CS257: Introduction to Automated Reasoning

Theory Combination

Stanford University

CENTAUR

Need for Combining Theories and Solvers

Recall: Many applications give rise to formulas like:

$$a = b + 2 \wedge A = \mathrm{write}(B, a+1, 4) \wedge$$
$$(\mathrm{read}(A, b+3) = 2 \vee f(a-1) \neq f(b+1))$$

Solving that formula requires reasoning over

- the theory of linear arithmetic (T_{LA})
- the theory of arrays (T_{A})
- the theory of uninterpreted functions (T_{UF})

Question: Given solvers for each theory, can we combine them modularly into one for $T_{\mathrm{LA}} \cup T_{\mathrm{A}} \cup T_{\mathrm{UF}}$?

Under certain conditions, we can do it with the Nelson-Oppen combination method

Reminder: First-Order Logic Symbols

The syntax of many-sorted FOL is defined with respect to a **signature**, $\Sigma := \{\Sigma^S, \Sigma^F\}$, where:

- Σ^S is a set of **sorts**: e.g., $\mathsf{Real}, \mathsf{Int}, \mathsf{Set}$
- Σ^F is a set of **function symbols**: e.g., $\in, +, +_{[2]}, <, \between$

In addition to the function symbols, the alphabet of FOL also contains logical symbols:

- Parentheses: "(", ")"
- Propositional connectives: $\rightarrow$, $\neg$
- Variables: v_1, v_2, ...
- Quantifiers: $\forall$
- Equality symbol: for each sort σ in Σ^S, there may be an optional symbol $=_\sigma$.

Reminder: First-Order Logic Signatures

The syntax of many-sorted FOL is defined with respect to a **signature**, $\Sigma := \{\Sigma^S, \Sigma^F\}$, where:

- Σ^S is a set of **sorts**: e.g., $\mathsf{Real}, \mathsf{Int}, \mathsf{Set}$
- Σ^F is a set of **function symbols**: e.g., $\in, +, +_{[2]}, <, \between$

For each variable v, we associate a sort $sort(v) \in \Sigma^S$.

For each function symbol $f \in \Sigma^F$ we associate an **arity** n, which is a natural number denoting the number of arguments f takes, and an $n+1$-tuple of sorts: $sort(f) = \langle \sigma_1, \ldots, \sigma_n, \sigma_{n+1} \rangle$. We say f **returns** σ_{n+1}.

Example: The function symbol $+$ has arity 2, and $sort(+) = \langle Real, Real, Real \rangle$ in the intended translation.

Reminder: First-Order Logic Signatures

We assume that Σ^S always includes a distinguished sort Bool and that Σ^F contains distinguished symbols $\{\top, \bot\}$.

We assume $sort(\bot) = sort(\top) = \langle \text{Bool} \rangle$

There are two special kinds of functions, **constant symbols** and **predicate symbols**:

- **Constant symbols** are 0-arity function symbols: e.g., $\bot$, $\top$, π, John, 0
- A **predicate symbol** is a function symbol that returns Bool
  - Each equality symbol $=_\sigma$ is a predicate symbol with arity 2 and $sort(=_\sigma) = \langle \sigma, \sigma, \text{Bool} \rangle$.

Example: $sort(\in) = \langle Set, Set, \text{Bool} \rangle$ in the intended translation.

To specify which **first-order language** we have before us, we need to:

- say whether the equality symbol is present;
- define the signature.

Reminder: First-Order Logic Semantics

Formally, the truth of a Σ-formula is determined by an **interpretation** I of Σ consisting of the following:

1. For each sort $\sigma \in \Sigma^S$, a nonempty set called the **domain** of σ, written $dom(\sigma)$
   - We always assume $dom(\text{Bool}) = \{\mathtt{T}, \mathtt{F}\}$
2. A mapping from each n-ary function symbol f in Σ^F of sort $sort(f) = \langle \sigma_1, \ldots, \sigma_n, \sigma_{n+1} \rangle$ to f^I, an n-ary function from $dom(\sigma_1) \times \cdots \times dom(\sigma_n)$ to $dom(\sigma_{n+1})$
   - We always assume $\bot^I = \mathtt{F}$, $\top^I = \mathtt{T}$, and $=^I_\sigma ab = \mathtt{T}$ iff $a = b$
3. A mapping from each variable v of sort σ to its interpretation v^I, an element of $dom(\sigma)$

(1) and (2) without (3) is called a **structure** or a **model**.

