

A Constraint Sequent Calculus for First-Order Logic with Linear Integer Arithmetic

Philipp Rümmer
Chalmers University of Technology, Gothenburg
philipp@chalmers.se

Talk at Microsoft Research
30th April 2008

- Background, logic, demo
- Constraints in tableau reasoning
- Calculus for first-order logic
- Calculus for integer arithmetic
- Results, conclusions

- Ideas partly developed in the context of the KeY system
⇒ Software verification

Two lines of work:

- Constraint solving to disprove program correctness, [Rümmer, Shah, TAP'07], [Velroyen, Rümmer, TAP'08]
- Handling of ground integer arithmetic (linear + nonlinear) in a sequent calculus, [Rümmer, Verify'07]

... which are here put together

But:

- Shown calculus/implementation is independent from KeY

The calculus in a nutshell

- Classical sequent/tableau calculus
- Non-normal-form calculus
- Free variables for handling quantifiers
- Constraints for describing variable instantiations
 - ⇒ Constraints are formulae in Presburger arithmetic
- Non-destructive
- Recursive application to handle constraints

- Complete for first-order logic (FOL)
- Decision procedure for Presburger arithmetic (PA)
- ... complete for further fragments (more details later)

- Partly implemented (“Princess”), more to be done

Linear integer arithmetic + uninterpreted predicates:

$$t ::= \alpha \mid \mathbf{x} \mid \mathbf{c} \mid \alpha t + \dots + \alpha t$$

$$\begin{aligned} \phi ::= & \phi \wedge \phi \mid \phi \vee \phi \mid \neg \phi \mid \forall \mathbf{x}.\phi \mid \exists \mathbf{x}.\phi \\ & \mid t \doteq 0 \mid t \geq 0 \mid t \leq 0 \mid \alpha \mid t \mid p(t, \dots, t) \end{aligned}$$

t ... terms

ϕ ... formulae

\mathbf{x} ... variables

\mathbf{c} ... constants

p ... uninterpreted predicates (fixed arity)

α ... integer literals (\mathbb{Z})

The Calculus in Detail

3 common ways to handle quantifiers (among others)

Trigger-matching

- Standard method in SMT-solvers
- Ground reasoning → efficient
- Heuristic → incomplete

Free-variable (FV) methods

- Standard method in FOL reasoning
- “Difficult” to integrate in tableau provers
- Good way to combine with theories yet to be found

Quantifier elimination for certain theories (like PA)

- Impossible for many logics
- Often very high complexity

3 common ways to handle quantifiers (among others)

Trigger-matching

- Standard method in SMT-solvers
- Ground reasoning → efficient
- Heuristic → incomplete

X Free-variable (FV) methods

- Standard method in FOL reasoning
- “Difficult” to integrate in tableau provers
- Good way to combine with theories yet to be found

X Quantifier elimination for certain theories (like PA)

- Impossible for many logics
- Often very high complexity

Background: FOL proving with FVs and constraints

$$\frac{}{\frac{}{\vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x))}}$$

$$\frac{\frac{}{\vdash (X = c \vee X = d) \wedge f(c) = f(X)}}{\vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x))}$$

$$\frac{\frac{\overline{\vdash X = c \vee X = d} \quad \vdash f(c) = f(X)}{\vdash (X = c \vee X = d) \wedge f(c) = f(X)}}{\vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x))}$$

$$\frac{\frac{\frac{\vdash X = c, X = d}{\vdash X = c \vee X = d} \quad \vdash f(c) = f(X)}{\vdash (X = c \vee X = d) \wedge f(c) = f(X)}}{\vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x))}$$

Background: FOL proving with FVs and constraints

$$\frac{\frac{[X \equiv c], [X \equiv d]}{\vdash X = c, X = d} \quad \frac{[f(c) \equiv f(X)]}{\vdash f(c) = f(X)}}{\vdash (X = c \vee X = d) \wedge f(c) = f(X)} \\ \vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x))$$

