"Descente Infinie" Induction Principle and Saturation-based Theorem Provers

Sorin Stratulat

Department of Computer Science, UFR-MIM

Université de Metz

GECCOO Seminars, September 2004

Property

- Specification: algorithm, description of a system, ...
- Properties: invariants, liveness, ...

- Specification: algorithm, description of a system, ...
- Properties: invariants, liveness, ...
- The specification is sound if the properties are satisfied

- Specification: algorithm, description of a system, ...
- Properties: invariants, liveness, ...
- The specification is sound if the properties are satisfied
- Justification: proofs, counterexamples

- Specification: algorithm, description of a system, ...
- Properties: invariants, liveness, ...
- The specification is sound if the properties are satisfied
- Justification: proofs, counterexamples

☞ Need of formal specification and reasoning

Formal Specifications: Algebraic Specifications

Syntax

$\mathbf{F}$ function symbols	$\mathcal{T}(\mathbf{F}, \mathbf{V})$ terms
$\mathbf{P}$ predicate symbols	$\mathcal{A}(\mathbf{P}, \mathbf{F}, \mathbf{V})$ atomic formulas
$\mathbf{V}$ variables	$\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$

Formal Reasoning

axioms $\boldsymbol{Ax}$, conjecture $\boldsymbol{\phi}$

consequence relation $\models$

$$\boldsymbol{Ax} \models^{?} \boldsymbol{\phi}$$

$\models$ is an inductive consequence if $\boldsymbol{Ax} \models \boldsymbol{\phi} \quad \Leftrightarrow \quad \boldsymbol{Ax} \models \boldsymbol{\phi\gamma}$

☞ Difficulty: check an infinite number of ground instances $\boldsymbol{\phi\gamma}$

$$\left.\begin{array}{lcl} \mathbf{0} + \boldsymbol{x} & = & \boldsymbol{x} \\ \boldsymbol{s}(\boldsymbol{x}) + \boldsymbol{y} & = & \boldsymbol{s}(\boldsymbol{x} + \boldsymbol{y}) \end{array}\right\} \models^{?} \boldsymbol{x} + \mathbf{0} = \boldsymbol{x}, \forall \boldsymbol{x} \in \mathbb{N}$$

☞ by induction + reasoning techniques

Induction Orders

☞ $\leq$ on $\mathcal{P}$ is a partial order if:

- reflexivity: $\phi \leq \phi, \forall \phi \in \mathcal{P}$
- transitivity: if $\phi \leq \psi$ and $\psi \leq \rho$ then $\phi \leq \rho, \forall \phi, \psi, \rho \in \mathcal{P}$
- antisymmetry: if $\phi \leq \psi$ and $\psi \leq \phi$ then $\phi = \psi, \forall \phi, \psi \in \mathcal{P}$

☞ $\leq$ is a total order if, in addition,

- comparability (Trichotomy law): $\phi \leq \psi$ or $\psi \leq \phi, \forall \phi, \psi \in \mathcal{P}$

☞ $\leq$ is Noetherian if there is no infinite sequence $\phi_0 > \phi_1 > \ldots$

"Descente infinie" Induction Principle [Fermat (1659)]

$\mathcal{P}$ (infinite) set of formulas with a Noetherian order

$\mathcal{P}$ has only true formulas iff
for any counterexample from $\mathcal{P}$ there exists a smaller one

Applications :

- implicit induction provers
- saturation-based provers

"Descente Infinie" Provers

1. the proofs are built by successive applications of inference rules: $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \boldsymbol{\Phi}$
2. the proof is finite
3. $\mathcal{P}$ is the set of ground instances of proof's formulas
4. the application of an inference rule over a counterexample guarantees the existence of a smaller one in $\mathcal{P}$

☞ if ϕ has a counterexample, then a smallest counterexample is in the last state of the proof.

Implicit Induction vs. Saturation-based Proofs

- implicit induction

  $\boldsymbol{Ax} \models \phi$ if $\{\phi\} \vdash \ldots \vdash \emptyset$

- saturation-based

  ☞ unsatisfiability of a set of saturated formulas $\boldsymbol{E}$

  $$\boldsymbol{E'} \vdash \ldots \vdash \boldsymbol{E}$$

  such that $\boldsymbol{E}$ contains a smallest counterexample easy to detect

  ☞ a set of axioms is built from $\boldsymbol{E}$ (model generation)

1. Methodology to build “descente infinie” provers
2. Saturation-based provers
3. Conclusions and future work

1.1 Contextual Cover Sets

[Stratulat, 2000]

☞ An inference rule can be applied on ϕ if it does not (globally) eliminate smallest counterexamples from the proof

1. $\boldsymbol{Ax} \models \phi$, i.e. ϕ has no counterexample, or
2. $\mathbf{C}^{\mathbf{2}}_{<\phi} \models \phi$, i.e. if ϕ has a counterexample then $\mathbf{C^2}$ contains a smaller one, or
3. $\mathbf{C}^{\mathbf{1}}_{\leq\phi} \models \phi$, i.e. if ϕ has a counterexample then there is a smaller or equivalent counterexample.