First-order theories and their combination

A **theory** T is a pair (Σ, S), where:

- Σ is a signature, which we recall from Lecture 4 consists of a set Σ^S of **sorts** and a set Σ^F of function symbols.
- S is a class (in the sense of set theory) of Σ-structures.

We limit interpretations of Σ-formulas to those that have their structures in S.

Theory combination: Let $T_1 = (\Sigma_1, S_1)$ and $T_2 = (\Sigma_2, S_2)$ be two theories. The combination of T_1 and T_2 is the theory $T_1 \oplus T_2 = (\Sigma, S)$ where $\Sigma = \Sigma_1 \cup \Sigma_2$ and $S = \{\Sigma\text{-structures } I \mid I^{\Sigma_1,\varnothing} \in S_1 \text{ and } I^{\Sigma_2,\varnothing} \in S_2\}$.

Above, I is an interpretation, and $I^{\Sigma,U}$ denotes the interpretation obtained by interpreting symbols in Σ and variables in U. Structures do not interpret variables, so U is empty above.

Theory Combination: Preliminaries

First-order theories without the equality symbol are rarely considered. We will follow this convention.

Convex theory: A Σ-theory T is convex if for every conjunctive Σ-formula ϕ:

$(\phi \rightarrow \bigvee_{i=1}^{n} x_i = y_i)$ is T-valid for some finite $n > 1 \rightarrow$
$(\phi \rightarrow x_i = y_i)$ is T-valid for some $i \in 1, ..., n$,

where x_i, y_i, for $i \in 1, ..., n$, are some variables.

Theory Combination: Preliminaries

Example (convex): Linear real arithmetic is convex. A conjunction of linear arithmetic predicates defines a set of values which can be empty, a singleton, as in

$x \leq 3 \wedge x \geq 3 \rightarrow x = 3$

or infinitely large, and hence it implies an infinite disjunction. All three cases fit the definition of convexity.

Example (non-convex): Linear integer arithmetic is non-convex. For example, while

$x_1 = 1 \wedge x_2 = 2 \wedge 1 \leq x_3 \wedge x_3 \leq 2 \rightarrow (x_3 = 1 \vee x_3 = 2)$ holds, neither

$x_1 = 1 \wedge x_2 = 2 \wedge 1 \leq x_3 \wedge x_3 \leq 2 \rightarrow x_3 = 1$, nor

$x_1 = 1 \wedge x_2 = 2 \wedge 1 \leq x_3 \wedge x_3 \leq 2 \rightarrow x_3 = 2$ holds.

Many theories used in practice are nonconvex, which makes them computationally harder to combine with other theories due to case splits, which we'll see.

Nelson-Oppen: Step 1, Purification

Given decision procedures for the satisfiability of formulas in theories T_1 and T_2, we are interested in constructing a decision procedure for the satisfiability of $T_1 \oplus T_2$.

Given a conjunctive formula ϕ (i.e., a conjunction of literals) over the combined signature $\Sigma_1 \cup \Sigma_2$, the first step is to purify ϕ by constructing and equisatisfiable set of conjunctive formulas $\phi_1 \cup \phi_2$ such that each ϕ_i consists of only Σ_i-formulas.

Purification:

Given a conjunctive formula, ϕ:

1. Find a pure sub-term (i.e., a Σ_i-sub-term for some i), t.
2. Replace t with a fresh variable v, and add the term $v = t$ to the conjunctive formula.
3. Repeat steps 1 and 2 until all atomic formulas are pure.
4. Split the resulting conjunctive formula into two formulas $\phi_1 \cup \phi_2$, which are linked by a set of shared variables.

Motivating Example (Convex Case)

Consider the following set of literals over $T_{\mathrm{LRA}} \cup T_{\mathrm{UF}}$ (T_{LRA}, linear real arithmetic):

$$\begin{aligned} f(f(x) - f(y)) &= a \\ f(0) &> a + 2 \\ x &= y \end{aligned}$$

First step: purify literals so that each belongs to a single theory

$$f(f(x) - f(y)) = a \implies \begin{aligned} f(e_1) &= a \\ e_1 &= f(x) - f(y) \end{aligned} \implies \begin{aligned} f(e_1) &= a \\ e_1 &= e_2 - e_3 \\ e_2 &= f(x) \\ e_3 &= f(y) \end{aligned}$$

$$f(0) > a + 2 \implies f(e) > a + 2 \implies f(e) = e$$

Nelson-Oppen: Step 2, Exchange Interface Equalities

Formulas ϕ_1 and ϕ_2, which were produces through purification are linked by a set of shared variables. Let V = shared(ϕ_1, ϕ_2) be these shared variables.