Background: FOL proving with FVs and constraints

$$\frac{\frac{[X \equiv c], [X \equiv d]}{\vdash X = c, X = d} \quad \frac{[f(c) \equiv f(X)]}{\vdash f(c) = f(X)}}{\frac{\vdash (X = c \vee X = d) \wedge f(c) = f(X)}{\vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x))}}$$

To close proof, compatible constraints have to be found for all branches:

$$X \equiv c \wedge f(c) \equiv f(X) \quad X \equiv d \wedge f(c) \equiv f(X)$$

(Martin Giese, PhD thesis: *Proof Search Without Backtracking for Free Variable Tableaux.*)

Full citizenship for constraints!

Constraint notation used here:

$$\Gamma \vdash \Delta \Downarrow C$$


Antecedent, Succedent
(sets of formulae)

Constraint
(formula)

Definition

$\Gamma \vdash \Delta \Downarrow C$ is *valid* if the formula $C \rightarrow \bigwedge \Gamma \rightarrow \bigvee \Delta$ is valid.

Full citizenship for constraints!

Constraint notation used here:

$$\underbrace{\Gamma \vdash \Delta}_{\text{Antecedent, Succedent}} \quad \underbrace{\Downarrow C}_{\text{Constraint}}$$

Antecedent, Succedent
(sets of formulae)

Constraint
(formula)

Definition

$\Gamma \vdash \Delta \Downarrow C$ is *valid* if the formula $C \rightarrow \bigwedge \Gamma \rightarrow \bigvee \Delta$ is valid.

In the example:

$$\frac{\frac{\frac{}{\vdash X = c, X = d} *}{\vdash X = c \vee X = d} * \Downarrow X = c}{\vdash (X = c \vee X = d) \wedge f(c) = f(X) \Downarrow X = c \wedge f(c) = f(X)} *}{\vdash \exists x. ((x = c \vee x = d) \wedge f(c) = f(x)) \Downarrow \dots}$$

$\Gamma \vdash \Delta \Downarrow ?$

Anticipated way of proof construction

analytic reasoning
about input formula \uparrow

$\Gamma \vdash \Delta \Downarrow ?$

Anticipated way of proof construction

analytic reasoning
about input formula \uparrow

$$\begin{array}{c} \Gamma_1 \vdash \Delta_1 \Downarrow? \\ \vdots \\ \Gamma \vdash \Delta \Downarrow? \end{array}$$

Anticipated way of proof construction

analytic reasoning
about input formula \uparrow

$$\frac{\Gamma_2 \vdash \Delta_2 \Downarrow ?}{\Gamma_1 \vdash \Delta_1 \Downarrow ?}$$
$$\vdots$$
$$\Gamma \vdash \Delta \Downarrow ?$$

Anticipated way of proof construction

analytic reasoning
about input formula \uparrow

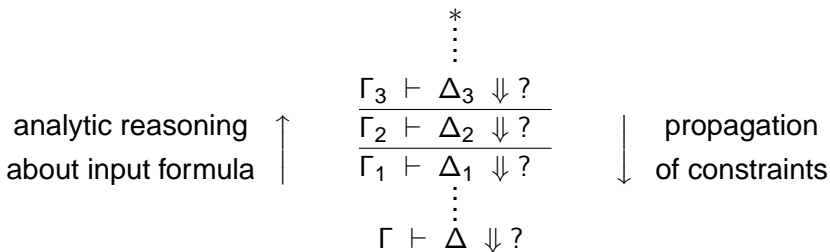
$$\frac{\frac{\Gamma_3 \vdash \Delta_3 \Downarrow?}{\Gamma_2 \vdash \Delta_2 \Downarrow?}}{\Gamma_1 \vdash \Delta_1 \Downarrow?}$$
$$\vdots$$
$$\Gamma \vdash \Delta \Downarrow?$$

