Satisfiability Modulo Theories

Lecture 1: Abstract Proof Systems

Lydia Kondylidou

WS 2025/26

Agenda

Abstract Proof Systems

Satisfiability Proof Systems

Soundness, Completeness, Termination, and Progressiveness

A Decision Procedure for Propositional Logic

Strategies

Proofs for Automated Reasoning

What is a *proof*?

A sequence of steps leading from some assumptions to some conclusions

Each step should be convincing and should be drawn from a set of accepted *proof rules*

Proofs for Automated Reasoning

Proof theory is a branch of mathematical logic in which proofs themselves are formal objects we can prove things about

In AR, representing algorithms as proof systems has several advantages:

- They are modular and composable
- It is easier to prove things about the algorithms
- Can choose which implementation aspects to highlight and which to leave out

Abstract Proof Systems

An *abstract proof system* is a tuple $\mathbb{P} = \langle \mathbb{S}, \mathbb{R} \rangle$ where $\mathbb{S}$ is a set of **proof states** and $\mathbb{R}$ is a set of **proof rules**

Proof state: Data structure representing what is known at each stage of the proof

Example: a set of propositional formulas

Proof Rule: A partial function from proof states to sets of proof states

Example: Modus Ponens maps a state $\mathcal{S} \supseteq \{ \alpha, \alpha \implies \beta \}$ to the state set $\{ \mathcal{S} \cup \{ \beta \} \}$

Proof Rules

- Take an input proof state $\mathcal{S}$
- Are only applicable if $\mathcal{S}$ satisfies some *premises*
- Return one or more *derived* proof states, the *conclusions*

Notation:

$$\textbf{R} \; \frac{P_1 \quad P_2 \quad \cdots \quad P_m}{C_1 \;\mid\; C_2 \;\mid\; \cdots \;\mid\; C_n}$$

R is the rule's name (for reference)

Each P_i is a premise, each C_i is a conclusion

Note: Intuitively, premises are **conjunctive**; conclusions are **disjunctive**

A Proof System for Propositional Logic

Let $\mathbb{P}_{\mathsf{PL}} = \langle \mathbb{S}_{\mathsf{PL}}, \mathbb{R}_{\mathsf{PL}} \rangle$ where every proof state $\mathcal{S} \in \mathbb{S}_{\mathsf{PL}}$ is a set of well-formed formulas of PL

If $\mathbb{R}_{\mathsf{PL}}$ contains the *modus ponens* rule (**MP** for short) we can write **MP** as follows:

$$\textbf{MP} \quad \frac{\alpha \in \mathcal{S} \quad \alpha \implies \beta \in \mathcal{S} \quad \beta \notin \mathcal{S}}{\mathcal{S} \cup \{\beta\}}$$

Technically, **MP** is a proof rule *schema*

α and β are *parameters*, and each possible instantiation with well-formed formulas is a separate proof rule

For convenience, we will refer to proof rule schemas also as *proof rules*

Example

$$\textbf{MP} \quad \frac{\alpha \in \mathcal{S} \quad \alpha \implies \beta \in \mathcal{S} \quad \beta \notin \mathcal{S}}{\mathcal{S} \cup \{\beta\}}$$

Let a, b, c, d be propositional variables

What is the result of applying **MP** to the following proof states?

1. $\{a, a \implies b\}$
2. $\{\neg d, a \vee \neg c, \neg d \implies b\}$
3. $\{c, d, c \implies d\}$

Example

$$\textbf{MP} \quad \frac{\alpha \in \mathcal{S} \quad \alpha \implies \beta \in \mathcal{S} \quad \beta \notin \mathcal{S}}{\mathcal{S} \cup \{\beta\}}$$

Let a, b, c, d be propositional variables

What is the result of applying **MP** to the following proof states?

1. $\{a, a \implies b\}$ $\{a, a \implies b, b\}$
2. $\{\neg d, a \vee \neg c, \neg d \implies b\}$ $\{a \vee \neg c, \neg d, \neg d \implies b, b\}$
3. $\{c, d, c \implies d\}$ does not apply

A Proof System for Propositional Logic

Let $\mathcal{V}$ be the set of all propositional variables

Consider the following rule for $\mathbb{P}_{\mathsf{PL}}$:

$$\textbf{Split} \quad \frac{\alpha \in \mathcal{V} \quad \alpha \text{ occurs in some formula of } \mathcal{S} \quad \alpha \notin \mathcal{S} \quad \neg\alpha \notin \mathcal{S}}{\mathcal{S} \cup \{\alpha\} \qquad \mid \qquad \mathcal{S} \cup \{\neg\alpha\}}$$

Can we apply **Split** to $\{a \vee (b \wedge c), \neg d\}$?

