Mechanizing the Splitting Framework

Ghilain Bergeron, Florent Krasnopol, Sophie Tourret

ITP

September 2025, Rekjavick, Iceland

Motivation

Motivation

The question

Is AVATAR refutational complete?

Motivation

The question

Is AVATAR refutational complete?

AVATAR = splitting technique implemented in Vampire

Motivation

The question

Is AVATAR refutational complete?

AVATAR $=$ splitting technique implemented in Vampire

very successful (+421 rank 1 TPTP problems when introduced [V. 2014])

Motivation

The question

Is AVATAR refutational complete?

AVATAR = splitting technique implemented in Vampire

very successful (+421 rank 1 TPTP problems when introduced [V. 2014])

Spoiler

Yes!*

Motivation

The question

Is AVATAR refutational complete?

AVATAR = splitting technique implemented in Vampire

very successful (+421 rank 1 TPTP problems when introduced [V. 2014])

Spoiler

Yes!*

* under some fairness conditions, as described in our framework [E. et. al. 2021] that is kind of tricky so we decided to verify it with the proof assistant Isabelle/HOL but there is a lot of work to do so for now we are only done with a simple version of splitting and this is what this talk is about.

Splitting Framework

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

× $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ Q(a) \vee Q(c)$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

- × $P(a),\ P(b),\ \neg P(x) \lor Q(x),\ Q(a) \lor Q(c)$
- ○ $P(a),\ P(b),\ \neg P(x) \lor Q(x),\ Q(a) \lor Q(c),\ Q(a),\ Q(b)$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

- × $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ Q(a) \vee Q(c)$
- ○ $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ \cancel{Q(a) \vee Q(c)},\ Q(a),\ Q(b)$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

× $P(a),\ P(b),\ \neg P(x) \lor Q(x),\ Q(a) \lor Q(c)$

○ $P(a),\ P(b),\ \neg P(x) \lor Q(x),$ ~~$Q(a) \lor Q(c)$~~, $Q(a),\ Q(b)$

× $P(a),\ \neg P(x) \lor P(f(x)),\ \neg P(f(a))$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

× $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ Q(a) \vee Q(c)$

○ $P(a),\ P(b),\ \neg P(x) \vee Q(x),$ ~~$Q(a) \vee Q(c)$~~, $Q(a),\ Q(b)$

× $P(a),\ \neg P(x) \vee P(f(x)),\ \neg P(f(a))$

○ $P(a),\ \neg P(x) \vee P(f(x)),\ P(f(a)),\ \neg P(f(a))$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

× $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ Q(a) \vee Q(c)$

○ $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ \cancel{Q(a) \vee Q(c)},\ Q(a),\ Q(b)$

× $P(a),\ \neg P(x) \vee P(f(x)),\ \neg P(f(a))$

○ $P(a),\ \neg P(x) \vee P(f(x)),\ P(f(a)),\ \neg P(f(a)),\ \bot$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

× $P(a),\ P(b),\ \neg P(x) \lor Q(x),\ Q(a) \lor Q(c)$

○ $P(a),\ P(b),\ \neg P(x) \lor Q(x),$ ~~$Q(a) \lor Q(c)$~~, $Q(a),\ Q(b)$

× $P(a),\ \neg P(x) \lor P(f(x)),\ \neg P(f(a))$

○ ~~$P(a),\ \neg P(x) \lor P(f(x)),\ P(f(a)),\ \neg P(f(a))$~~, $\bot$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

× $P(a),\ P(b),\ \neg P(x) \lor Q(x),\ Q(a) \lor Q(c)$

○ $P(a),\ P(b),\ \neg P(x) \lor Q(x),$ ~~$Q(a) \lor Q(c)$~~, $Q(a),\ Q(b)$

× $P(a),\ \neg P(x) \lor P(f(x)),\ \neg P(f(a))$

○ ~~$P(a),\ \neg P(x) \lor P(f(x)),\ P(f(a)),\ \neg P(f(a))$~~, $\bot$

× $P(a),\ \neg P(x) \lor P(f(x))$

Saturation

saturate (verb): apply a given calculus and given redundancy deletion techniques to a set of formulas up to the limit.

