Designing a safe forward-chaining tactic using productive proofs

Kaustuv Chaudhuri, Arunava Gantait, Dale Miller

28 Sept 2025

Introduction

Saturation

Definition

Saturation means to compute the closure of a given set of formulas under a given set of inference rules.

(From Harald Ganzinger, 1996)

Saturation

In refutational theorem proving[1]

Set of formulas: FO formulas built from variables, function symbols, predicate symbols, logical connectives

Inference rule: *Resolution*:

$$\frac{C \vee A \quad D \vee \neg B}{(C \vee D)\sigma} \qquad \sigma = \mathrm{mgu}(A, B)$$

[1](From Leo Bachmair and Harald Ganzinger, 2001)

Saturation

In refutational theorem proving
Given a set of clauses, draw inferences from the clauses using the resolution rule

Add the conclusions to the set

Remove redundant clauses along the way

Termination: As soon as $\perp$ is obtained.

Saturation

In refutational theorem proving

Drawbacks
User has very little control over the saturation process

Output only makes sense once $\perp$ is obtained

Useful in fully automated systems, less useful as an interactive tactic

Saturation

In Datalog and similar systems[2]

Set of formulas: `Datalog` *clauses*:

> *Facts*: Atomic formulas: $P(x, y, z)$ for some predicate P
>
> *Rules*: $\forall \bar{x}.(A_1 \wedge A_2 \wedge \cdots \wedge A_n) \supset A_0$ where A_i's are atomic formulas. Also written `A0 :- A1,...,An`

Inference rule: *Elementary Production* (EP):

Consider a rule `R` of the form `A0 :- A1,...,An` and a list of ground facts `F1,...,Fn`. If there is a substitution θ such that for $1 \leq i \leq n$, $A_i\theta = F_i$, then infer in one step the fact $A_0\theta$.

[2](From S. Ceri, and G. Gottlob, and L. Tanca, 2012)

Saturation

In Datalog and similar systems
Given a set of `Datalog` clauses S, find all *ground* facts which can be inferred from S using the elementary production rule in one step

Add them to S

Maintaining that S is a set automatically takes care of redundancies

Termination: When all consequences of S are derived

Saturation

In Datalog and similar systems

Drawbacks
User has more control, but not general enough

Termination is guaranteed only because *ground facts* are considered

Saturation

Our system
We have tried to be as general as possible while also providing a high degree of control over the process

At the same time, we have formulated a general subset of formulas for which saturation (as defined in the paper) is provably terminating

Saturation

In our system

Set of formulas: A *context*:

Atomic formulas A

Non-atomic formulas B of a certain form (bipolar formulas)

Inference rule: The *fc* rule

Saturation

In our system
Given a context, find all consequences of the formulas B using the fc rule

Discard the ones which are already provable in one step from the context

Add the remaining to the set of atoms and continue

Termination: When the context is saturated

Saturation

In our system

Example

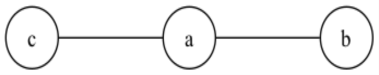
Consider defined:

A unary relation node (representing the nodes of a graph)

A binary relation adj (representing the adjacency relation)

A binary relation path (representing paths in the graph)

For the following graph:



Saturation

In our system

Example

So we have the following formulas in the context:

Atomic formulas:

$$\mathrm{node(a), node(b), node(c)}$$
$$\mathrm{adj(a, b), adj(a, c)}$$

Non-atomic formulas

$$\forall xy.(\mathrm{adj}(x, y) \supset \mathrm{path}(x, y))$$
$$\forall x.(\mathrm{node}(x) \supset \mathrm{path}(x, x))$$
$$\forall xy.(\mathrm{path}(x, y) \supset \mathrm{path}(y, x))$$
$$\forall xyz.((\mathrm{path}(x, y) \wedge \mathrm{path}(y, z)) \supset \mathrm{path}(x, z))$$

Saturation

In our system

Example

It is easy to see how, with this context, we can derive the formulas path(x, x) for $x \in \{a, b, c\}$ and also path(a, b) and path(a, c).

So assume we also have these in our context, and let us call it Γ.

