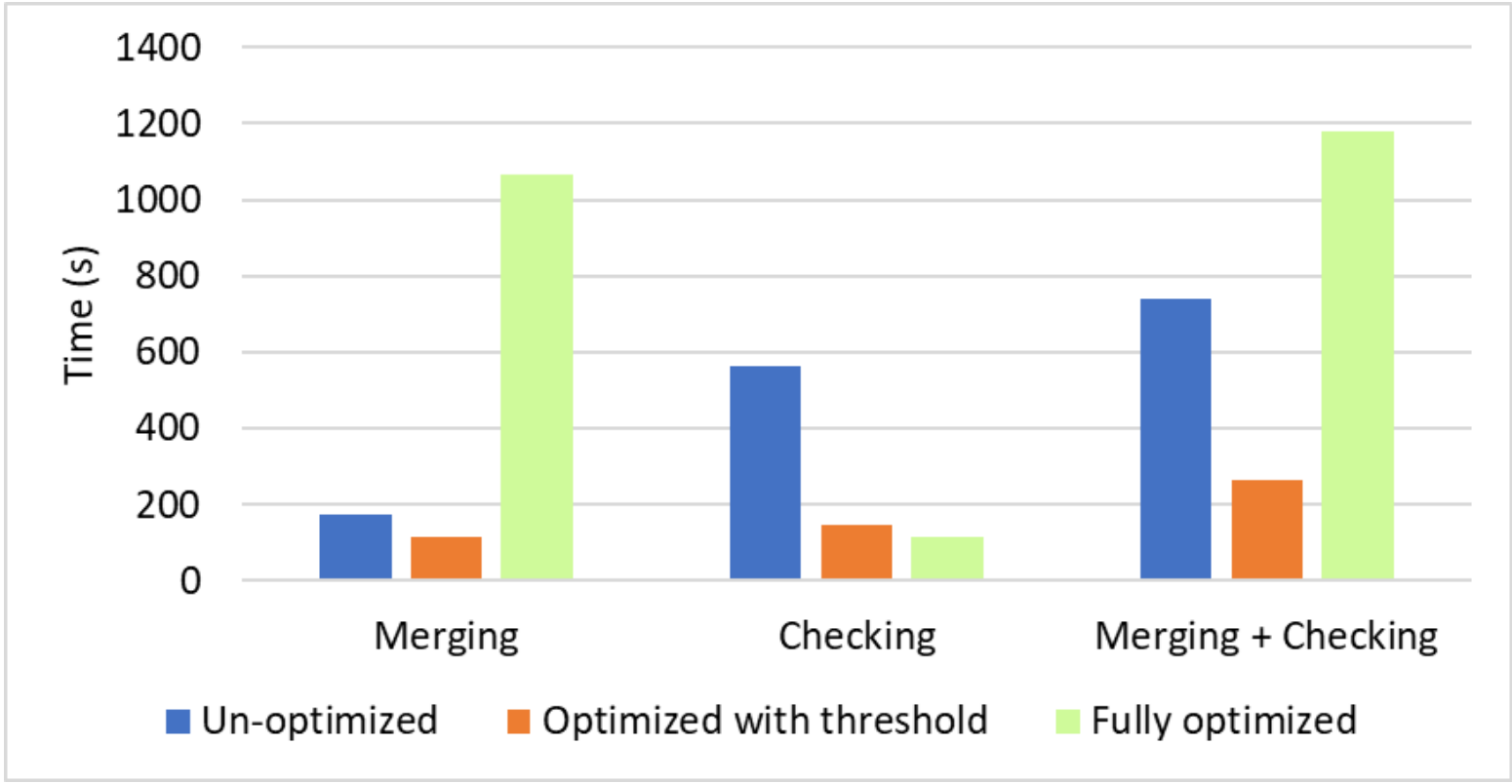
Outputs:

- Stitched refutation in DRAT format

Optimization levels:

- No optimization
- Heuristic-based: Optimize based on threshold value
- Full optimization: All refutations optimized

Results – Benefit of heuristic-based optimization



Merging + Checking time is minimum when heuristic is used for optimization

Results – Benefit of heuristic-based optimization

Final refutations are compressed (upto 20x improvement)

Benchmark	Unoptimized	Optimized
	Refutation Size (MB)	Refutation Size (MB)
p01_lb_05	1700	184
ktf_TF	385	77
satch2way12	1600	655
pb_300_10	536	27
mp1-Nb6T06	586	222
E02F17	1500	294

Contributions

Proof-Stitch Algorithm

Merges sub-problem refutations into single refutation

Validity of output refutation theoretically proven

Proof-Stitch Tool

Open-source tool leveraging parallel processing

In-built optimization heuristic to improve runtime