With resolution and subsumption

- × $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ Q(a) \vee Q(c)$
- ○ $P(a),\ P(b),\ \neg P(x) \vee Q(x),\ \cancel{Q(a) \vee Q(c)},\ Q(a),\ Q(b)$

- × $P(a),\ \neg P(x) \vee P(f(x)),\ \neg P(f(a))$
- ○ $\cancel{P(a),\ \neg P(x) \vee P(f(x)),\ P(f(a)),\ \neg P(f(a))},\ \bot$

- × $P(a),\ \neg P(x) \vee P(f(x))$
- ○ $P(a),\ \neg P(x) \vee P(f(x)),\ P(f(a)),\ P(f(f(a))),\ \ldots$

Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \ldots \qquad P_n}{C}$ implies $\{P_1, \ldots, P_n\} \models \{C\}$

Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \ldots \qquad P_n}{C}$ implies $\{P_1, \ldots, P_n\} \models \{C\}$

simplifications: $\begin{array}{c} P_1 \qquad \ldots \qquad P_n \\ \hline\hline C_1 \qquad \ldots \qquad C_m \end{array}$ implies $\{C_1, \ldots, C_m\} \models \{P_i\}$ for all i

Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \ldots \qquad P_n}{C}$ implies $\{P_1, \ldots, P_n\} \models \{C\}$

simplifications: $\dfrac{P_1 \qquad \ldots \qquad P_n}{\overline{C_1 \qquad \ldots \qquad C_m}}$ implies $\{C_1, \ldots, C_m\} \models \{P_i\}$ for all i

Completeness

$N_0 \models \{\bot\}$

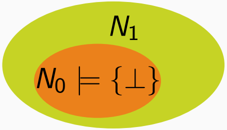
Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \dots \qquad P_n}{C}$ implies $\{P_1, \dots, P_n\} \models \{C\}$

simplifications: $\begin{array}{ccc} P_1 & \dots & P_n \\ \hline\hline C_1 & \dots & C_m \end{array}$ implies $\{C_1, \dots, C_m\} \models \{P_i\}$ for all i

Completeness



N_1

$N_0 \models \{\bot\}$

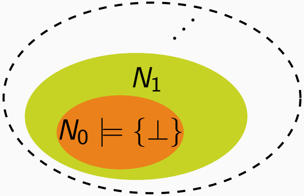
Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \ldots \qquad P_n}{C}$ implies $\{P_1, \ldots, P_n\} \models \{C\}$

simplifications: $\dfrac{P_1 \qquad \ldots \qquad P_n}{\dfrac{}{C_1 \qquad \ldots \qquad C_m}}$ implies $\{C_1, \ldots, C_m\} \models \{P_i\}$ for all i

Completeness



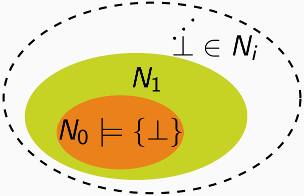
Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \quad \dots \quad P_n}{C}$ implies $\{P_1, \dots, P_n\} \models \{C\}$

simplifications: $\dfrac{P_1 \quad \dots \quad P_n}{\dfrac{}{C_1 \quad \dots \quad C_m}}$ implies $\{C_1, \dots, C_m\} \models \{P_i\}$ for all i

Completeness



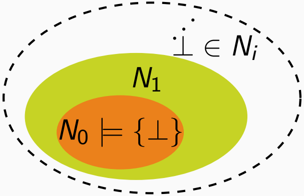
Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \dots \qquad P_n}{C}$ implies $\{P_1, \dots, P_n\} \models \{C\}$

simplifications: $\begin{array}{ccc} P_1 & \dots & P_n \\ \hline\hline C_1 & \dots & C_m \end{array}$ implies $\{C_1, \dots, C_m\} \models \{P_i\}$ for all i

Dynamic Completeness



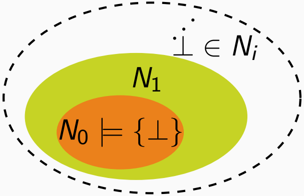
Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \dots \qquad P_n}{C}$ implies $\{P_1, \dots, P_n\} \models \{C\}$

simplifications: $\begin{array}{c} P_1 \qquad \dots \qquad P_n \\ \hline\hline C_1 \qquad \dots \qquad C_m \end{array}$ implies $\{C_1, \dots, C_m\} \models \{P_i\}$ for all i

Dynamic Completeness



Static Completeness

N saturated and $N \models \{\bot\}$ implies $\bot \in N$.