We also include in our setting a *signature* Σ along with the context, containing all the symbols defined so far, as well as their types.

Assume some goal P

Saturation

In our system

Example (A derivation)

$$\dfrac{\text{(other stuff)} \qquad \dfrac{\Sigma :: \Gamma \Uparrow \mathrm{path}(b,a), \mathrm{path}(c,a) \vdash P \quad \text{(other stuff)}}{\Sigma :: \Gamma \Downarrow \mathrm{path}(b,a), \mathrm{path}(c,a), \mathrm{path}(a,a), \mathrm{path}(b,b), \mathrm{path}(c,c) \vdash P \Downarrow}\, R_l^d}{\Sigma :: \Gamma \Downarrow \forall xyz.(\mathrm{path}(x,y) \supset \mathrm{path}(y,x)) \vdash P \Downarrow}\, fc$$

What does it all mean??

Lots of jargon introduced in the last section (!):

What's a bipolar formula?

What do the *fc* and R_l^d rules look like?

What even *is* a saturated context?

And what do any of these have to do with a *safe forward-chaining tactic*?

Bipolar Formulas

Bipolar Formulas

Definition
A bipolar formula can be thought of a conjunction ($\wedge$) of formulas of the form

$$\forall \bar{x}.(A_1 \wedge \cdots \wedge A_n) \supset P$$

where A_i's are positive atoms and P is a purely positive formula. They are called *geometric implications* [a]

[a] (from Sara Negri, 2003)

Note
Compare this to `Datalog`'s *rules* which can be written in the form

$$\forall \bar{x}.(A_1 \wedge \cdots \wedge A_n) \supset A_0$$

Bipolar Formulas

The difference between `Datalog`'s rules and our generalized bipolar formulas is that the consequent of the implication is now a *purely positive*[3] *formula*.

These are given by:

$$P ::= \mathsf{t} \mid A \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid \exists x.P \mid t_1 = t_2$$

[3] This terminology comes from the literature on *focusing* in proof theory. (see, for example, Jean-Marc Andreoli, 1992, Chuck Liang and Dale Miller, 2007, 2009)

Bipolar Formulas

Example

The non-atomic formulas for path shown previously are all bipolar formulas. Other examples: (assume nat and plus are defined)

$$
\begin{aligned}
\text{associativity} :& \forall xyzwuv.\ \mathrm{plus}(x, y, z) \to \\
& \mathrm{plus}(z, w, u) \to \\
& \mathrm{plus}(y, w, v) \to \mathrm{plus}(x, v, u) \\
\text{determinacy} :& \forall xyzw.\ \mathrm{plus}(x, y, z) \to \\
& \mathrm{plus}(x, y, u) \to (z = u) \\
\text{totality} :& \forall xy.\mathrm{nat}(x) \to \mathrm{nat}(y) \to \\
& \exists z.(\mathrm{nat}(z) \land \mathrm{plus}(x, y, z))
\end{aligned}
$$

Bipolar Formulas

Why bipolar formulas?

Property: 1
For any P purely positive, it is *decidable* whether or not $\Sigma :: \Gamma \vdash P \Downarrow$ [4] is provable

The proof is immediate upon inspecting the focused proof system we are using (given in the paper)

Note
The provability of the *unfocused version* $\Sigma :: \Gamma \vdash P$ is not, in fact, decidable

[4] This notation is one of the two *phases* of focused proof systems. The other phase looks like $\Gamma \Uparrow \Delta \vdash P \Uparrow$ and will appear briefly later

Bipolar Formulas

Why bipolar formulas?
We add a restriction on the structure of bipolar formulas: for every geometric implication $\forall \bar{x}.(A_1 \wedge \cdots \wedge A_n) \supset P$, the free variables in P are also free in at least one of $A_1, \ldots, A_n$. Such formulas are called allowed clauses [5]

Property: 2
For an allowed clause $\forall \bar{x}.(P \supset Q)$, there are only *finitely many* substitutions θ such that $\Sigma :: \Gamma \vdash P\theta \Downarrow$ is provable

[5](from J. Lloyd and R. Topor, 1986)