A Proof System for Propositional Logic

Let $\mathcal{V}$ be the set of all propositional variables

Consider the following rule for $\mathbb{P}_{\mathsf{PL}}$:

$$\textbf{Split} \quad \frac{\alpha \in \mathcal{V} \quad \alpha \text{ occurs in some formula of } \mathcal{S} \quad \alpha \notin \mathcal{S} \quad \neg\alpha \notin \mathcal{S}}{\mathcal{S} \cup \{\alpha\} \qquad \mid \qquad \mathcal{S} \cup \{\neg\alpha\}}$$

Can we apply **Split** to $\{a \vee (b \wedge c), \neg d\}$?

Yes, if we choose to instantiate α with a, b, or c but not d

Let $\textbf{Split}_b$ be the proof rule obtained by instantiating α with b

Then, formally:

$$\{a \vee (b \wedge c), \neg d\} \xmapsto{\textbf{Split}_b} \{\{a \vee (b \wedge c), \neg d, b\}, \{a \vee (b \wedge c), \neg d, \neg b\}\}$$

A Proof System for Propositional Logic

Let $\mathcal{V}$ be the set of all propositional variables and let $\mathcal{L} = \mathcal{V} \cup \{ \neg\alpha \mid \alpha \in \mathcal{V} \}$

$\mathcal{L}$ is the set of all propositional *literals*, variables or negations of variables

Now consider the following rule for $\mathbb{P}_{\textsf{PL}}$:

$$\textbf{Contr} \quad \frac{\alpha \in \mathcal{V} \quad \alpha \in \mathcal{S} \quad \neg\alpha \in \mathcal{S}}{\textsc{unsat}}$$

where $\textsc{unsat}$ is a distinguished state

Note: The rule applies only to states with contradictory literals

Derivation Trees

Let $\mathbb{P} = \langle \mathbb{S}, \mathbb{R} \rangle$ be an abstract proof system

A *derivation tree* (in $\mathbb{P}$) from $\mathcal{S}_0$ is a finite tree with

- nodes from $\mathbb{S}$
- root $\mathcal{S}_0$
- an edge from a node $\mathcal{S}$ to a node $\mathcal{S}'$ iff
  $\mathcal{S}'$ is a conclusion of the application of a rule of $\mathbb{R}$ to $\mathcal{S}'$

A proof state $\mathcal{S} \in \mathbb{S}$ is *reducible* (in $\mathbb{P}$) if one or more proof rules of $\mathbb{R}$ applies to $\mathcal{S}$ (It is *irreducible* (in $\mathbb{P}$) otherwise)

A derivation tree is *reducible* (in $\mathbb{P}$) if at least one of its leaves is reducible (It is *irreducible* (in $\mathbb{P}$) otherwise)

Derivation Tree Example

What could a derivation tree from $\{b \implies c, \neg b \implies c, \neg c\}$ look like?

$$\{b \implies c, \neg b \implies c, \neg c\}$$

$$\dfrac{\{b \implies c, \neg b \implies c, \neg c\}}{\{b \implies c, \neg b \implies c, \neg c, b\} \qquad \{b \implies c, \neg b \implies c, \neg c, \neg b\}}\ \textbf{Split}$$

$$\dfrac{\{b \implies c, \neg b \implies c, \neg c\}}{\dfrac{\{b \implies c, \neg b \implies c, \neg c, b\}}{\{b \implies c, \neg b \implies c, \neg c, b, c\}}\ \textbf{MP} \qquad \dfrac{\{b \implies c, \neg b \implies c, \neg c, \neg b\}}{}}\ \textbf{Split}$$

$$\dfrac{\{b \implies c, \neg b \implies c, \neg c\}}{\dfrac{\dfrac{\{b \implies c, \neg b \implies c, \neg c, b\}}{\{b \implies c, \neg b \implies c, \neg c, b, c\}}\ \textbf{MP}}{\text{UNSAT}}\ \textbf{Contr} \qquad \dfrac{\{b \implies c, \neg b \implies c, \neg c, \neg b\}}{-}}\ \textbf{Split}$$