Let E be an equivalence relation over V. The **arrangement** $A(V, E)$ of V induced by E is the formula:

$$A(V,E): \bigwedge_{u,v \in V. uEv} u = v \wedge \bigwedge_{u,v \in V. \neg uEv} u \neq v,$$

which asserts that variables related by E are equal and that variables unrelated by E are not equal. The original formula ϕ is $(T_1 \cup T_2)$-satisfiable iff there exists an equivalence relation E of V such that:

- $\phi_1 \wedge A(V, E)$ is T_1-satisfiable, and
- $\phi_2 \wedge A(V, E)$ is T_2-satisfiable

Exchanging interface equalities: Step 2 of the Nelson-Oppen procedure asks decision procedures P_1 and P_2 for theories T_1 and T_2, respectively, to propagate information to each other in the form of entailed equalities over shared variables.

Motivating Example (Convex Case)

Second step: exchange entailed interface equalities, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$. **Note:** We can view variables as being existentially quantified or as free constants, i.e., constant symbols not in the theory signature.

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3$ $\quad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\mathrm{UF}} \perp$
$L_2 \models_{\mathrm{LRA}} \perp$ Report unsatisfiable

Motivating Example (Non-convex Case)

Consider the following unsatisfiable set of literals over $T_{\text{LIA}} \cup T_{\text{UF}}$ (T_{LIA}, linear integer arithmetic):

$$
\begin{aligned}
1 \leq & \quad x & \leq 2 \\
f(1) & \quad = & a \\
f(2) & \quad = & f(1) + 3 \\
a & \quad = & b + 2
\end{aligned}
$$

First step: purify literals so that each belongs to a single theory

$$f(1) = a \quad \Longrightarrow \quad f(e_1) = a$$

Motivating Example (Non-convex Case)

Second step: exchange entailed interface equalities over shared constants $x, e_1, a, b, e_2, e_3, e_4$.

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

No more entailed equalities, but $L_1 \models_{\text{LIA}} x = e_1 \vee x = e_2$ Consider each case of $x = e_1 \vee x = e_2$ separately. **Note:** For convex theories, entailed clauses consisting of equality literals over shared constants are unit. For non-convex theories, case-splitting is necessary. Case 1) $x = e_1$ $L_2 \models_{\text{UF}} a = b$, which entails $\bot$ when sent to L_1

Motivating Example (Non-convex Case)

Second step: exchange entailed interface equalities over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_2$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_2$	

Case 2) $x = e_2$ $\quad$ $L_2 \models_{\mathrm{UF}} e_3 = b$, which entails $\perp$ when sent to L_1

Non-convex case: Disjunctions of equalities

A procedure for a non-convex theory T_i must be able to find disjunctions of equalities that are entailed by a Σ_i-formula ϕ_i. Disjunctions should be as small as possible since the Nelson-Oppen method must branch on each disjunct.

A disjunction is **minimal** if it is implied by ϕ_i and each smaller disjunciton is not implied by ϕ_i.

A simple procedure to find a minimal disjuction:

- First, consider the disjunction of all equalities at once.
- If it is not implied, then no subset is implied either, so we are done.
- Otherwise, drop each equality in turn: if the remaining disjunction is still implied by ϕ_i, continue with this smaller disjunction; otherwise, restore the equality and continue.
- When all equalities have been considered, the resulting disjunction is minimal.

The Nelson-Oppen Method

- For $i = 1, 2$, let T_i be a first-order theory of signature Σ_i (which includes $=$)
- Let $T = T_1 \cup T_2$
- Let C be a finite set of free constants (i.e., not in $\Sigma_1 \cup \Sigma_2$)

We consider only input problems of the form

$$L_1 \cup L_2$$

where each L_i is a finite set of ground (i.e., variable-free) $(\Sigma_i \cup C)$-literals

Note: Because of purification, there is no loss of generality in considering only ground $(\Sigma_i \cup C)$-literals

The Nelson-Oppen Method

Bare-bones, non-deterministic, non-incremental version:

Input: $L_1 \cup L_2$ with L_i finite set of ground $(\Sigma_i \cup C)$-literals
Output: **sat** or **unsat**

1. Guess an arrangement A, i.e., a set of equalities and disequalities over C such that

$$c = d \in A \quad \text{or} \quad c \neq d \in A \quad \text{for all} \quad c, d \in C$$

2. If $L_i \cup A$ is T_i-unsatisfiable for $i = 1$ or $i = 2$, return **unsat**
3. Otherwise, return **sat**

Correctness of the NO Method

Proposition (Termination) The method is terminating. (Trivially, because there is only a finite number of arrangements to guess.)