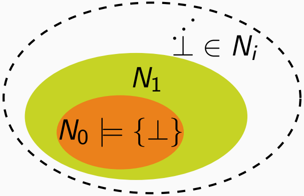
Desired Properties for a Saturation-based Calculus [B. & G. 2001, W. et al. 2020]

Soundness

inferences: $\dfrac{P_1 \qquad \ldots \qquad P_n}{C}$ implies $\{P_1, \ldots, P_n\} \models \{C\}$

simplifications: $\dfrac{P_1 \qquad \ldots \qquad P_n}{C_1 \qquad \ldots \qquad C_m}$ implies $\{C_1, \ldots, C_m\} \models \{P_i\}$ for all i

Dynamic Completeness



Static Completeness

$\Longleftrightarrow$ N saturated and $N \models \{\bot\}$ implies $\bot \in N$.

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

With resolution

$$\begin{array}{c} \neg P(a) \qquad P(x) \vee Q(y,b) \qquad \neg Q(z,z) \\ \hline\hline | \end{array}$$

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

With resolution

$$\begin{array}{c} \neg P(a) \qquad P(x) \vee Q(y,b) \qquad \neg Q(z,z) \\ \hline\hline \dfrac{\neg P(a) \qquad P(x)}{} \qquad \neg Q(z,z) \qquad \mid \end{array}$$

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

With resolution

$$\begin{array}{c} \neg P(a) \qquad P(x) \lor Q(y, b) \qquad \neg Q(z, z) \\ \hline\hline \dfrac{\neg P(a) \qquad P(x)}{\bot} \qquad \neg Q(z, z) \qquad \mid \end{array}$$

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

With resolution

$$\dfrac{\neg P(a) \qquad P(x) \lor Q(y,b) \qquad \neg Q(z,z)}{\dfrac{\neg P(a) \qquad P(x)}{\bot} \qquad \neg Q(z,z) \quad \mid \quad \neg P(a) \qquad \dfrac{Q(y,b) \qquad \neg Q(z,z)}{}}$$

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

With resolution

$$\dfrac{\neg P(a) \qquad P(x) \lor Q(y, b) \qquad \neg Q(z, z)}{\dfrac{\neg P(a) \qquad P(x)}{\bot} \qquad \neg Q(z, z) \qquad \mid \qquad \neg P(a) \qquad \dfrac{Q(y, b) \qquad \neg Q(z, z)}{\bot}}$$

Splitting

Intuition

Partition the search space by splitting a formula into independent subformulas.

With resolution

$$\begin{array}{c} \neg P(a) \qquad P(x) \lor Q(y, b) \qquad \neg Q(z, z) \\ \hline\hline \dfrac{\neg P(a) \qquad P(x)}{\bot} \qquad \neg Q(z, z) \qquad \Big| \qquad \neg P(a) \qquad \dfrac{Q(y, b) \qquad \neg Q(z, z)}{\bot} \end{array}$$

Works for any saturation-based calculus!

Properties Specific to Splitting

Static Completeness

If

- base calculus statically complete
- N saturated
- $N \models \{\bot\}$

then $\bot \in N$.

Dynamic Completeness

If

- base calculus dynamically complete
- derivation $(N_i)_i$ *fair*
- $N_0 \models \{\bot\}$

then $\bot \in N_i$ for some i.

Properties Specific to Splitting

Strong Static Completeness

If

- base calculus statically complete
- N locally saturated
- $N \models \{\bot\}$

then $\bot \in N$.