Anticipated way of proof construction

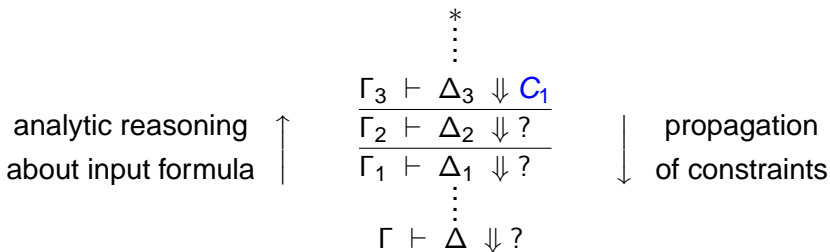
analytic reasoning
about input formula \uparrow

$$\begin{array}{c} * \\ \vdots \\ \Gamma_3 \vdash \Delta_3 \Downarrow ? \\ \hline \Gamma_2 \vdash \Delta_2 \Downarrow ? \\ \hline \Gamma_1 \vdash \Delta_1 \Downarrow ? \\ \vdots \\ \Gamma \vdash \Delta \Downarrow ? \end{array}$$

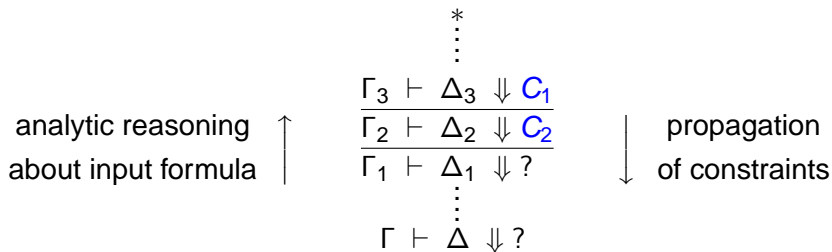
Anticipated way of proof construction



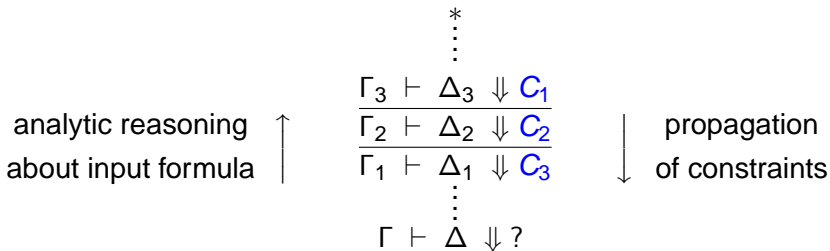
Anticipated way of proof construction



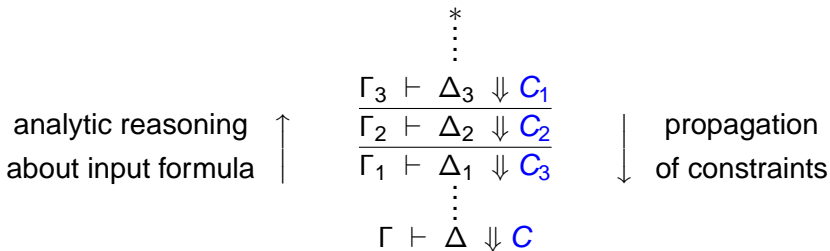
Anticipated way of proof construction



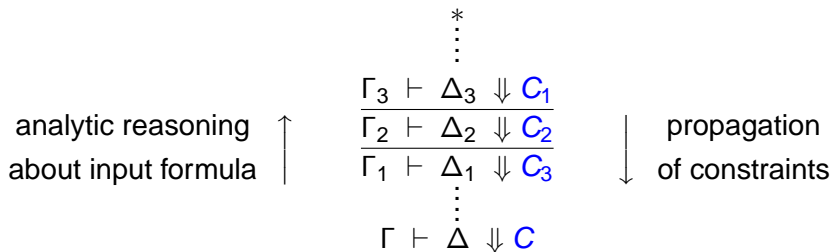
Anticipated way of proof construction



Anticipated way of proof construction



Anticipated way of proof construction



- Constraints are simplified during propagation
- If C is valid, then so is $\Gamma \vdash \Delta$
- If C is satisfiable, it describes a “solution” for $\Gamma \vdash \Delta$
- If C is unsatisfiable, expand the proof tree further ...