☞ $\mathbf{C}^{\mathbf{1}}_{\leq\phi}$ and $\mathbf{C}^{\mathbf{2}}_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C^1}, \mathbf{C^2})$

☞ $\mathbf{C}^{\mathbf{1}}_{\leq\phi}$ and $\mathbf{C}^{\mathbf{2}}_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C^1}, \mathbf{C^2})$

When $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \mathbf{\Phi}$

$\mathbf{\Phi}$ is a contextual cover set (C.C.S.) of ϕ in $\mathbf{C}$ iff

$$\boldsymbol{Ax} \cup \mathbf{C}^{\mathbf{1}}_{\leq\phi\gamma} \cup \mathbf{C}^{\mathbf{2}}_{<\phi\gamma} \cup \mathbf{\Phi}_{\leq\phi\gamma} \models \phi\gamma$$

for any ground substitution γ

☞ $\mathbf{C}^1_{\leq\phi}$ and $\mathbf{C}^2_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C}^1, \mathbf{C}^2)$

When $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \mathbf{\Phi}$

> $\mathbf{\Phi}$ is a contextual cover set (C.C.S.) of ϕ in $\mathbf{C}$ iff
>
> $$\boldsymbol{Ax} \cup \mathbf{C}^1_{\leq\phi\gamma} \cup \mathbf{C}^2_{<\phi\gamma} \cup \mathbf{\Phi}_{\leq\phi\gamma} \models \phi\gamma$$
>
> for any ground substitution γ

Particular cases: cover set strict empty universal

☞ $\mathbf{C}^{\mathbf{1}}_{\leq\phi}$ and $\mathbf{C}^{\mathbf{2}}_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C}^{\mathbf{1}}, \mathbf{C}^{\mathbf{2}})$

When $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \mathbf{\Phi}$

> $\mathbf{\Phi}$ is a contextual cover set (C.C.S.) of ϕ in $\mathbf{C}$ iff
>
> $$\boldsymbol{Ax} \cup \mathbf{\Phi}_{\leq\phi\gamma} \models \phi\gamma$$
>
> for any ground substitution γ

Particular cases: cover set strict empty universal

☞ $\mathbf{C}^1_{\leq\phi}$ and $\mathbf{C}^2_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C}^1, \mathbf{C}^2)$

When $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \mathbf{\Phi}$

> $\mathbf{\Phi}$ is a contextual cover set (C.C.S.) of ϕ in $\mathbf{C}$ iff
>
> $$\boldsymbol{Ax} \cup \mathbf{C}^1_{\leq\phi\gamma} \cup \mathbf{C}^2_{<\phi\gamma} \cup \mathbf{\Phi}_{<\phi\gamma} \models \phi\gamma$$
>
> for any ground substitution γ

Particular cases: cover set strict empty universal

☞ $\mathbf{C}^1_{\leq\phi}$ and $\mathbf{C}^2_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C^1}, \mathbf{C^2})$

When $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \mathbf{\Phi}$

> $\mathbf{\Phi}$ is a contextual cover set (C.C.S.) of ϕ in $\mathbf{C}$ iff
>
> $$\boldsymbol{Ax} \cup \mathbf{C}^1_{\leq\phi\gamma} \cup \mathbf{C}^2_{<\phi\gamma} \models \phi\gamma$$
>
> for any ground substitution γ

Particular cases: cover set strict empty universal

☞ $\mathbf{C}^{\mathbf{1}}_{\leq\phi}$ and $\mathbf{C}^{\mathbf{2}}_{<\phi}$ represent induction hypotheses

To sum up, the used information when transforming ϕ in $\mathbf{\Phi}$:

axioms + context $\mathbf{C}$: $(\mathbf{C^1}, \mathbf{C^2})$

When $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \mathbf{\Phi}$

> $\mathbf{\Phi}$ is a contextual cover set (C.C.S.) of ϕ in $\mathbf{C}$ iff
>
> $$\mathbf{C}^{\mathbf{1}}_{\leq\phi\gamma} \cup \mathbf{C}^{\mathbf{2}}_{<\phi\gamma} \cup \mathbf{\Phi}_{\leq\phi\gamma} \models \phi\gamma$$
>
> for any ground substitution γ