Derivation Tree Example

$$\cfrac{\{b \implies c, \neg b \implies c, \neg c\}}{\cfrac{\{b \implies c, \neg b \implies c, \neg c, b\}}{\cfrac{\{b \implies c, \neg b \implies c, \neg c, b, c\}}{\text{UNSAT}}\;\textbf{Contr}}\;\textbf{MP} \qquad \cfrac{\{b \implies c, \neg b \implies c, \neg c, \neg b\}}{\cfrac{\{b \implies c, \neg b \implies c, \neg c, \neg b, c\}}{}}\;\textbf{MP}}\;\textbf{Split}$$

$$\cfrac{\{b \implies c, \neg b \implies c, \neg c\}}{\cfrac{\{b \implies c, \neg b \implies c, \neg c, b\}}{\cfrac{\{b \implies c, \neg b \implies c, \neg c, b, c\}}{\text{UNSAT}}\;\textbf{Contr}}\;\textbf{MP} \qquad \cfrac{\{b \implies c, \neg b \implies c, \neg c, \neg b\}}{\cfrac{\{b \implies c, \neg b \implies c, \neg c, \neg b, c\}}{\text{UNSAT}}\;\textbf{Contr}}\;\textbf{MP}}\;\textbf{Split}$$

This tree is **irreducible**

Derivations

Let $\mathbb{P} = \langle \mathbb{S}, \mathbb{R} \rangle$ be an abstract proof system

A *derivation* (in $\mathbb{P}$) from a derivation tree τ_0 is a (possibly infinite) sequence $\tau_0, \tau_1, \ldots$ of derivation trees where each τ_{i+1} is derivable from τ_i by applying a rule from $\mathbb{R}$ to a leaf of τ_i

A derivation is *saturated* if it is finite and ends with an irreducible tree

Satisfiability Proof Systems

Let $\mathbb{P} = \langle \mathbb{S}, \mathbb{R} \rangle$ be an abstract proof system

$\mathbb{P}$ is a *satisfiability proof system* if $\mathbb{S}$ includes the distinguished states $\textsc{sat}$ and $\textsc{unsat}$

A rule of $\mathbb{R}$ is a *refuting* rule if its only conclusion is $\textsc{unsat}$

A rule of $\mathbb{R}$ is a *corroborating* rule if its only conclusion is $\textsc{sat}$

A *refutation tree* (from $\mathcal{S}$ in $\mathbb{P}$) is a derivation tree from $\mathcal{S}$ with only $\textsc{unsat}$ leaves

A *refutation* (of $\mathcal{S}$ in $\mathbb{P}$) is a derivation from $\mathcal{S}$ ending with a refutation tree

A *corroboration tree* (from $\mathcal{S}$ in $\mathbb{P}$) is a derivation tree from $\mathcal{S}$ with at least one $\textsc{sat}$ leaf

A *corroboration* (of $\mathcal{S}$ in $\mathbb{P}$ from) is a derivation from $\mathcal{S}$ ending with a corroborating tree

A Satisfiability Proof System for Propositional Logic

Can we extend $\mathbb{P}_{\mathsf{PL}}$ to be a satisfiability proof system?

Yes, simply by adding $\textsc{sat}$ to $\mathbb{S}_{\mathsf{PL}}$

Rule **Contr** is a refuting rule

We have no corroborating rules, yet

Soundness

Let $\mathbb{P} = \langle \mathbb{S}, \mathbb{R} \rangle$ be a satisfiability proof system

A set of *satisfiable proof states*, or *satisfiability predicate*, is a subset $\mathbb{S}^{\mathsf{Sat}} \subseteq \mathbb{S}$ such that
$\textsc{sat} \in \mathbb{S}^{\mathsf{Sat}}$ and $\textsc{unsat} \notin \mathbb{S}^{\mathsf{Sat}}$

$\mathbb{P}$ is *refutation sound* (wrt $\mathbb{S}^{\mathsf{Sat}}$) if **no** state $\mathcal{S} \in \mathbb{S}$ that has a refutation in $\mathbb{P}$ is in $\mathbb{S}^{\mathsf{Sat}}$

$\mathbb{P}$ is *solution sound* (wrt $\mathbb{S}^{\mathsf{Sat}}$) if **every** $\mathcal{S} \in \mathbb{S}$ that has a corroboration in $\mathbb{P}$ is in $\mathbb{S}^{\mathsf{Sat}}$

$\mathbb{P}$ is *sound* (wrt $\mathbb{S}^{\mathsf{Sat}}$) if it is **both** refutation and solution sound (wrt $\mathbb{S}^{\mathsf{Sat}}$)

Completeness and Termination

Let $\mathbb{P}$ be a satisfiability proof system with satisfiability predicate $\mathbb{S}^{\mathsf{Sat}}$

$\mathbb{P}$ is *complete* (wrt $\mathbb{S}^{\mathsf{Sat}}$) if for every $\mathcal{S} \in \mathbb{S}$, there exists either a corroboration or a refutation (wrt $\mathbb{S}^{\mathsf{Sat}}$) of $\mathcal{S}$ in $\mathbb{P}$