FOL rules on constrained sequents

$$\frac{\Gamma \vdash \phi, \Delta \Downarrow C \quad \Gamma \vdash \psi, \Delta \Downarrow D}{\Gamma \vdash \phi \wedge \psi, \Delta \Downarrow C \wedge D} \text{ AND-RIGHT}$$

$$\frac{\Gamma, \phi \vdash \Delta \Downarrow C \quad \Gamma, \psi \vdash \Delta \Downarrow D}{\Gamma, \phi \vee \psi \vdash \Delta \Downarrow C \wedge D} \text{ OR-LEFT}$$

$$\frac{\Gamma, \phi, \psi \vdash \Delta \Downarrow C}{\Gamma, \phi \wedge \psi \vdash \Delta \Downarrow C} \text{ AND-LEFT}$$

$$\frac{\Gamma \vdash \phi, \psi, \Delta \Downarrow C}{\Gamma \vdash \phi \vee \psi, \Delta \Downarrow C} \text{ OR-RIGHT}$$

$$\frac{\Gamma \vdash \phi, \Delta \Downarrow C}{\Gamma, \neg\phi \vdash \Delta \Downarrow C} \text{ NOT-LEFT}$$

$$\frac{\Gamma, \phi \vdash \Delta \Downarrow C}{\Gamma \vdash \neg\phi, \Delta \Downarrow C} \text{ NOT-RIGHT}$$

$$\frac{\Gamma \vdash [x/c]\phi, \exists x.\phi, \Delta \Downarrow [x/c]C}{\Gamma \vdash \exists x.\phi, \Delta \Downarrow \exists x.C} \text{ EX-RIGHT}$$

$$\frac{\Gamma, [x/c]\phi, \forall x.\phi \vdash \Delta \Downarrow [x/c]C}{\Gamma, \forall x.\phi \vdash \Delta \Downarrow \exists x.C} \text{ ALL-LEFT}$$

$$\frac{\Gamma \vdash [x/c]\phi, \Delta \Downarrow [x/c]C}{\Gamma \vdash \forall x.\phi, \Delta \Downarrow \forall x.C} \text{ ALL-RIGHT}$$

$$\frac{\Gamma, [x/c]\phi \vdash \Delta \Downarrow [x/c]C}{\Gamma, \exists x.\phi \vdash \Delta \Downarrow \forall x.C} \text{ EX-LEFT}$$

Closure rules on constrained sequents

$$\frac{\Gamma, p(s_1, \dots, s_n) \vdash p(t_1, \dots, t_n), \bigwedge_i s_i - t_i \doteq 0, \Delta \Downarrow C}{\Gamma, p(s_1, \dots, s_n) \vdash p(t_1, \dots, t_n), \Delta \Downarrow C} \text{ PRED-UNIFY}$$

$$\frac{\Gamma, \phi_1, \dots, \phi_n \vdash \psi_1, \dots, \psi_m, \Delta \Downarrow \neg\phi_1 \vee \dots \vee \neg\phi_n \vee \psi_1 \vee \dots \vee \psi_m}{\Gamma, \phi_1, \dots, \phi_n \vdash \psi_1, \dots, \psi_m, \Delta \Downarrow \neg\phi_1 \vee \dots \vee \neg\phi_n \vee \psi_1 \vee \dots \vee \psi_m}^* \text{ CLOSE}$$

- Side-condition: CLOSE is only applied to predicate-free formulae
⇒ Constraints are PA formulae

Lemma (Completeness for FOL)

If ϕ is a theorem of FOL, then there is a valid PA formula C such that $\vdash \phi \Downarrow C$ is provable.