Strong Dynamic Completeness

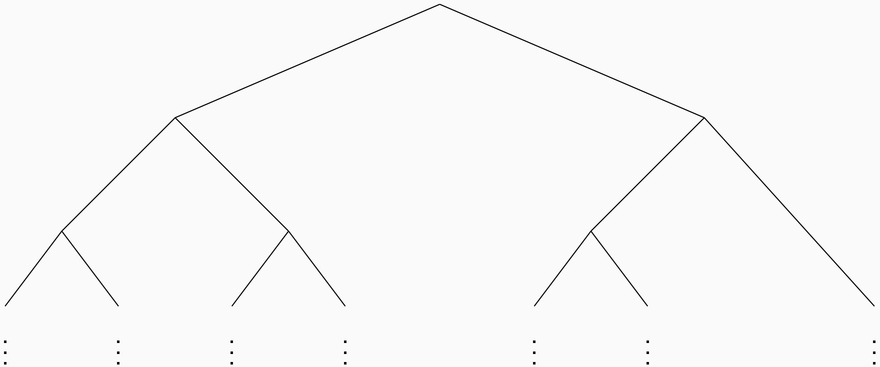
If

- base calculus dynamically complete
- derivation $(N_i)_i$ locally *fair*
- $N_0 \models \{\bot\}$

then $\bot \in N_i$ for some i.

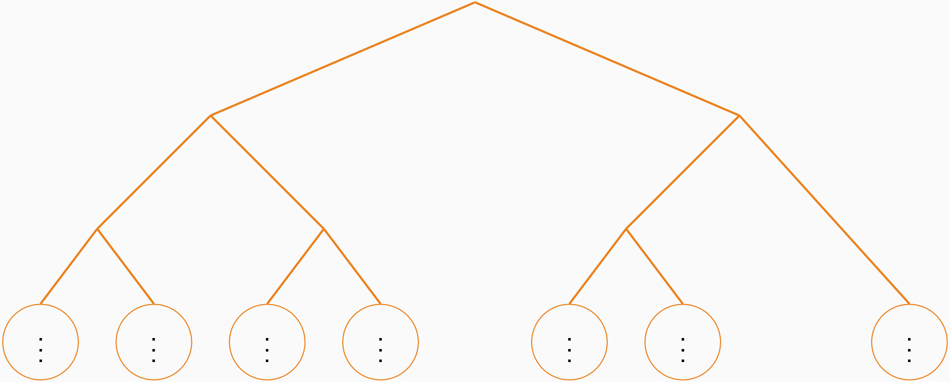
Locality

Saturation and fairness are too strong.



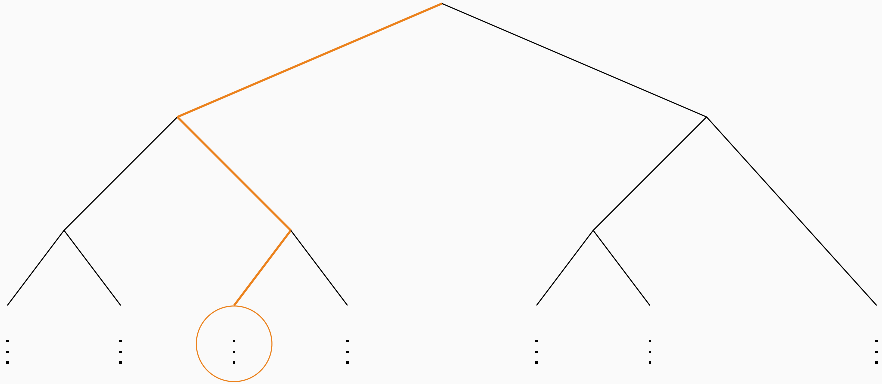
Locality

Saturation and fairness are too strong.



Locality

Saturation and fairness are too strong.



Mechanizing Splitting

Rules, Soundness and Completeness

Rules: Base, Unsat (mandatory)

Rules, Soundness and Completeness

Rules: Base, Unsat (mandatory)

Rules, Soundness and Completeness

Rules: Base, Unsat (mandatory) Tauto, Approx, StrongUnsat (optional)

Rules, Soundness and Completeness

Rules: BASE, UNSAT (mandatory)   TAUTO, APPROX, STRONGUNSAT (optional)

Rules, Soundness and Completeness

Rules: Base, Unsat (mandatory) Tauto, Approx, StrongUnsat (optional)

Simplifications: Split, Trim, Collect (optional)

Rules, Soundness and Completeness

Rules: Base, Unsat (mandatory) Tauto, Approx, StrongUnsat (optional)

Simplifications: Split, Trim, Collect (optional)

Soundness:

	✎	Isabelle		✎	Isabelle
Base	✓	✓	StrongUnsat	✓	✓
Unsat	✓	✓	Split	✗	✓
Tauto	✓	✓	Trim	✗	✓
Approx	✓	✓	Collect	✗	✓