The fc and R^d_l rules

The *fc* rule

Read: *forward-chain*

$$\frac{(\Sigma :: \Gamma \vdash P\theta_i \Downarrow)_{i=1}^{n} \quad \Sigma :: \Gamma \Downarrow Q\theta_1, \ldots, Q\theta_n, \Theta \vdash R}{\Sigma :: \Gamma \Downarrow \forall \bar{x}.(P \supset Q), \Theta \vdash R}\ fc$$

Here the θ_i are the substitutions for variables $\bar{x}$ for which $\Sigma :: \Gamma \vdash P\theta_i \Downarrow$ is provable. For allowed clauses, this is a finite set (cf. Property 2)

The *fc* rule

$$\frac{(\Sigma :: \Gamma \vdash P\theta_i \Downarrow)_{i=1}^{n} \quad \Sigma :: \Gamma \Downarrow Q\theta_1, \ldots, Q\theta_n, \Theta \vdash R}{\Sigma :: \Gamma \Downarrow \forall \bar{x}.(P \supset Q), \Theta \vdash R}\ fc$$

Operationally:

Select a geometric implication $\forall \overline{x}.(P \supset Q)$ from the *focus*

Find all θ_i such that $\Sigma :: \Gamma \vdash P\theta_i \Downarrow$ is provable

Add the corresponding $Q\theta_i$ to the focus

Note

The *fc* rule is not doing anything new. Here is how we would deal with a geometric implication in the context without *fc*: (Assume we have $A_1\theta, \ldots, A_n\theta$ in context for some θ, and goal C)

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, A_1\theta, \ldots, A_n\theta \vdash A_1\theta \wedge \cdots \wedge A_n\theta \Downarrow}\,(\wedge R, I_r)* \qquad \dfrac{\Gamma, A_1\theta, \ldots, A_n\theta \Uparrow P\theta \vdash C}{\Gamma, A_1\theta, \ldots, A_n\theta \Downarrow P\theta \vdash C}\,R_l}{\Gamma, A_1\theta, \ldots, A_n\theta \Downarrow A_1\theta \wedge \cdots \wedge A_n\theta \supset P\theta \vdash C}\,\supset L}{\Gamma, A_1\theta, \ldots, A_n\theta \Downarrow \forall \bar{x}.(A_1 \wedge \cdots \wedge A_n \supset P) \vdash C}\,(\forall L)*}{\Gamma, A_1\theta, \ldots, A_n\theta \vdash C}\,D_l
$$

The *fc* rule

The only new thing is *finding* all θ which can prove the antecedent instead of depending on them being in the context.

The R_l^d rule

Read: *Release-left-with-discard*

$$\frac{\Sigma :: \Gamma \Uparrow Q_1, \ldots, Q_n \vdash R \quad (\Sigma :: \Gamma \vdash P_i \Downarrow)_{i=1}^{m} \quad (\Sigma :: \Gamma \nvdash Q_j \Downarrow)_{j=1}^{n}}{\Sigma :: \Gamma \Downarrow P_1, \ldots, P_m, Q_1, \ldots, Q_n \vdash R}\ R_l^d$$

P_i's and Q_i's are positive formulas

P_i's are derivable from the context, Q_i's are not

Since they are *positive*, we can check whether $\Sigma :: \Gamma \vdash P \Downarrow$ or not (cf. Property: 1)

The R_l^d rule

$$\frac{\Sigma :: \Gamma \Uparrow Q_1, \ldots, Q_n \vdash R \quad (\Sigma :: \Gamma \vdash P_i \Downarrow)_{i=1}^{m} \quad (\Sigma :: \Gamma \nvdash Q_j \Downarrow)_{j=1}^{n}}{\Sigma :: \Gamma \Downarrow P_1, \ldots, P_m, Q_1, \ldots, Q_n \vdash R} \; R_l^d$$

Operationally:

Given a set of positive formulas in the context

Check which ones among them are provable from the context

Discard the redundant ones, *release* the rest (if they are atomic, they simply get added to context)

The R_l^d rule

This *is new*

The "usual" rule (*release-left*) looks like this:

$$\frac{\Sigma :: \Gamma \Uparrow \mathcal{P} \vdash Q}{\Sigma :: \Gamma \Downarrow \mathcal{P} \vdash Q} \; R_l$$

for $\mathcal{P}$ a set of positive formulas

The R_I^d rule

The rule R_I^d improves upon the usual R_I because it checks for redundancies in the set of formulas it is about to "release" (add to context) and discards those which are redundant

Now what?