Lemma (Fair Proof Construction for FOL)

*If ϕ is a theorem of FOL, then fair application of rules eventually leads to a closed proof with valid constraint.
(Special handling of rule CLOSE is necessary).*

Adding Linear Integer Arithmetic

Rules for integer arithmetic

One possibility: move integer handling into constraints

- In principle: any (external) PA procedure could be used to decide constraints

Built-in PA rules seem more clever, however:

- Eager simplification of equations, inequalities to prune search space
- Ground problems \rightarrow no constraints are necessary

PA rules shown here correspond to Omega:

- Equations are solved and eliminated
- Fourier-Motzkin + case analysis to handle inequalities

Rules for equations

$$\frac{\Gamma, t \doteq 0 \vdash \phi[\mathbf{s} + \alpha \cdot \mathbf{t}], \Delta \Downarrow \mathbf{C}}{\Gamma, t \doteq 0 \vdash \phi[\mathbf{s}], \Delta \Downarrow \mathbf{C}} \text{ RED}$$

($t \doteq 0$ and $\phi[\mathbf{s}]$ are different formulae)

$$\frac{\Gamma, \alpha(\mathbf{u} + \mathbf{c}') + \mathbf{t} \doteq 0, \mathbf{c} - \mathbf{u} - \mathbf{c}' \doteq 0 \vdash \Delta \Downarrow [\mathbf{x}/\mathbf{c}']\mathbf{C}}{\Gamma, \alpha\mathbf{c} + \mathbf{t} \doteq 0 \vdash \Delta \Downarrow \forall \mathbf{x}.\mathbf{C}} \text{ COL-RED}$$

(\mathbf{c}' a constant that does not occur in the conclusion or in \mathbf{u})

$$\frac{\Gamma, \alpha(\mathbf{u} + \mathbf{c}') + \mathbf{t} \doteq 0, \mathbf{c} - \mathbf{u} - \mathbf{c}' \doteq 0 \vdash \Delta \Downarrow [\mathbf{x}/\mathbf{c}']\mathbf{C}}{\Gamma, \alpha\mathbf{c} + \mathbf{t} \doteq 0 \vdash \Delta \Downarrow [\mathbf{x}/\mathbf{c} - \mathbf{u}]\mathbf{C}} \text{ COL-RED-SUBST}$$

(\mathbf{c}' a constant that does not occur in the conclusion or in \mathbf{u})

Rules for divisibility

$$\frac{\Gamma, \exists x. \alpha x + t \doteq 0 \vdash \Delta \Downarrow C}{\Gamma, \alpha \mid t \vdash \Delta \Downarrow C} \text{ DIV-LEFT}$$

(x a variable that does not occur in the conclusion)

$$\frac{\Gamma, (\alpha \mid t + 1) \vee \dots \vee (\alpha \mid t + \alpha - 1) \vdash \Delta \Downarrow C}{\Gamma \vdash \alpha \mid t, \Delta \Downarrow C} \text{ DIV-RIGHT } (\alpha > 0)$$

$$\frac{\Gamma, \alpha c - t \doteq 0 \vdash \Delta \Downarrow C}{\Gamma, \alpha c - t \doteq 0 \vdash \Delta \Downarrow [x/t]C' \vee \alpha \nmid t} \text{ DIV-CLOSE}$$

(c does not occur in t or in C' , C' a PA formula such that $C \Leftrightarrow [x/\alpha c]C'$)

Rules for inequalities

$$\frac{\Gamma \vdash t \dot{\leq} 0, \Delta \Downarrow C \quad \Gamma \vdash t \dot{\geq} 0, \Delta \Downarrow D}{\Gamma \vdash t \dot{=} 0, \Delta \Downarrow C \wedge D} \text{ SPLIT-EQ}$$

$$\frac{\Gamma, t \dot{=} 0 \vdash \Delta \Downarrow C}{\Gamma, t \dot{\leq} 0, t \dot{\geq} 0 \vdash \Delta \Downarrow C} \text{ ANTI-SYMM}$$

$$\frac{\Gamma, \alpha c + s \dot{\geq} 0, \beta c + t \dot{\leq} 0, \beta s - \alpha t \dot{\geq} 0 \vdash \Delta \Downarrow C}{\Gamma, \alpha c + s \dot{\geq} 0, \beta c + t \dot{\leq} 0 \vdash \Delta \Downarrow C} \text{ FM-ELIM}$$