Table 1: Inferences (premises $\models$ conclusions)

Rules, Soundness and Completeness

Rules: Base, Unsat (mandatory) Tauto, Approx, StrongUnsat (optional)

Simplifications: Split, Trim, Collect (optional)

Soundness:

	📝	Isabelle		📝	Isabelle
Base	✓	✓	StrongUnsat	✓	✓
Unsat	✓	✓	Split	✗	✓
Tauto	✓	✓	Trim	✗	✓
Approx	✓	✓	Collect	✗	✓

Table 1: Inferences (premises $\models$ conclusions)

	📝	Isabelle
Split	✗	✓
Trim	✗	✓
Collect	✓	✓

Table 2: Simplifications (conclusions $\models$ premises)

Rules, Soundness and Completeness

Rules: BASE, UNSAT (mandatory)   TAUTO, APPROX, STRONGUNSAT (optional)

Simplifications: SPLIT, TRIM, COLLECT (optional)

Soundness:

	📝	Isabelle
BASE	✓	✓
UNSAT	✓	✓
TAUTO	✓	✓
APPROX	✓	✓

	📝	Isabelle
STRONGUNSAT	✓	✓
SPLIT	✗	✓
TRIM	✗	✓
COLLECT	✗	✓

Table 1: Inferences (premises $\models$ conclusions)

	📝	Isabelle
SPLIT	✗	✓
TRIM	✗	✓
COLLECT	✓	✓

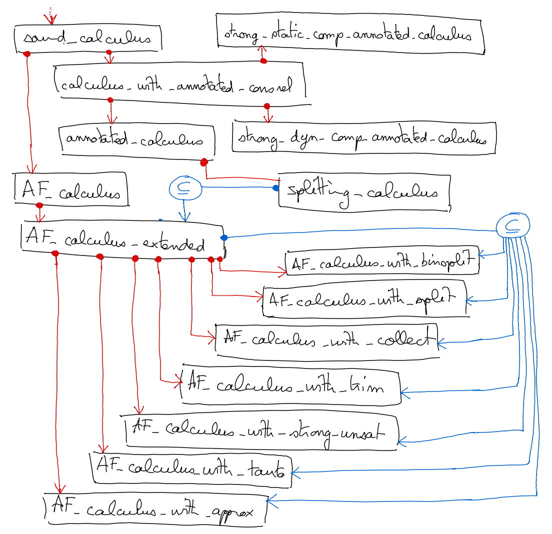
Table 2: Simplifications (conclusions $\models$ premises)

Completeness:

static ✓   dynamic ✓   strong static ✓   strong dynamic ✓

Modularity

Locale Structure (excerpt):



Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

- based on *Splitting without Backtracking* [R. & V. 2001];

Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

- based on *Splitting without Backtracking* [R. & V. 2001];
- implemented in Zipperposition.

Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

- based on *Splitting without Backtracking* [R. & V. 2001];
- implemented in Zipperposition.

Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

- based on *Splitting without Backtracking* [R. & V. 2001];
- implemented in Zipperposition.

The model

- uses binary-only splitting BinSplit (sound ✓) instead of Split;

Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

- based on *Splitting without Backtracking* [R. & V. 2001];
- implemented in Zipperposition.

The model

- uses binary-only splitting BINSPLIT (sound ✓) instead of SPLIT;
- plugs resolution in the framework;

Modeling Lightweight AVATAR with Resolution

Lightweight AVATAR is

- based on *Splitting without Backtracking* [R. & V. 2001];
- implemented in Zipperposition.

The model

- uses binary-only splitting BinSplit (sound ✓) instead of Split;
- plugs resolution in the framework;
- discharges assumptions.