Particular cases: cover set strict empty universal

$\boldsymbol{\Psi} \sqsubseteq_{\mathbf{C}} \boldsymbol{\Phi}$ $\triangleq$ $\boldsymbol{\Phi}$ is a C.C.S. of any $\phi \in \boldsymbol{\Psi}$ in $\mathbf{C}$

1.1.1 Properties of C.C.S.

☞ the "contextually covers" relation $\sqsubseteq$ is a quasi-order
(reflexivity + transitivity)

$\boldsymbol{\Psi} \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}$ $\triangleq$ $\boldsymbol{\Phi}$ is a C.C.S. of any $\phi \in \boldsymbol{\Psi}$ in C

1.1.1 Properties of C.C.S.

☞ the "contextually covers" relation $\sqsubseteq$ is a quasi-order
(reflexivity + transitivity)

☞ horizontal composition

$$\boldsymbol{\Phi}_1 \sqsubseteq_{\mathsf{C}} \ldots \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}_i \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}_{i+1} \sqsubseteq_{\mathsf{C}} \ldots \boldsymbol{\Phi}_n$$

$\boldsymbol{\Phi}_i \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}_{i+1}$ such that $\boldsymbol{\Phi}_{i+1}$ strict, then $\boldsymbol{\Phi}_{j>i}$ strict

$\boldsymbol{\Psi} \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}$ $\triangleq$ $\boldsymbol{\Phi}$ is a C.C.S. of any $\phi \in \boldsymbol{\Psi}$ in C

1.1.1 Properties of C.C.S.

☞ the "contextually covers" relation $\sqsubseteq$ is a quasi-order (reflexivity + transitivity)

☞ horizontal composition

$\boldsymbol{\Phi}_1 \sqsubseteq_{\mathsf{C}} \ldots \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}_i \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}_{i+1} \sqsubseteq_{\mathsf{C}} \ldots \boldsymbol{\Phi}_n$

$\boldsymbol{\Phi}_i \sqsubseteq_{\mathsf{C}} \boldsymbol{\Phi}_{i+1}$ such that $\boldsymbol{\Phi}_{i+1}$ strict, then $\boldsymbol{\Phi}_{j>i}$ strict

☞ vertical composition $\boldsymbol{\Phi} = \{\phi_1, \ldots, \phi_n\}$

$\forall i \in [1..n],\ \{\phi_i\} \sqsubseteq_{\mathsf{C}} \boldsymbol{\Psi}_i \quad \Rightarrow \quad \boldsymbol{\Phi} \sqsubseteq_{\mathsf{C}} \bigcup_{j=1}^{n} \boldsymbol{\Psi}_j$

1.2 An Abstract Inference System: A

☞ abstraction of reasoning techniques

☞ automated reasoning process

Inference rule:

$$\boxed{\text{NAME}} \quad (\boldsymbol{E} \cup \{\phi\},\ \boldsymbol{H}) \vdash^{\mathsf{A}} (\boldsymbol{E} \cup \boldsymbol{\Phi},\ \boldsymbol{H'}) \ \text{[Conditions]}$$

- $\boldsymbol{E}$, ϕ, $\boldsymbol{\Phi}$ conjectures
- $\boldsymbol{H}$, $\boldsymbol{H'}$ premises = conjectures that do not contain a smallest counterexample. For example, when $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \boldsymbol{\Phi}$ and $\boldsymbol{\Phi}$ is a strict C.C.S. of ϕ, ϕ becomes a premise

AddPremise

$$(E \cup \{\phi\},\ H) \ \vdash^{\mathsf{A}} \ (E \cup \underbrace{\Phi_1 \cup \ldots \cup \Phi_p}_{\Phi},\ H \cup \{\phi\})$$

where (a) $\{\phi\} \ \sqsubseteq_{(H,\ E \cup \Phi)} \ \cup_{i=1}^{p} \{\psi_i\}$ and

(b) $\{\psi_i\} \ \sqsubseteq_{(H,\ E \cup \{\phi\} \cup (\Phi \setminus \Phi_j))}$ Φ_j strict, $\forall j \in [1..p]$

Simplify

$$(E \cup \{\phi\},\ H) \ \vdash^{\mathsf{A}} \ (E \cup \underbrace{\Phi_1 \cup \ldots \cup \Phi_p}_{\Phi},\ H)$$

where (a) $\{\phi\} \ \sqsubseteq_{(E \cup H \cup \Phi, \emptyset)} \ \cup_{i=1}^{p} \{\psi_i\}$ and

(b) $\{\psi_i\} \ \sqsubseteq_{(E \cup H \cup (\Phi \setminus \Phi_j), \{\phi\})}$ Φ_j, $\forall j \in [1..p]$