$\mathbb{P}$ is *terminating* if every derivation in $\mathbb{P}$ is finite

Recall

$\mathbb{P}$ is **sound** (wrt $\mathbb{S}^{\mathsf{Sat}}$) if (*i*) **no** state $\mathcal{S} \in \mathbb{S}$ that has a **refutation** in $\mathbb{P}$ is in $\mathbb{S}^{\mathsf{Sat}}$, and
(*ii*) **every** $\mathcal{S} \in \mathbb{S}$ that has a **corroboration** in $\mathbb{P}$ is in $\mathbb{S}^{\mathsf{Sat}}$

Proof Systems and Decision Procedures

If $\mathbb{P}$ is **sound** and **complete** wrt $\mathbb{S}^{\mathsf{Sat}}$ and **terminating**,
it induces a **decision procedure** for checking whether a $\mathcal{S}$ is in $\mathbb{S}^{\mathsf{Sat}}$:

Simply start with $\mathcal{S}$ and produce any derivation

It must eventually terminate

If the final tree is a refutation tree, then $\mathcal{S} \notin \mathbb{S}^{\mathsf{Sat}}$

Otherwise, $\mathcal{S} \in \mathbb{S}^{\mathsf{Sat}}$

A Decision Procedure for Propositional Logic

Recall: A **variable assignment** v is a partial mapping from $\mathcal{V}$ to $\{\text{true}, \text{false}\}$, and $v \models \mathcal{S}$ means that each formula in $\mathcal{S}$ evaluates to true under v

Let $\mathcal{S}$ be a set of propositional formulas

The *variable assignment v induced by* $\mathcal{S}$ is defined as follows:

$$v(p) = \begin{cases} \text{true} & \text{if } p \in \mathcal{S} \\ \text{false} & \text{if } \neg p \in \mathcal{S} \\ \textit{undefined} & \text{otherwise} \end{cases}$$

$\mathcal{S}$ *fully defines* v if

1. v is the variable assignment induced by $\mathcal{S}$ and
2. for each variable p occurring in $\mathcal{S}$, either $p \in \mathcal{S}$ or $\neg p \in \mathcal{S}$

A Decision Procedure for Propositional Logic

Let $\mathbb{P}_\mathsf{E} = \langle \mathbb{S}_\mathsf{E}, \mathbb{R}_\mathsf{E} \rangle$ where

$\mathbb{S}_\mathsf{E}$ consists of all sets of wffs plus the distinguished states $\textsc{sat}$ and $\textsc{unsat}$

$\mathbb{R}_\mathsf{E}$ consists of the following proof rules:

$$\textbf{Split} \quad \frac{p \in \mathcal{V} \qquad p \text{ occurs in some formula in } \mathcal{S} \qquad p \notin \mathcal{S} \qquad \neg p \notin \mathcal{S}}{\mathcal{S} \cup \{p\} \qquad | \qquad \mathcal{S} \cup \{\neg p\}}$$

$$\textbf{Sat} \quad \frac{\mathcal{S} \text{ fully defines } v \qquad v \models \mathcal{S}}{\textsc{sat}}$$

$$\textbf{Unsat} \quad \frac{\mathcal{S} \text{ fully defines } v \qquad v \not\models \alpha \quad \text{for some } \alpha \in \mathcal{S}}{\textsc{unsat}}$$

A Decision Procedure for Propositional Logic

Let $\mathbb{S}^{\mathsf{Sat}}$ consist of $\textsc{sat}$ and all satisfiable sets of wffs

- Each rule in $\mathbb{P}_{\mathsf{E}}$ is satisfiability preserving wrt $\mathbb{S}^{\mathsf{Sat}}$
- $\mathbb{P}_{\mathsf{E}}$ is sound wrt $\mathbb{S}^{\mathsf{Sat}}$
- $\mathbb{P}_{\mathsf{E}}$ is terminating
- $\mathbb{P}_{\mathsf{E}}$ is complete

Therefore, $\mathbb{P}_{\mathsf{E}}$ can be used as a decision procedure for the SAT problem