Proposition (Refutation Soundness) If the method returns **unsat** for every arrangement, the input is $(T_1 \cup T_2)$-unsatisfiable. (Because unsatisfiability in $(T_1 \cup T_2)$ is preserved.)

Proposition (Solution Soundness) If $\Sigma_1 \cap \Sigma_2 = \varnothing$ and T_1 and T_2 are stably infinite, when the method returns **sat** for some arrangement, the input is $(T_1 \cup T_2)$-is satisfiable. (Because satisfiability in $(T_1 \cup T_2)$ is preserved for stably infinite theories.)

Proposition (Completeness) For every arrangement, there is a terminating and progressive strategy to return **sat** or **unsat**. (Because the method is terminating - above - and never gets stuck on its way to deriving **sat** or **unsat**.)

Stably Infinite Theories

Def. Let Σ be a signature, let $S \subset \Sigma^S$ be a set of sorts, and let **T** be a Σ-theory. We say that **T** is stably-infinite with respect to S if for every **T**-satisfiable quantifier-free Σ-formula ϕ, there exists a **T**-interpretation I satisfying ϕ, such that $dom(\sigma)$ is infinite for each sort $\sigma \in S$. Nelson-Oppen requires that T_1 and T_2, which are to be combined, are stably-infinite over (at least) the set of common sorts $\Sigma_1^S \cap \Sigma_2^S$.

Many interesting theories are stably infinite:

- Theories of an infinite structure (e.g., integer arithmetic)
- Complete theories with an infinite model (e.g., theory of dense linear orders (over rationals or reals), theory of lists (of integers))
- Convex theories (e.g., EUF, linear real arithmetic)

Def. A theory **T** is convex iff, for any set L of literals $L \vDash_T s_1 = t_1 \vee \cdots \vee s_n = t_n \implies L \vDash_T s_i = t_i$ for some i

Note: With convex theories, arrangements do not need to be guessed—they can be computed by (theory) propagation

Stably Infinite Theories

Def. Let Σ be a signature, let $S \subset \Sigma^S$ be a set of sorts, and let $\mathbf{T}$ be a Σ-theory. We say that $\mathbf{T}$ is stably-infinite with respect to S if for every $\mathbf{T}$-satisfiable quantifier-free Σ-formula ϕ, there exists a $\mathbf{T}$-interpretation I satisfying ϕ, such that $dom(\sigma)$ is infinite for each sort $\sigma \in S$.

Other interesting theories are not stably infinite:

- Theories of a finite structure (e.g., theory of bit vectors of finite size, arithmetic modulo n)
- Theories with models of bounded cardinality (e.g., theory of strings of bounded length)
- Some equational/Horn theories

The Nelson-Oppen method has been extended to some classes of non-stably infinite theories

Stably Infinite Theories: Example

The theory of fixed-size bit-vectors contains sorts whose domains are all finite. Hence, this theory cannot be stably-infinite.

Example: Consider T_{array} where both indices and elements are of the same sort `bv`, so that the sorts of T_{array} are $\{\texttt{array}, \texttt{bv}\}$, and a theory T_{bv} that requires the sort `bv` to be interpreted as bit-vectors of size 1.

- Both theories are decidable and we would like to decide the combination theory in a Nelson-Oppen-like framework.
- Let $a_1, \ldots, a_5$ be array variables and consider the following constraints: $a_i \neq a_j$, for $1 \leq i < j \leq 5$.
- These constraints are entirely within T_{array}. Array theory solver is given all constraints and the bit-vector theory solver is given none.
- **Problem:** Array solver tells us these constraints are SAT, but there are only four possible different arrays with elements and indices over bit-vectors of size 1.

SMT Solving with Multiple Theories

Let $T_1, \dots, T_n$ be theories with respective solvers $S_1, \dots, S_n$

How can we integrate all of them cooperatively into a single SMT solver for $T = T_1 \cup \dots \cup T_n$?