1.3 Reasoning Techniques

☞ perform computation, i.e. show how to apply $\boldsymbol{E} \cup \{\phi\} \vdash \boldsymbol{E} \cup \boldsymbol{\Phi}$

1.4 Reasoning Modules

implementation of non-compositional (elementary) C.C.S. with reasoning techniques

☞ generation function $\boldsymbol{g}$

☞ condition function $\boldsymbol{cond}$ (optional), such that

If $\boldsymbol{g}(\phi, \mathbf{C}) = \boldsymbol{\Phi}$ then $\{\phi\} \sqsubseteq_{\mathbf{C}} \boldsymbol{\Phi}$ under the condition $\boldsymbol{cond}(\phi, \mathbf{C})$

1.5 Methodology for Building "Descente Infinie" Provers

Given a set of conjectures $\mathbf{\Phi}$ and an inductive consequence relation $\models$

1. provide a Noetherian order over formulas (compatible with $\mathbf{\Phi}$)
2. provide a set of reasoning techniques (compatible with $\models$)
3. build the reasoning modules from the reasoning techniques
4. instantiate **A** with reasoning modules

2. Saturation-based Provers

2.1 Proof by Saturation Principle

Definition 2.1 *An* $\boldsymbol{I}$*-proof state* $(\boldsymbol{E}, \boldsymbol{H})$ *is saturated if* $\boldsymbol{E}$ *is closed under* $\boldsymbol{I}$*.*

☞ Saturated sets check the satisfiability of a set of formulas

Definition 2.2 *An inference system is refutationally complete if the empty clause* $\square$ *is included in any unsatisfiable saturated set of formulas.*

Application: proof by consistency

To prove $\boldsymbol{Ax} \models \phi$:

i. add $\{\neg\phi\}$ to $\boldsymbol{Ax}$ and saturate

ii. get $\square$

2.2 Candidate Models

☞ no more axioms during the saturation process

Notation: $\mathbf{A}_s = \mathbf{A}$ with universal C.C.S.

Admissible conditions to prove the refutational completeness:

1. a Noetherian order over formulas (here, universal clauses)
2. a candidate model for any saturated set of clauses E with no $\square$ such that there is an $\mathbf{A}_s$-inference rule applicable on any clause from E containing a smallest counterexample.

Theorem 2.1 $\mathbf{A}_s$ *is refutationally complete.*

2.3 Analysis of Existing Systems

1. representation of the system states in the $\mathbf{A}_s$ form (E, H)
2. verification of the admissible conditions
3. definition of the reasoning modules
4. definition of the inference rules using reasoning modules
5. instantiation test of the inference rules w.r.t. $\mathbf{A}_s$-rules
6. proposition of "improvements"

Examples:

1. $\mathcal{G}$ paramodulation-based system [Nieuwenhuis and Rubio, 2001]
2. RP resolution-based system [Bachmair and Ganzinger, 2001]

2.4 The $\mathcal{G}$ Paramodulation-based Inference System

SUPERPOSITION RIGHT:

$$E \cup \{\Gamma \Rightarrow s = t\} \vdash^{\mathcal{G}} E \cup \{\Gamma \Rightarrow s = t\} \cup \{\Gamma', \Gamma \Rightarrow s[r]_p = t\}$$

if there exists a clause $\Gamma' \Rightarrow l = r$ from E such that:
(a) $s/_p \equiv l, l \succ r, s \succ t$ **and**
(b) $l \succ u$ for all u occurring in Γ' **and**
(c) $s \succ v$ for all v occurring in Γ

SUPERPOSITION LEFT:

$$E \cup \{\Gamma, s = t \Rightarrow \Delta\} \vdash^{\mathcal{G}} E \cup \{\Gamma, s = t \Rightarrow \Delta\} \cup \{\Gamma', \Gamma, s[r]_p = t \Rightarrow \Delta\}$$

if there exists a clause $\Gamma' \Rightarrow l = r$ from E such that:
(a) $s/_p \equiv l, l \succ r, s \succ t$ **and**
(b) $l \succ u$ for all u occurring in Γ' **and**
(c) $s \succeq v$ for all v occurring in Γ, Δ

EQUALITY RESOLUTION:

$$E \cup \{\Gamma, s = s \Rightarrow \Delta\} \vdash^{\mathcal{G}} E \cup \{\Gamma, s = s \Rightarrow \Delta\} \cup \{\Gamma \Rightarrow \Delta\}$$

if $s \succeq v$ for all v occurring in Γ, Δ

2.4.1 The (E, H) Representation: $\mathcal{G}'$

SUPERPOSITION RIGHT:

$$(E \cup \{\Gamma \Rightarrow s = t\}, H) \vdash^{\mathcal{G}'} (E \cup \{\Gamma \Rightarrow s = t\} \cup \{\Gamma', \Gamma \Rightarrow s[r]_p = t\}, H)$$

if there exists a clause $\Gamma' \Rightarrow l = r$ from E such that:
(a) $s/_p \equiv l, l \succ r, s \succ t$ **and**
(b) $l \succ u$ for all u occurring in Γ' **and**
(c) $s \succ v$ for all v occurring in Γ