$(\alpha > 0, \beta > 0)$

$$\frac{\Gamma, \bigwedge_{i,j} \alpha_i b_j - a_i \beta_j - (\alpha_i - 1)(\beta_j - 1) \dot{\geq} 0 \quad \vee \quad \bigvee_i \bigvee_{k=0}^{m_i} \left(\begin{array}{l} \alpha_i c - a_i - k \dot{=} 0 \wedge \\ \bigwedge_i \alpha_i c - a_i \dot{\geq} 0 \wedge \bigwedge_j \beta_j c - b_j \dot{\leq} 0 \end{array} \right) \vdash \Delta \Downarrow C}{\Gamma, \{\alpha_i c - a_i \dot{\geq} 0\}_i, \{\beta_j c - b_j \dot{\leq} 0\}_j \vdash \Delta \Downarrow C} \text{ OMEGA-ELIM}$$

$(\alpha_i > 0, \beta_j > 0)$

Lemma

There is an application strategy for the PA rules such that:

- *application of rules to a PA formula ϕ terminates,*
- *the produced constraint C is equivalent to ϕ , and*
- *if ϕ only contains existential quantifiers, then C is ground.*

- PA calculus eliminates quantifiers
- Quantifiers in constraints \Rightarrow recursive application

Lemma

There is an application strategy for the PA rules such that:

- *application of rules to a PA formula ϕ terminates,*
- *the produced constraint C is equivalent to ϕ , and*
- *if ϕ only contains existential quantifiers, then C is ground.*

- PA calculus eliminates quantifiers
- Quantifiers in constraints \Rightarrow recursive application

DEMO

Further completeness-results

Lemma (Existential formulae (ground formulae))

If ϕ is an unsatisfiable formula that only contains existential quantifiers, then there is a valid constraint C such that $\phi \vdash \Downarrow C$ is provable.

Lemma (Universal formulae)

If ϕ is an unsatisfiable formula that only contains universal quantifiers, then there is a valid constraint C such that $\phi \vdash \Downarrow C$ is provable.

Further completeness-results (2)

Lemma (Universal formulae with finite parametrisation)

Suppose $\exists \bar{a}.(\phi \wedge \psi)$ is an unsatisfiable formula, where:

- ϕ is a PA formula over \bar{a} that only has finitely many solutions, and
- ψ is an arbitrary formula over \bar{a} that only contains universal quantifiers.

Then there is a valid constraint C such that $\exists \bar{a}.(\phi \wedge \psi) \vdash \downarrow C$ is provable.

\Rightarrow These are the formulae handled by $\mathcal{ME}(\text{LIA})$

- $\mathcal{ME}(\text{LIA})$: model evolution modulo linear integer arithmetic, [Baumgartner, Tinelli, Fuchs, 08]
- Various approaches to integrate theories in saturation calculi, e.g. [Stickel, JAR'85], [Bürchert, CADE'90], [Korovin, Voronkov, CSL'07], [Prevosto, Waldmann, ESCoR'06]
- Various SMT-solvers

Conclusion, Future work

- Combination of different techniques:
SMT-like ground reasoning, tableau-like free-variable reasoning, quantifier elimination
- Comparatively strong completeness properties
- The shown calculus is still very “unrefined”
⇒ Refinements to make it practically usable necessary

- Continue implementation . . .
- Model construction?
- Add missing result: fair application strategy is complete
- Investigate connection conditions
(in particular, hypertableau strategy)
- Further investigate connection to SMT-calculi
- Direct support for function symbols?

Thanks for your attention!

More information:

<http://www.cs.chalmers.se/~philipp/princess/>