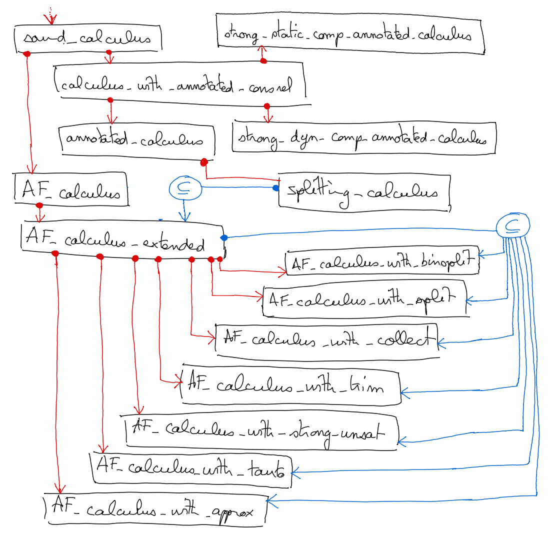
Resolution in Isabelle/HOL

Three entries in the Archive of Formal Proofs:

- **The Resolution Calculus for First-Order Logic**
  Anders Schlichtkrull, 2016.
- **Formalization of Bachmair and Ganzinger's Ordered Resolution Prover**
  Anders Schlichtkrull, Jasmin Blanchette, Dmitriy Traytel, Uwe Waldmann, 2018
- **Extensions to the Comprehensive Framework for Saturation Theorem Proving**
  Jasmin Blanchette, Sophie Tourret, 2020.

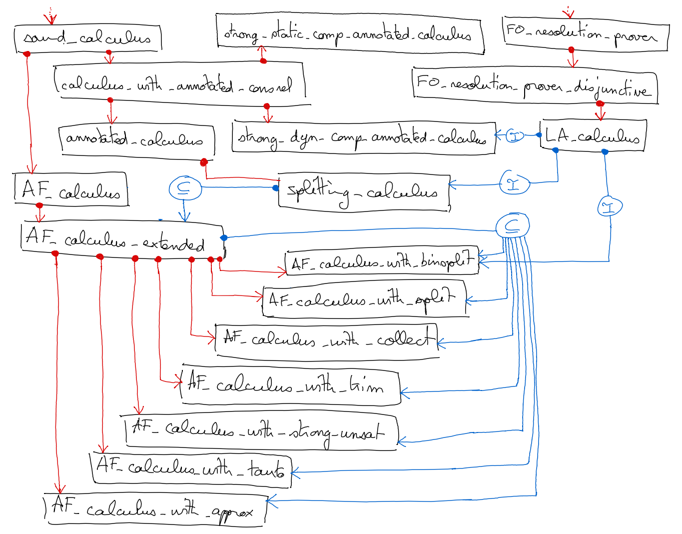
Plug and Play

Locale Structure (excerpt):



Plug and Play

Locale Structure (excerpt) + Resolution:



Conjunctive vs Disjunctive Entailment

If $M \models N$ means...

Conjuncive vs Disjunctive Entailment

If $M \models N$ means…

… $\bigwedge M \to \bigwedge N$ (as usual) then

Conjuntive vs Disjunctive Entailment

If $M \models N$ means...

... $\bigwedge M \to \bigwedge N$ (as usual) then

- $\{\bot\} \models N$
- subsets entailed (non-strict)
- $(\forall C \in N_2.\ N_1 \models \{C\}) \implies N_1 \models N_2$
- transitivity

Conjunctive vs Disjunctive Entailment

If $M \models N$ means...

... $\bigwedge M \to \bigwedge N$ (as usual) then

- $\{\bot\} \models N$
- subsets entailed (non-strict)
- $(\forall C \in N_2.\ N_1 \models \{C\}) \implies N_1 \models N_2$
- transitivity

... $\bigwedge M \to \bigvee N$ (needed for splitting) then

Conjunctive vs Disjunctive Entailment

If $M \models N$ means...

... $\bigwedge M \to \bigwedge N$ (as usual) then

- $\{\bot\} \models N$
- subsets entailed (non-strict)
- $(\forall C \in N_2.\ N_1 \models \{C\}) \implies N_1 \models N_2$
- transitivity

... $\bigwedge M \to \bigvee N$ (needed for splitting) then

- $\{\bot\} \models \{\}$
- reflexivity
- supersets entailment (both sides)
- cut rule
- compactness (a form of)

Conjunctive vs Disjunctive Entailment

If $M \models N$ means...

... $\bigwedge M \to \bigwedge N$ (as usual) then

- $\{\bot\} \models N$
- subsets entailed (non-strict)
- $(\forall C \in N_2.\ N_1 \models \{C\}) \implies N_1 \models N_2$
- transitivity

... $\bigwedge M \to \bigvee N$ (needed for splitting) then

- $\{\bot\} \models \{\}$
- reflexivity
- supersets entailment (both sides)
- cut rule
- compactness (a form of)

$\models_\wedge$ can be defined easily from $\models_\vee$ and agree on atomic sets.