SUPERPOSITION LEFT:

$$(E \cup \{\Gamma, s = t \Rightarrow \Delta\}, H) \vdash^{\mathcal{G}'} (E \cup \{\Gamma, s = t \Rightarrow \Delta\} \cup \{\Gamma', \Gamma, s[r]_p = t \Rightarrow \Delta\}, H)$$

if there exists a clause $\Gamma' \Rightarrow l = r$ from E such that:
(a) $s/_p \equiv l, l \succ r, s \succ t$ **and**
(b) $l \succ u$ for all u occurring in Γ' **and**
(c) $s \succeq v$ for all v occurring in Γ, Δ

EQUALITY RESOLUTION:

$$(E \cup \{\Gamma, s = s \Rightarrow \Delta\}, H) \vdash^{\mathcal{G}'} (E \cup \{\Gamma, s = s \Rightarrow \Delta\} \cup \{\Gamma \Rightarrow \Delta\}, H)$$

if $s \succeq v$ for all v occurring in Γ, Δ

2.4.2 Admissible Conditions

☞ multiset extension of a total Noetherian order over ground terms.

Definition 2.3 *(Candidate Model for Equational Ground Horn Clauses) Let* ϕ *be a clause in a set of equational ground Horn clauses* Φ*. Then* $\epsilon_\phi = \{l \to r\}$ *iff* ϕ *is of the form* $\Gamma \Rightarrow l = r$ *and*

1. $R^*_\phi \not\models \phi$,
2. $l \succ r$ *and* $l \succ u$, *for all* u *occurring in* Γ, *and*
3. l *is irreducible by* R_ϕ,

where $R_\phi = \bigcup_{\phi \prec_c \psi} \epsilon_\psi$, *otherwise* $\epsilon_\phi = \emptyset$. *Given a set of clauses* Φ, *we can define* $R = \bigcup_{\psi \in \Phi} \epsilon_\phi$.

2.4.3 Reasoning Modules

1. $\boldsymbol{SR}$, based on Superposition Right reasoning technique that rewrites the maximal term of a ground Horn clause from the positive literal. $g_{SR}(\Gamma \Rightarrow s = t, (\{\Gamma' \Rightarrow l = r\}, \emptyset)) = \{\Gamma', \Gamma \Rightarrow s[r]_p = t\}$. (strict C.C.S.)

SUPERPOSITION RIGHT:

$$(E \cup \{\Gamma \Rightarrow s = t\}, H) \vdash^{\mathcal{G}'} (E \cup \{\Gamma \Rightarrow s = t\} \cup \{\Gamma', \Gamma \Rightarrow s[r]_p = t\}, H)$$

if there exists a clause $\Gamma' \Rightarrow l = r$ from E such that:
(a) $s/_p \equiv l, l \succ r, s \succ t$ **and**
(b) $l \succ u$ for all u occurring in Γ' **and**
(c) $s \succ v$ for all v occurring in Γ

2. ***SL***, based on Superposition Left reasoning technique that rewrites the maximal term from a negative literal. $g_{LR}((\Gamma, s = t \Rightarrow \Delta), (\{\Gamma' \Rightarrow l = r\}, \emptyset)) = \{\Gamma', \Gamma, s[r]_p = t \Rightarrow \Delta\}$. (strict C.C.S)

3. ***ER***, based on Equality Reasoning which eliminates negative identity literals. $g_{ER}((\Gamma, s = s \Rightarrow \Delta), (\emptyset, \emptyset)) = \{\Gamma \Rightarrow \Delta\}$. (strict C.C.S. if s is maximal)

2.4.4 Instantiation Result (w.r.t. Simplify)

$\mathcal{G}'$-rule (ϕ)	S. R.	S. L.	E. R.	SIMPLIFY
g_{RM}, Cxt (a)	$\{\phi\}, (\emptyset, \emptyset)$	$\{\phi\}, (\emptyset, \emptyset)$	$\{\phi\}, (\emptyset, \emptyset)$	$\cup_{i=1}^{p} \{\psi_i\}, (E \cup H \cup \Phi, \emptyset)$
RM, Cxt (b)	$SR, (E, \emptyset)$	$SL, (E, \emptyset)$	$ER, (\emptyset, \emptyset)$	$(E \cup H \cup (\Phi \backslash \Phi_j), \{\phi\})$