So we have a class of formulas with nice properties and two new rules which exploit those properties.

How do we use these rules?

Are they even correct (i.e., sound and complete)?

How are they related to saturated contexts? Or safe forward-chaining tactics?

Saturation, proof-theoretically

Forward-chaining phase

A forward-chaining phase does the following:

> Selects (*focuses on*) the multiset of geometric formulas in the context

Forward-chaining phase

A forward-chaining phase does the following:

- Selects (*focuses on*) the multiset of geometric formulas in the context
- Applies the *fc* rule repeatedly. Thus, it replaces a geometric implication $\forall \bar{x}.(P \supset Q)$ in focus with the set $Q\theta_1, \ldots, Q\theta_n$ of its consequences

Forward-chaining phase

A forward-chaining phase does the following:

- Selects (*focuses on*) the multiset of geometric formulas in the context
- Applies the *fc* rule repeatedly. Thus, it replaces a geometric implication $\forall \bar{x}.(P \supset Q)$ in focus with the set $Q\theta_1, \ldots, Q\theta_n$ of its consequences
- Applies R_l^d to the final context with only positive formulas in focus to discard redundant formulas and only add new ones.

Forward-chaining phase

Observations
A forward-chaining phase affects only the context, not the goal formula

In the second step, n may be 0 for some geometric implication which is not provable in the current context

The R_l^d rule may end up discarding all the added positive formulas, if all of them are already derivable. In this case the phase is called a useless phase.

Saturated Context

Definition

A proof context consisting of the signature Σ + formula context Γ is said to be saturated if any forward-chaining phase starting from this context is bound to be a useless phase.

Saturated Context

Example (From the paper)

Assume that i is a primitive type, and the signature contains $f : i \to i$, $p : i \to o$, $q : i \to i \to o$ where o is the type of propositions, and t is a Σ-term of type i

$\mathrm{p}(t)$ and $\forall_i x.(\mathrm{p}(x) \supset \mathrm{p}(f(x)) \in \Gamma$:not saturated

$\Gamma = \{\mathrm{q}(a, b), \mathrm{q}(b, a), \forall_i x \forall_i y.\mathrm{q}(x, y) \supset \mathrm{q}(y, x)\}$:saturated

Correctness: Productive Proofs

Correctness of our rules

The *fc* rule, as stated, does not do anything "new". It's correctness is justified by Property:1

The R_l^d rule *does* do something new. In order to justify it's correctness we need the notion of productive proofs

Productive Proofs

Recall the "usual" release-left rule:

$$\frac{\Sigma :: \Gamma \Uparrow \mathcal{P} \vdash Q}{\Sigma :: \Gamma \Downarrow \mathcal{P} \vdash Q} \; R_l$$

Definition

An application of this rule is called *unproductive* if $\Sigma :: \Gamma \vdash P \Downarrow$ is provable for some $P \in \mathcal{P}$ and *productive* otherwise.

Productive Proofs

Definition

A proof is *productive* if all occurrences of R_I in the proof are productive

Intuitively, no R_I adds redundant information to the context in a productive proof

Evidently, productive proofs are sound.

Productive Proofs

The main theoretical result of this paper is that:

Theorem (Completeness of Productive Proofs)

Productive proofs are complete for a certain class of sequents, which look like:

$$B_1, \ldots, B_n, \mathcal{A} \vdash P$$

where B_i's are bipolar formulas, $\mathcal{A}$ is a set of atomic formulas, and P is a positive formula

The proof involves the following variant of *cut*:

$$\frac{\Sigma :: \Gamma \vdash P \Downarrow \qquad \Sigma :: \Gamma \Uparrow P, \Theta \vdash Q}{\Sigma :: \Gamma \Uparrow \Theta \vdash Q}\ cut \Updownarrow$$

Productive Proofs

Recall, from the introduction:

Set of formulas (a *context*):

Atomic formulas A

Non-atomic formulas of a certain kind B (bipolar formulas)

This is exactly the type of sequents for which productive proofs are complete

Almost there!