Example

Consider the set of propositional formulas $\{a, \neg a \vee b, a \implies \neg b\}$

$$\frac{\{a, \neg a \vee b, a \implies \neg b\}}{\quad}$$

$$\frac{\{a, \neg a \vee b, a \implies \neg b\}}{\{a, \neg a \vee b, a \implies \neg b, b\} \qquad \{a, \neg a \vee b, a \implies \neg b, \neg b\}}\ \textbf{Split}$$

$$\frac{\{a, \neg a \vee b, a \implies \neg b\}}{\dfrac{\{a, \neg a \vee b, a \implies \neg b, b\}}{\text{UNSAT}}\ \textbf{Unsat} \qquad \dfrac{\{a, \neg a \vee b, a \implies \neg b, \neg b\}}{\quad}}\ \textbf{Split}$$

$$\frac{\{a, \neg a \vee b, a \implies \neg b\}}{\dfrac{\{a, \neg a \vee b, a \implies \neg b, b\}}{\text{UNSAT}}\ \textbf{Unsat} \qquad \dfrac{\{a, \neg a \vee b, a \implies \neg b, \neg b\}}{\text{UNSAT}}\ \textbf{Unsat}}\ \textbf{Split}$$

Example

Alternatively, consider the set of propositional formulas $\{a, \neg a \vee \neg b, a \wedge \neg b\}$

$$\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b\}}{}$$

$$\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b\}}{\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b, b\}}{} \qquad \dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b, \neg b\}}{}}\ \textbf{Split}$$

$$\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b\}}{\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b, b\}}{\text{UNSAT}}\ \textbf{Unsat} \qquad \dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b, \neg b\}}{}}\ \textbf{Split}$$

$$\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b\}}{\dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b, b\}}{\text{UNSAT}}\ \textbf{Unsat} \qquad \dfrac{\{a, \neg a \vee \neg b, a \wedge \neg b, \neg b\}}{\text{SAT}}\ \textbf{Sat}}\ \textbf{Split}$$

Derivation Strategies

Sometimes, a proof system has some desirable properties

only if the rules are applied in a specific way

We capture those specific ways with rule application **strategies**

Derivation Strategies

Let $\mathbb{P} = \langle \mathbb{S}, \mathbb{R} \rangle$ be a proof system

A *(derivation) strategy* for $\mathbb{P}$ is a partial function that, when defined, takes a derivation tree τ in $\mathbb{P}$ and returns a new derivation tree τ' such that (τ, τ') is a derivation in $\mathbb{P}$

A derivation D in $\mathbb{P}$ *follows* a strategy π for $\mathbb{P}$

1. if each non-initial derivation tree in D is the result of applying π to the previous derivation tree, and
2. if D is finite, π is not defined for the final derivation tree

Derivation Strategy Example

Let $\prec$ be a total order on literals in $\mathcal{L}$ defined as alphabetical by variable name, with variables smaller than their negations (e.g., $a \prec \neg a \prec b \prec \neg b \prec \cdots$)

Consider the following strategy π_{PL} for $\mathbb{P}_{\mathrm{PL}}$, usable when every formula is either a literal or an implication between literal:

1. Find the first reducible leaf in a left-to-right depth-first traversal of the tree; if none, then stop (π_{PL} is undefined in that case)
2. if **MP** applies, apply it to the formulas l_1 and $l_1 \implies l_2$ where l_1 is minimal according to $\prec$, breaking ties by choosing a minimal l_2
3. Otherwise, if **Split** applies, apply it to the smallest variable p among those occurring in the state
4. Otherwise, apply **Contr** if possible

Properties of Strategies

Let $\mathbb{S}^{\mathsf{Sat}}$ be a satisfiability predicate for $\mathbb{P}$

A strategy π for $\mathbb{P}$ is

- *solution sound* wrt to $\mathbb{S}^{\mathsf{Sat}}$ if $\mathcal{S} \in \mathbb{S}^{\mathsf{Sat}}$ whenever there exists a corroboration in $\mathbb{P}$ from $\mathcal{S}$ following π
- *refutation sound* wrt to $\mathbb{S}^{\mathsf{Sat}}$ if $\mathcal{S} \notin \mathbb{S}^{\mathsf{Sat}}$ whenever there exists a refutation in $\mathbb{P}$ from $\mathcal{S}$ following π
- *sound* wrt $\mathbb{S}^{\mathsf{Sat}}$ if it is both refutation sound and solution sound wrt $\mathbb{S}^{\mathsf{Sat}}$
- *terminating* if every derivation in $\mathbb{P}$ following π is finite
- *progressive* if it is defined for every derivation tree that is not a refutation tree or a saturated tree

Properties of Strategies

Let $\mathbb{S}^{\mathsf{Sat}}$ be a satisfiability predicate for $\mathbb{P}$

Note:

If $\mathbb{P}$ is sound wrt $\mathbb{S}^{\mathsf{Sat}}$, then every strategy for $\mathbb{P}$ is also sound wrt $\mathbb{S}^{\mathsf{Sat}}$

If $\mathbb{P}$ is terminating, then every strategy for $\mathbb{P}$ is also terminating

$\mathbb{P}$ is complete iff there exists a progressive and terminating strategy for it