☞ also instances of ADDPREMISE

Theorem 2.2 $\mathcal{G}'$ *is refutationally complete.*

2.4.5 Improvements

☞ partial order over ground terms

☞ instances of SIMPLIFY:

- reduction on non-maximal terms (for S.R. and S.L.)
- $s = s$ is eliminated even if s is not a maximal term

☞ instances of ADDPREMISE:

- adding the processed conjecture to H
- no need to preserve the processed conjecture

Theorem 2.3 *The improved versions of* $\mathcal{G}$ *and* $\mathcal{G}'$ *are refutationally complete.*

2.5 The RP Resolution-based Inference System

1. Tautology Deletion:

$$\mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$$

if C is a tautology

2. Forward Subsumption:

$$\mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$$

if some clause in $\mathcal{P} \cup \mathcal{O}$ subsumes C

3. Backward Subsumption:

for $\mathcal{P}$: $\mathcal{N} \mid \mathcal{P} \cup \{C\} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$

for $\mathcal{O}$: $\mathcal{N} \mid \mathcal{P} \mid \mathcal{O} \cup \{C\} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$

if some clause in $\mathcal{N}$ properly subsumes C

6. Clause Processing:

$$\mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \cup \{C\} \mid \mathcal{O}$$

4. Forward Reduction:

$$\mathcal{N} \cup \{C \vee L\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O}$$

if it exists $D \vee L'$ in $\mathcal{P} \cup \mathcal{O}$ s. t. $\overline{L} = L'\sigma$ **and** $D\sigma \subseteq C$

5. Backward Reduction:

- for $\mathcal{P}$: $\mathcal{N} \mid \mathcal{P} \cup \{C \vee L\} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O}$

- for $\mathcal{O}$: $\mathcal{N} \mid \mathcal{P} \mid \mathcal{O} \cup \{C \vee L\} \vdash^{RP} \mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O}$

if it is a clause $D \vee L'$ in $\mathcal{N}$ s. t. $\overline{L} = L'\sigma$ **and** $D\sigma \subseteq C$

7. Inference Computation:

$$\emptyset \mid \mathcal{P} \cup \{C\} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O} \cup \{C\}$$

where $\mathcal{N}$ is $O_S^{\succ}(\mathcal{O}, C)$

2.5.1 The (E, H) Representation: RP'

☞ $\mathcal{N}$ and $\mathcal{P}$ ➠ E $\mathcal{O}$ ➠ H

Tautology Deletion:

$$(E \cup \{C\}, H) \vdash^{RP'} (E, H)$$

if C is a tautology

Forward Subsumption:

$$(E \cup \{C\}, H) \vdash^{RP'} (E, H)$$

if some clause in $E \cup H$ subsumes C

Backward Subsumption:

- for E: $(E \cup \{C\}, H) \vdash^{RP'} (E, H)$

- for H: $(E, H \cup \{C\}) \vdash^{RP'} (E, H)$

if some clause in E properly subsumes C

Forward Reduction:

$$(E \cup \{C \vee L\}, H) \vdash^{RP'} (E \cup \{C\}, H)$$

if it exists $D \vee L'$ in $E \cup H$ s. t. $\overline{L} = L'\sigma$ **and** $D\sigma \subseteq C$

Backward Reduction:

$$(E, H \cup \{C \vee L\}) \vdash^{RP'} (E \cup \{C\}, H)$$

if it is a clause $D \vee L'$ in E s. t. $\overline{L} = L'\sigma$ **and** $D\sigma \subseteq C$

Inference Computation:

$$(E \cup \{C\}, H) \vdash^{RP'} (E \cup \{\psi\}, H \cup \{C\})$$

where ψ is $O^{\succ}_{S}(H, C)$

2.5.2 Admissible Conditions

- $\prec$ is total on ground clauses
- the resolution-based computation $O^{\succ}_{S}(H, C)$ returns a new clause and operates on the clause C by using smaller clauses from H chosen according to a selection function S
- RP' satisfies the applicability property for counterexamples by the means of a candidate model built for $O^{\succ}_{S}(H, C)$.

2.5.3 Reasoning Modules

- TD, based on Tautology Deletion, builds an empty CCS. $g_{TD}(C, (\emptyset, \emptyset)) = \emptyset$.
- S, based on Subsumption, has $g_S(C, (\mathbf{C^1}, \emptyset)) = \emptyset$ if it exists a clause in $\mathbf{C^1}$ that subsumes C.
- SS, based on Strict Subsumption, has $g_S(C, (\emptyset, \mathbf{C^2})) = \emptyset$ if it exists a clause in $\mathbf{C^2}$ that properly subsumes C.
- SR, based on Subsumption Resolution, builds the CCS $g_{SR}(C \vee L, (\mathbf{C^1}, \emptyset)) = \{C\}$ if it exists $D \vee L'$ in $\mathbf{C^1}$ such that $\overline{L} = L'\sigma$ and $D\sigma \subseteq C$.