Completeness of productive proofs justifies the use of the R_l^d rule.

Thus, we now have:

- Two new rules *fc* and R_l^d which are correct for a certain class of formulas
- A saturation strategy (forward-chaining phases) which uses the above two rules
- A concrete termination condition (saturated contexts) for our strategy

This forward-chaining phase *is* the basis for the forward-chaining tactic in our paper

Almost there!

There are two more things to discuss:

- What does "safe" forward-chaining tactic mean?
- A note about the implementation

Safe-for-Saturation

Safety First

Another key result in our paper is the identification of a sufficient condition for a context for it to be safe to saturate with

Here "saturate with" = Keep on applying forward-chaining phases, until the context is saturated

And "safe to saturate with" = The context is guaranteed to saturate finitely

Safe-for-Saturation

Definition

The rigorous definition is in the paper. Intuitively, a context is safe for saturation if for every geometric implication $\forall \bar{x}.(P \supset Q)$ in it, Q is composed of only:

f, t, $\wedge, \vee$, equality

atomic formulas *without constructors of non-zero arity*

Note

This definition is very conservative. We expect future work to reveal a much larger subset of formulas which are safe.

Example (Safe formulas)

$$
\begin{aligned}
\text{commutativity :} & \forall xyz.\ \mathrm{nat}(z) \rightarrow \\
& \mathrm{plus}(x, y, z) \rightarrow \mathrm{plus}(y, x, z) \\
\text{determinacy :} & \forall xyzw.\ \mathrm{plus}(x, y, z) \rightarrow \\
& \mathrm{plus}(x, y, u) \rightarrow (z = u) \\
\text{associativity :} & \forall xyzwuv.\ \mathrm{plus}(x, y, z) \rightarrow \\
& \mathrm{plus}(z, w, u) \rightarrow \\
& \mathrm{plus}(y, w, v) \rightarrow \mathrm{plus}(x, v, u)
\end{aligned}
$$

Safe-for-Saturation

Example (Unsafe formulas)

$$\begin{aligned} \mathsf{totality} : \forall xy.\mathsf{nat}(x) \to \mathsf{nat}(y) \to \\ \exists z.(\mathsf{nat}(z) \wedge \mathsf{plus}(x, y, z)) \\ \mathsf{nat\text{-}nums} : \forall x.\mathsf{nat}(x) \to \mathsf{nat}(\mathsf{S}(x)) \end{aligned}$$

A Note on Implementation

Decising wisely

At multiple places I have specified that we *select* a multiset of formulas from the context

In theory the context only contains (conjunctions of) geometric formulas and positive atoms

We pick all geometric implications in the context using what is known as the $[\![D_I]\!]$ rule:

$$\frac{\Sigma :: \mathcal{N}, \mathcal{P} \Downarrow [\![\mathcal{N}]\!] \vdash Q}{\Sigma :: \mathcal{N}, \mathcal{P} \vdash Q}$$

Deciding wisely

In practice we cannot *a priori* limit the shape of the formulas in the context, especially in an interactive setting

So we allow the user to choose the formulas to apply the forward-chaining tactic on. (and hope they are geometric!)

This also makes this a truly interactive tactic which one can use in incremental construction of proofs

Conclusion

Summary

Goal: A forward-chaining tactic that will be as automated as possible and can be used to saturate a context

Methodology: Use a focused proof system, in particular, the positive formulas, which are already conducive to forward-chaining proofs

Summary

Contributions:

- Identification of a class of formulas (bipolar formulas) with nice properties
- Formulation of two new rules (fc and R_l^d) that use those nice properties to model forward-chaining and and removal of redundant formulas
- Development of the theoretical background using productive proofs that are sound and complete for sequents with bipolar formulas in context and a positive formula as the goal
- Identification of a safe subset of formulas which guarantee termination of the saturation process
- Implementation of the theory as a safe forward-chaining tactic in Abella