Quick Solution:

1. Combine $S_1, \dots, S_n$ with Nelson-Oppen into a theory solver for **T**
2. Build a DPLL(T) solver as usual

Better Solution:

1. Extend DPLL(T) to DPLL($T_1, \dots, T_n$)
2. Lift Nelson-Oppen to the DPLL($X_1, \dots, X_n$) level
3. Build a DPLL($T_1, \dots, T_n$) solver

Modeling DPLL($T_1, \ldots, T_n$) Abstractly

- Let $n = 2$, for simplicity
- Let T_i be of signature Σ_i for $i = 1, 2$, with $\Sigma_1 \cap \Sigma_2 = \varnothing$
- Let C be a set of free constants
- Assume wlog that each input literal has signature $(\Sigma_1 \cup C)$ or $(\Sigma_2 \cup C)$ (no mixed literals)
- Let $\mathrm{M}|_i \stackrel{\text{def}}{=} \{(\Sigma_i \cup C)\text{-literals of M and their complement}\}$
- Let $\mathrm{I}(\mathrm{M}) \stackrel{\text{def}}{=} \{c = d \mid c, d \text{ occur in } C, \mathrm{M}|_1 \text{ and } \mathrm{M}|_2\} \cup \{c \neq d \mid c, d \text{ occur in } C, \mathrm{M}|_1 \text{ and } \mathrm{M}|_2\}$

  (interface literals)

Abstract DPLL Modulo Multiple Theories

Propagate, **Conflict**, **Explain**, **Backjump**, **Fail** (unchanged)

Decide $$\frac{l \in \texttt{Lits}(F) \cup \mathrm{I}(M) \quad l, \neg l \notin M}{M := M \bullet l}$$

Only change: decide on interface equalities as well

T-**Propagate** $$\frac{l \in \texttt{Lits}(F) \cup \mathrm{I}(M) \quad i \in \{1, 2\} \quad M \models_{T_i} l \quad l, \neg l \notin M}{M := M\ l}$$

Only change: propagate interface equalities as well, but reason locally in each T_i

Abstract DPLL Modulo Multiple Theories

***T*-Conflict**

$$\frac{\mathsf{C} = \mathtt{no} \quad l_1, \ldots, l_n \in \mathsf{M} \quad l_1, \ldots, l_n \models_{T_i} \bot \quad i \in \{1, 2\}}{\mathsf{C} := \neg l_1 \vee \cdots \vee \neg l_n}$$

***T*-Explain**

$$\frac{\mathsf{C} = l \vee D \quad \neg l_1, \ldots, \neg l_n \models_{T_i} \neg l \quad i \in \{1, 2\} \quad \neg l_1, \ldots, \neg l_n \prec_{\mathsf{M}} \neg l}{\mathsf{C} := l_1 \vee \cdots \vee l_n \vee D}$$

Only change: reason locally in each T_i

I-Learn

$$\frac{\models_{T_i} l_1 \vee \cdots \vee l_n \quad l_1, \ldots, l_n \in \mathsf{M}|_i \cup \mathrm{I}(\mathsf{M}) \quad i \in \{1, 2\}}{\mathsf{F} := \mathsf{F} \cup \{l_1 \vee \cdots \vee l_n\}}$$

New rule: for entailed disjunctions of interface literals

Example — Convex Theories

$$F := \underbrace{f(e_1) = a}_{0} \land \underbrace{f(x) = e_2}_{1} \land \underbrace{f(y) = e_3}_{2} \land \underbrace{f(e_4) = e_5}_{3} \land \underbrace{x = y}_{4} \land$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \land \underbrace{e_4 = 0}_{6} \land \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by ***T*-Propagate** $(1, 2, 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by ***T*-Propagate** $(5, 6, 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by ***T*-Propagate** $(0, 3, 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\neg 7 \lor \neg 10$	by ***T*-Conflict** $(7, 10 \models_{\mathrm{LRA}} \bot)$
Fail			by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \land \overbrace{f(x) = b}^{1} \land \overbrace{f(e_2) = e_3}^{2} \land \overbrace{f(e_1) = e_4}^{3} \land$$
$$\underbrace{1 \le x}_{4} \land \underbrace{x \le 2}_{5} \land \underbrace{e_1 = 1}_{6} \land \underbrace{a = b + 2}_{7} \land \underbrace{e_2 = 2}_{8} \land \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by **T-Propagate** $(0, 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \neg 4 \lor \neg 5 \lor 11 \lor 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	no	by **T-Propagate** $(0, 1, 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	$\neg 7 \lor \neg 13$	by **T-Conflict** $(7, 13 \models_{\mathrm{UF}} \bot)$
$0 \cdots 9\ 10\ \neg 13$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \neg 13\ \neg 11$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	no	by **T-Propagate** $(0, 1, \neg 13 \models_{\mathrm{UF}} \neg 11)$
$0 \cdots 9\ 10\ \neg 13\ \neg 11\ 12$	$F,\ \neg 4 \lor \neg 5 \lor 11 \lor 12$	no	by **Propagate**
$\vdots$			(exercise)
Fail	$\cdots$	$\cdots$	by **Fail**