- O, based on $O_S^{\succ}$, builds a strict CCS such that $g_O(C, (\emptyset, \mathbf{C^2})) = O_S^{\succ}(\mathbf{C^2}, C)$.

2.5.4 Instantiation Result

RP'-rule	T.D.	F.S.	B.S.(FOR E)
A_s-rule	DELETE	DELETE	DELETE
g_{RM}, Cxt (a)	$\{C\}, (\emptyset, \emptyset)$	$\{C\}, (\emptyset, \emptyset)$	$\{E \cup C\}, (\emptyset, \emptyset)$
RM, Cxt (b)	$TD, (\emptyset, \emptyset)$	$S, (E \cup H, \emptyset)$	$SS, (\emptyset, E)$

RP'-rule	F. R.	I.C.
A_s-rule	SIMPLIFY	ADDPREMISE
g_{RM}, Cxt (a)	$\{C \cup L\}, (\emptyset, \emptyset)$	$\{C\}, (\emptyset, \emptyset)$
RM, Cxt (b)	$SR, (E \cup H, \emptyset)$	$O, (H, \emptyset)$

Theorem 2.4 *RP' is refutationally complete.*

2.5.5 Improvements

1. Backward Subsumption for E is eliminated since it is a particular case of Forward Subsumption.
2. Backward Subsumption for H is an instance of Eliminate Premise, for which a premise can become a conjecture unconditionally.
3. Forward Reduction can be presented in two versions:
   a) As instance of Simplify, as in the original form.
   b) As instance of AddPremise, if L is maximal in $L \vee C$ and either i) $D \vee L'$ in H such that $\overline{L} = L'\sigma$ and $D\sigma \subseteq C$, or ii) $D \vee L'$ in E such that $C \vee L \succ D\sigma \vee L'\sigma$.
4. Inference Computation may use clauses from E, smaller than C, and clauses from H, smaller or equal than C.

2 Saturation-based Provers

2.5.6 The Improved Version of RP

Tautology Deletion:

$$\mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$$

if C is a tautology

Forward Subsumption:

$$\mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$$

if a clause in $\mathcal{N} \cup \mathcal{P} \cup \mathcal{O}$ subsumes C

Backward Subsumption:

$$\mathcal{N} \mid \mathcal{P} \cup \{C\} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$$

if a clause in $\mathcal{N} \cup \mathcal{P} \cup \mathcal{O}$ subsumes C

Eliminate Premise:

$$\mathcal{N} \mid \mathcal{P} \mid \mathcal{O} \cup \{C\} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O}$$

Backward Premise:

$$\mathcal{N} \mid \mathcal{P} \mid \mathcal{O} \cup \{C\} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \cup \{C\} \mid \mathcal{O}$$

Backward Conjecture:

$$\mathcal{N} \mid \mathcal{P} \cup \{C\} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O}$$

Forward Reduction - Simplify:

$$\mathcal{N} \cup \{C \vee L\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O}$$

if $\exists\ D \vee L'$ in $\mathcal{N} \cup \mathcal{P} \cup \mathcal{O}$ s. t. $\overline{L} = L'\sigma$ **and** $D\sigma \subseteq C$

Forward Reduction - AddPremise:

$$\mathcal{N} \cup \{C \vee L\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \cup \{C \vee L\}$$

if L is maximal in $L \vee C$ **and either**
i) $D \vee L'$ in $\mathcal{O}$ such that $\overline{L} = L'\sigma$ **and** $D\sigma \subseteq C$, **or**
ii) $D \vee L'$ in $\mathcal{N} \cup \mathcal{P}$ such that $C \vee L \succ D\sigma \vee L'\sigma$.

Clause Processing:

$$\mathcal{N} \cup \{C\} \mid \mathcal{P} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \cup \{C\} \mid \mathcal{O}$$

Inference Computation:

$$\emptyset \mid \mathcal{P} \cup \{C\} \mid \mathcal{O} \vdash^{RP} \mathcal{N} \mid \mathcal{P} \mid \mathcal{O} \cup \{C\}$$

where $\mathcal{N}$ is $O_S^{\succ}(\mathcal{O}_{\preceq C}, C)$ **or** $O_S^{\succ}(\mathcal{P}_{\preceq C}, C)$

Comments:

1. Forward Subsumption may also use clauses from $\mathcal{N}$ to subsume C.
2. Backward Subsumption for $\mathcal{P}$ may also use clauses from $\mathcal{N} \cup \mathcal{P} \cup \mathcal{O}$ to subsume C. The "properly subsumption" restriction is no longer needed.
3. Backward Subsumption for $\mathcal{O}$ is replaced by the more general rule Eliminate Premise.
4. Forward Reduction comes in two versions: Forward Reduction - Simplify and Forward Reduction - AddPremise. The original version corresponds to Forward Reduction - Simplify, while the restricted version Forward Reduction - AddPremise transfers the processed conjecture in the "old" set of clauses, for participating in further inferences.
5. Backward Reduction for $\mathcal{P}$ can be simulated by applying firstly Backward Conjecture (the inverse of Clause Processing) and then Forward Reduction - Simplify followed by Clause Processing. It becomes useless.
6. Backward Reduction for $\mathcal{O}$ is simulated by Backward Premise followed by Forward Reduction - Simplify. It becomes useless, too.
7. Inference Computation may use clauses from E, smaller than C, and clauses from H, smaller or equal than C.

Theorem 2.5 *RP, RP' and their improved versions are refutationally complete.*

3. Conclusions and Future Work

3.1 Conclusions

☞ methodology for building "descente infinie" automated theorem provers

✔ clear separation between logic and computation

✔ modular construction w.r.t. reasoning techniques

✔ **A** and $\mathbf{A}_s$ are landmarks for other inference systems

- define at any proof step the induction hypotheses
- the contexts used in the inference rules show the potential of a prover

- applicability to any inductive consequence relation:

$\models_{ini}$	equational or conditional specifications
	$(e_1 \wedge \ldots \wedge e_n \Rightarrow e'_1)$
$\models_{obs}$	conditional specifications
$\models_{par}$	parameterized specifications
$\models_{pos/neg}$	positive/negative parameterized specifications
	$(\neg e_1 \wedge \ldots \wedge e_n \Rightarrow e'_1)$

3.2 Future Work

✔ Design of less-restrictive rules

$$\ldots (\{\phi\}, H) \vdash (\{\phi_1, \ldots, \phi_n\}, H) \vdash \ldots$$

3.2 Future Work

✔ Design of less-restrictive rules

$$\ldots(\{\phi\}, H) \vdash (\{\phi_1, \ldots, \quad\}, H) \vdash \ldots$$

3.2 Future Work

✔ Design of less-restrictive rules

$$\ldots(\{\phi\}, H) \vdash (\{\phi_1, \ldots, \quad\}, H) \vdash \ldots \vdash (\{\ldots \phi_n \ldots\}, H)$$

3.2 Future Work

✔ Design of less-restrictive rules

$$\ldots(\{\phi\}, H) \vdash (\{\phi_1, \ldots, \quad\}, H) \vdash \ldots \vdash (\{\ldots \phi_n \ldots\}, H)$$

☞ conjecture tracking

3.2 Future Work

- ✔ Design of less-restrictive rules

$$\ldots(\{\phi\}, H) \vdash (\{\phi_1, \ldots, \quad\}, H) \vdash \ldots \vdash (\{\ldots \phi_n \ldots\}, H)$$

  ☞ conjecture tracking

- ✔ Adding interactivity

3.2 Future Work

- ✔ Design of less-restrictive rules

  $$\ldots(\{\phi\}, H) \vdash (\{\phi_1, \ldots, \quad\}, H) \vdash \ldots \vdash (\{\ldots \phi_n \ldots\}, H)$$

  ☞ conjecture tracking

- ✔ Adding interactivity
- ✔ More reasoning techniques: library of reasoning modules
- ✔ Schema for the integration of reasoning modules

3.2 Future Work

- ✔ Design of less-restrictive rules

  $$\ldots(\{\phi\}, H) \vdash (\{\phi_1, \ldots, \quad\}, H) \vdash \ldots \vdash (\{\ldots \phi_n \ldots\}, H)$$

  ☞ conjecture tracking

- ✔ Adding interactivity
- ✔ More reasoning techniques: library of reasoning modules
- ✔ Schema for the integration of reasoning modules
- ✔ DEEP - a "DEscente infiniE" Prover

Stay tuned !!!

"Descente Infinie" Induction Principle and Saturation-based Theorem Provers

Sorin Stratulat

Department of Computer Science, UFR-MIM

Université de Metz

CASSIS Seminars, June 2004

[Bachmair and Ganzinger, 2001] L. Bachmair and H. Ganzinger. Resolution theorem proving. In Handbook of Automated Reasoning, pages 19–99, 2001.

[Nieuwenhuis and Rubio, 2001] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In Handbook of Automated Reasoning, pages 371–443, 2001.

[Stratulat, 2000] S. Stratulat. A general framework to build contextual cover set induction provers. Journal of Symbolic Computation, 2000. 43 pages (to appear).