Defining a Suitable Disjunctive Entailment from a Conjunctive One

$$M \models_{\vee} N \quad = \quad M \models_{\wedge} N$$

- ✗ $\{\bot\} \models \{\}$
- ● reflexivity
- ✗ supersets entailment (both sides)
- ✗ cut elimination
- ✗ compactness (a form of)

Defining a Suitable Disjunctive Entailment from a Conjunctive One

$$\begin{aligned} M \models_{\vee} N &= \cancel{M \models_{\wedge} N} \\ &= \exists C \in N.\ M \models_{\wedge} \{C\} \end{aligned}$$

- ✗ $\{\bot\} \models \{\}$
- reflexivity
- supersets entailment (both sides)
- cut elimination
- ✗ compactness (a form of)

Defining a Suitable Disjunctive Entailment from a Conjunctive One

$$\begin{aligned} M \models_{\vee} N \quad &= \quad \cancel{M \models_{\wedge} N} \\ &= \quad \cancel{\exists C \in N.\ M \models_{\wedge} \{C\}} \\ &= \quad M \models_{\wedge} \{\bot\} \vee \exists C \in N.\ M \models_{\wedge} \{C\} \end{aligned}$$

- $\{\bot\} \models \{\}$
- reflexivity
- supersets entailment (both sides)
- cut elimination

✗ compactness (a form of)

Defining a Suitable Disjunctive Entailment from a Conjunctive One

$$\begin{aligned} M \models_\vee N \quad &= \quad \cancel{M \models_\wedge N} \\ &= \quad \cancel{\exists C \in N.\ M \models_\wedge \{C\}} \\ &= \quad M \models_\wedge \{\bot\} \vee \exists C \in N.\ M \models_\wedge \{C\} \end{aligned}$$

- $\{\bot\} \models \{\}$
- reflexivity
- supersets entailment (both sides)
- cut elimination

✗ compactness (a form of) unless $\models_\wedge$ is already compact

Defining a Suitable Disjunctive Entailment from a Conjunctive One

$$\begin{aligned} M \models_\vee N &= \cancel{M \models_\wedge N} \\ &= \cancel{\exists C \in N.\ M \models_\wedge \{C\}} \\ &= \cancel{M \models_\wedge \{\bot\} \vee \exists C \in N.\ M \models_\wedge \{C\}} \\ &= M \models_\wedge \{\bot\} \vee \exists \text{ finite } M' \subseteq M.\ \exists C \in N.\ M' \models_\wedge \{C\} \end{aligned}$$

- $\{\bot\} \models \{\}$
- reflexivity
- supersets entailment (both sides)
- cut elimination
- compactness (a form of)

Work Done and Perspectives

Isabelle/HOL mechanization

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)
- TODO

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)
- TODO
  - ★ model-guidance + labeled splitting

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)
- TODO
  - ★ model-guidance + labeled splitting
  - ★ locking + SMT with complete enumerative instantiation

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)
- TODO
  - ★ model-guidance + labeled splitting
  - ★ locking + SMT with complete enumerative instantiation
  - ★ timestamps + AVATAR

Splitting Framework

Work Done and Perspectives

Isabelle/HOL mechanization

- Done ($\sim$ 8600 lines)
  - ✓ preliminary notions ($\sim$ 3500 lines)
  - ✓ splitting calculus ($\sim$ 3400 lines)
  - ✓ Lightweight AVATAR ($\sim$ 1700 lines)
- TODO
  - ★ model-guidance + labeled splitting
  - ★ locking + SMT with complete enumerative instantiation
  - ★ timestamps + AVATAR

Splitting Framework

Thank you!

Splitting Rules

$$\frac{C}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \qquad\qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}}\ \textsc{Split} \qquad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}} \textsc{Split} \qquad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}} \textsc{Split} \qquad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \qquad)_{i=1}^{n}}{D} \textsc{Base}$$

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}}\ \textsc{Split} \qquad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \leftarrow A_i)_{i=1}^{n}}{D \leftarrow \bigcup_{i=1}^{n} A_i}\ \textsc{Base}$$

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}}\ \textsc{Split} \quad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \leftarrow A_i)_{i=1}^{n}}{D \leftarrow \bigcup_{i=1}^{n} A_i}\ \textsc{Base} \qquad\qquad \frac{(\bot \leftarrow A_i)_{i=1}^{n}}{\bot}\ \textsc{Unsat}$$

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^n \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^n} \textsc{Split} \quad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \leftarrow A_i)_{i=1}^n}{D \leftarrow \bigcup_{i=1}^n A_i} \textsc{Base} \qquad \frac{(\bot \leftarrow A_i)_{i=1}^n}{\bot} \textsc{Unsat} \quad \text{if } \bigcup_{i=1}^n \bot \leftarrow A_i \models \bot$$

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}} \textsc{Split} \quad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \leftarrow A_i)_{i=1}^{n}}{D \leftarrow \bigcup_{i=1}^{n} A_i} \textsc{Base} \qquad \frac{(\bot \leftarrow A_i)_{i=1}^{n}}{\bot} \textsc{Unsat} \quad \text{if } \bigcup_{i=1}^{n} \bot \leftarrow A_i \models \bot$$

$+$ more optional rules

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}}\ \textsc{Split} \qquad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \leftarrow A_i)_{i=1}^{n}}{D \leftarrow \bigcup_{i=1}^{n} A_i}\ \textsc{Base} \qquad\qquad \frac{(\bot \leftarrow A_i)_{i=1}^{n}}{\bot}\ \textsc{Unsat} \qquad \text{if } \bigcup_{i=1}^{n} \bot \leftarrow A_i \models \bot$$

$+$ more optional rules

- approximate formula with constraint

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}} \textsc{Split} \quad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

$$\frac{(C_i \leftarrow A_i)_{i=1}^{n}}{D \leftarrow \bigcup_{i=1}^{n} A_i} \textsc{Base} \qquad \frac{(\bot \leftarrow A_i)_{i=1}^{n}}{\bot} \textsc{Unsat} \quad \text{if } \bigcup_{i=1}^{n} \bot \leftarrow A_i \models \bot$$

$+$ more optional rules

- approximate formula with constraint
- remove formula with unsat constrainst

Splitting Rules

$$\frac{C \leftarrow A}{\bot \leftarrow \{\neg a_i\}_{i=1}^{n} \cup A \qquad (C_i \leftarrow \{a_i\})_{i=1}^{n}}\ \textsc{Split} \qquad \text{if } C \text{ splittable into } C_1, \ldots, C_n$$

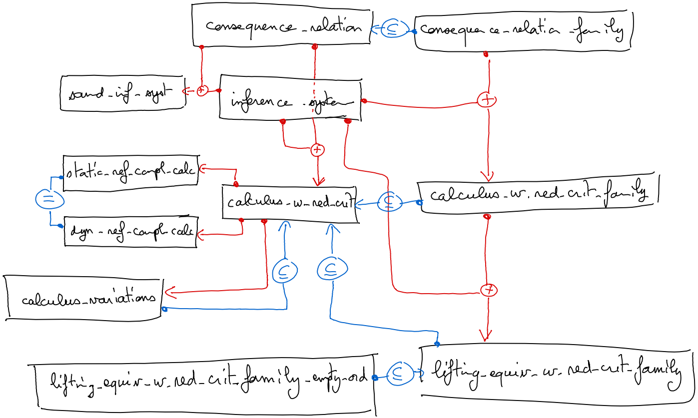
$$\frac{(C_i \leftarrow A_i)_{i=1}^{n}}{D \leftarrow \bigcup_{i=1}^{n} A_i}\ \textsc{Base} \qquad \frac{(\bot \leftarrow A_i)_{i=1}^{n}}{\bot}\ \textsc{Unsat} \quad \text{if } \bigcup_{i=1}^{n} \bot \leftarrow A_i \models \bot$$

$+$ more optional rules

- approximate formula with constraint
- remove formula with unsat constrainst
- trim constraints
- ...

Finding a Suitable Redundancy Criterion

Saturation Framework's Core:



Finding a Suitable Redundancy Criterion

Saturation Framework's Core + Ordered Resolution: