

Theorema—Ultra—Omega Workshop, Saarbrücken, 2005

Finding Witness Terms in ETRCF by Quantifier Elimination Techniques

(Robert Vajda, RISC, Linz)

rvajda@risc.uni-linz.ac.at

■ Outline of the Talk

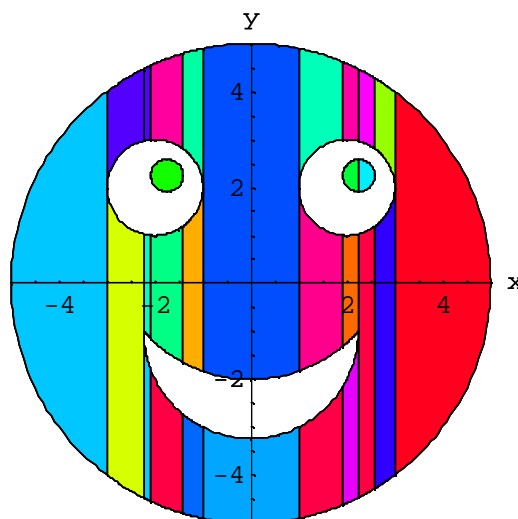
Specification of the Problem

Motivation

Description of the Method

Example

Conclusion and Future Work



■ Specification of the problem

Find an instantiation of variables (a collection of witness terms), which satisfies domain-specific constraints.

E.g., in order to prove the formula

$$(1) \quad \exists_{y \in \mathbb{R}} \forall_{x \in \mathbb{R}} x * (y - 1) = x ,$$

the only proper instantiation of the existentially quantified variable is:

$$y \leftarrow 2 .$$

In [Zimmer-Melis 2004], it is emphasized that

- instantiation is a hard task not only for the automated provers, but also for the humans,
- unification is not the appropriate instantiation technique, when the constraints are specific to a domain,
- instead of unification, constraint solvers can be used.

In this presentation we deal with real algebraic constraints, i.e. the constraints are expressible in the **E**lementary **T**heory of **R**eal **C**losed **F**ields (ETRCF), see e.g. [Winkler 1996]

$$\left(\text{CS} : 0, 1 ; \text{FS} : +, * ; \text{PS} : <, \leq, >, \geq, =, \neq \right)$$

Recall that a typical formula in this theory looks as follows:

$$(2) \quad \exists_{x,y} (x^2 + y^2 < 4 \wedge y^2 - 2x + 2 < 0)$$

Quantifier elimination, decidability, extension of the language

■ Motivation

CreaComp Project (E-Schulung von **K**reativität und Problemlöse**k**ompetenz);

Educational Unit for Elementary Analysis (see W. Windsteiger's talk).

In the first phase of the exploration cycle, we focus in this educational unit on proving properties of particular functions.

Typical properties which are investigated in the unit —expressed in *Theorema* syntax:

$$\begin{array}{ll}
 \text{FromAAndBBoundedSequence}[f] & \Leftrightarrow \exists k, K \in \mathbb{R} \forall n \in \mathbb{N} \quad (k < f[n] \wedge f[n] < K) \\
 \text{BoundedSequence}[f] & \Leftrightarrow \exists K \in \mathbb{R} \forall n \in \mathbb{N} \quad (|f[n]| < K) \\
 \text{ConvergentSequence}[f] & \Leftrightarrow \exists a \in \mathbb{R} \forall \epsilon \in \mathbb{R} \epsilon > 0 \exists N \in \mathbb{N} \forall n \in \mathbb{N} n \geq N \quad (|f[n] - a| < \epsilon) \\
 \text{LocalContinuousFunction}[f, a] & \Leftrightarrow \forall \epsilon \in \mathbb{R} \epsilon > 0 \exists \delta \in \mathbb{R} \delta > 0 \forall y \in \mathbb{R} |y - a| < \delta \quad (|f[y] - f[a]| < \epsilon) \\
 \text{UniformContinuousFunction}[f] & \Leftrightarrow \forall \epsilon \in \mathbb{R} \epsilon > 0 \exists \delta \in \mathbb{R} \delta > 0 \forall x, y \in \mathbb{R} |y - x| < \delta \quad (|f[y] - f[x]| < \epsilon)
 \end{array}$$

■ Method

The characteristic property of these notions is that their defining formula has *a sequence of alternating quantifiers*

(\vec{x}_i, \vec{y}_j denote a finite sequence of variables with length ≥ 0 , the matrix \mathbf{P} is quantifier-free).

$$\left(\begin{array}{c} \forall \exists \forall \exists \forall \exists \dots \\ \vec{x}_1 \vec{y}_1 \vec{x}_2 \vec{y}_2 \vec{x}_3 \vec{y}_3 \end{array} \right) \mathbf{P} [\vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2, \dots]$$

– We use the PCS method introduced by Buchberger [Buchberger 2001] to prove propositions belonging to this domain.

PCS : Combination of **P**roving, **S**ymbolic **C**omputing and **S**olving phases

– first implementation by Dupre [Dupre 2000]

– one crucial step in those proofs is to construct witness terms, typically constants or unary or binary Skolem functions (Solve).

Improvement of the solving phase:

We use a variant of the quantifier elimination (QE) technique in ETRCF — which is based on Collins' Cylindrical Algebraic Decomposition Method (CAD) [Collins 1975]

We emphasize, that proper pre-processing is needed to get an instantiation of an existentially quantified variable of the goal formula in the proof situation; use the constraint solver as an *oracle*, which gives a good guess for the witness term.

■ Interface to the Solver

- Preliminary step: Partition the repeated alternating quantifier sequence in the prefix of the goal formula

Round 1:

- Eliminate (pre-process) in one step (only) the first segment of the quantifiers,

$$\left(\begin{array}{c} \forall \exists \\ \vec{x}_1 \vec{y}_1 \end{array} \right) \left(\begin{array}{c} \forall \exists \\ \vec{x}_2 \vec{y}_2 \end{array} \right) P[\vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_1] \gg$$

$$\left(\begin{array}{c} \forall \exists \\ \vec{x}_2 \vec{y}_2 \end{array} \right) P[\mathbf{x}_{11_0}, \dots, \mathbf{x}_{1m_0}, \mathbf{y}_{11^*}, \dots, \mathbf{y}_{1n^*}, \vec{x}_2, \vec{y}_2]$$

where universally quantified variables eliminated by "the arbitrary but fixed"- inference rule and existentially quantified variables by introducing metavariables:

$$(\mathbf{x}_{11} \rightarrow \mathbf{x}_{11_0}, \mathbf{x}_{12} \rightarrow \mathbf{x}_{12_0}, \dots, \mathbf{x}_{1m} \rightarrow \mathbf{x}_{1m_0}, \mathbf{y}_{11} \rightarrow \mathbf{y}_{11^*}, \mathbf{y}_{12} \rightarrow \mathbf{y}_{12^*}, \dots, \mathbf{y}_{1n} \rightarrow \mathbf{y}_{1n^*})$$

- Use QE as an oracle: $\mathbf{y}_{11^*} \leftarrow \mathbf{f}_{11}[\mathbf{x}_{11_0}, \dots, \mathbf{x}_{1m_0}], \dots, \mathbf{y}_{1n^*} \leftarrow \mathbf{f}_{1n}[\mathbf{x}_{11_0}, \dots, \mathbf{x}_{1m_0}]$
(appropriate post-processing of the output must be done to get $\mathbf{f}_{11}, \dots, \mathbf{f}_{1n}$)

After the first step we remain with

$$\left(\begin{array}{c} \forall \exists \\ \vec{x}_2 \vec{y}_2 \end{array} \right) P[\mathbf{x}_{11_0}, \dots, \mathbf{x}_{1m_0}, \mathbf{f}_{11}[\mathbf{x}_{11_0}, \dots, \mathbf{x}_{1m_0}],$$

$$\dots, \mathbf{f}_{1n}[\mathbf{x}_{11_0}, \dots, \mathbf{x}_{1m_0}], \vec{x}_2, \vec{y}_2]$$

Since Theorema is implemented on top of *Mathematica*, it is convenient for us to exploit the available *Mathematica* algorithms like

- **CylindricalDecomposition** (New in 4.x, experimental package, Standard package from 5.x)
- **Resolve** (New in 5.x)

■ **Example:** $s1 : n \rightarrow \frac{2}{n^2 + 3 * n}$ is convergent

Definition

$$s1[n] := \frac{2}{n^2 + 3 * n}$$

$$\text{ConvergentSequence}[f] \Leftrightarrow \exists_{a \in \mathbb{R}} \forall_{\epsilon \in \mathbb{R}, \epsilon > 0} \exists_{N \in \mathbb{N}} \forall_{n \in \mathbb{N}, n \geq N} |f[n] - a| < \epsilon$$

Proposition

$$\text{ConvergentSequence}[s1]$$

Proof

Step 1: Partition the prefix (and an additional RW step)

$$\exists_{a \in \mathbb{R}} \forall_{\epsilon \in \mathbb{R}} \exists_{N \in \mathbb{N}} \forall_{n \in \mathbb{N}} \left\{ \text{seq01}[n] - a \right\} < \epsilon$$

$\epsilon > 0$ $n \geq N$

→

$$\left(\exists_{a \in \mathbb{R}} \right) \left(\forall_{\substack{\epsilon \in \mathbb{R} \\ \epsilon > 0}} \exists_{N \in \mathbb{N}} \right) \left(\forall_{\substack{n \in \mathbb{N} \\ n \geq N}} \right) \left| \frac{2}{n^2 + 3 * n} - a \right| < \epsilon$$

Step 2: pre-process the first segment of the prefix by introducing a metavariable ($a \rightarrow a^*$)

$$\left(\begin{array}{c} \exists \\ a \in \mathbb{R} \end{array} \right) \left(\begin{array}{cc} \forall & \exists \\ \epsilon \in \mathbb{R} & N \in \mathbb{N} \\ \epsilon > 0 & \end{array} \right) \left(\begin{array}{c} \forall \\ n \in \mathbb{N} \\ n \geq N \end{array} \right) \left| \frac{2}{n^2 + 3 * n} - a \right| < \epsilon \rightarrow$$

$$\left(\begin{array}{cc} \forall & \exists \\ \epsilon \in \mathbb{R} & N \in \mathbb{N} \\ \epsilon > 0 & \end{array} \right) \left(\begin{array}{c} \forall \\ n \in \mathbb{N} \\ n \geq N \end{array} \right) \left| \frac{2}{n^2 + 3 * n} - a^* \right| < \epsilon$$

Step 3: Call the oracle (QE)

$$\text{In[10]:= Resolve} \left[\forall_{\epsilon, \epsilon > 0} \exists_{N, N \geq 1} \forall_{n, n \geq 1} \left((n \geq N) \Rightarrow \text{Abs} \left[\frac{2}{n^2 + 3 * n} - a^* \right] < \epsilon \right), \{a^*\}, \text{Reals} \right]$$

$$\text{Out[10]= } a^* == 0$$

The oracle returns the instantiation $a^* \leftarrow 0$, so our goal would be implied by

$$\left(\begin{array}{cc} \forall & \exists \\ \epsilon \in \mathbb{R} & N \in \mathbb{N} \\ \epsilon > 0 & \end{array} \right) \left(\begin{array}{c} \forall \\ n \in \mathbb{N} \\ n \geq N \end{array} \right) \left| \frac{2}{n^2 + 3 * n} - 0 \right| < \epsilon$$

Step 4: Pre-process the second segment of the prefix ($\epsilon \rightarrow \epsilon_0, N \rightarrow N^*$)

$$\left(\begin{array}{l} \forall \\ \epsilon \in \mathbb{R} \\ \epsilon > 0 \end{array} \exists \begin{array}{l} N \in \mathbb{N} \\ n \in \mathbb{N} \\ n \geq N \end{array} \right) \left| \frac{2}{n^2 + 3n} - 0 \right| < \epsilon \rightarrow$$

$$\left(\forall_{n \in \mathbb{N}} \right) (n \geq N^*) \Rightarrow \left| \frac{2}{n^2 + 3n} - 0 \right| < \epsilon_0$$

Step 5: Call the oracle (QE) once more, since a universal quantifier is still involved in the remaining formula

In[15]=

Resolve [

$$\forall_{n, n \geq 1} \left(n \geq N^* \Rightarrow \left(\text{Abs} \left[\frac{2}{n^2 + 3n} - 0 \right] < \epsilon_0 \right) \right) \wedge \epsilon_0 > 0 \wedge N^* \geq 1, \{ \epsilon_0, N^* \}, \text{Reals}]$$

$$\text{Out[15]=} \left(0 < \epsilon_0 \leq \frac{1}{2} \ \&\& \ N^* > -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8+9\epsilon_0}{\epsilon_0}} \right) \ || \ \left(\epsilon_0 > \frac{1}{2} \ \&\& \ N^* \geq 1 \right)$$

(The oracle checks the projection condition and since we know that the crucial term $-\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8+9\epsilon_0}{\epsilon_0}}$ is nonnegative for all positive ϵ , the oracle simplifies the term (i.e avoids disjunctive branching)

In[5]=

CylindricalDecomposition[$\epsilon_0 > 0 \Rightarrow (0 < \epsilon_0 \leq 1/2 \vee \epsilon_0 > 1/2), \{ \epsilon_0 \}$]

Out[5]= True

In[4]= **CylindricalDecomposition**[$\epsilon_0 > 0 \Rightarrow -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8+9*\epsilon_0}{\epsilon_0}} \geq 0, \{ \epsilon_0 \}$]

Out[4]= True

and it yields finally, that, by choosing $N^* \left\lceil -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8+9\epsilon_0}{\epsilon_0}} + 1 \right\rceil$, the goal would be implied by

$$\left(\forall_{n \in \mathbb{N}} \right) \left(n \geq \left\lceil -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8+9*\epsilon_0}{\epsilon_0}} + 1 \right\rceil \Rightarrow \left| \frac{2}{n^2 + 3n} - 0 \right| < \epsilon_0 \right)$$

Step 6-7: Pre-process the last segment of the prefix ($n \rightarrow n_0$), since the remaining formula is a quantifier- and meta-free, use a constraint checker to validate the goal.

$$\left(\forall_{n \in \mathbb{N}} \left(n \geq \left\lfloor -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8 + 9 * \epsilon_0}{\epsilon_0}} + 1 \right\rfloor \Rightarrow \left| \frac{2}{n^2 + 3 * n} - 0 \right| < \epsilon_0 \right) \right) \rightarrow$$

$$n_0 \geq \left\lfloor -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8 + 9 * \epsilon_0}{\epsilon_0}} + 1 \right\rfloor \Rightarrow \left| \frac{2}{n_0^2 + 3 * n_0} - 0 \right| < \epsilon_0$$

In[20]:=

Experimental`ForAllRealQ[($\epsilon_0 > 0 \wedge n_0 \geq 1$) \Rightarrow

$$\left(n_0 \geq -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8 + 9 * \epsilon_0}{\epsilon_0}} + 1 \Rightarrow \text{Abs} \left[\frac{2}{n_0^2 + 3 * n_0} - 0 \right] < \epsilon_0 \right), \{ \epsilon_0, n_0 \}]$$

Out[20]= True

Out[20]=

$$\{ \text{True}, a^* \leftarrow 0, N^* \leftarrow \left\lfloor -\frac{3}{2} + \frac{1}{2} \sqrt{\frac{8 + 9 * \epsilon_0}{\epsilon_0}} + 1 \right\rfloor \}$$

⌞

⌞

11 of 13

Theorema Session —Prover in Action**Init****Theorema Database, Proposition, Calls**

TS_In[5]:=

$$\text{Definition}["s1", \text{any}[n], \text{seq01}[n] = \frac{2}{n^2 + 3 * n}]$$
TS_In[7]:= **Definition**["convergent", any[f], with[IsSeq[f]],
$$\text{Convergent}[f] \Leftrightarrow \exists_{\substack{a \in \mathbb{R} \\ \epsilon > 0}} \forall_{\substack{\epsilon \in \mathbb{R} \\ \epsilon > 0}} \exists_{\substack{N \in \mathbb{N} \\ n \geq N}} \forall_{\substack{n \in \mathbb{N} \\ n \geq N}} (|f[n] - a| < \epsilon)$$

TS_In[10]:=

Proposition["conv1", Convergent[seq01]]TS_In[11]:= **Prove**[**Proposition**["conv1"],

using → {**Definition**["convergent"], **Definition**["s1"], **Lemma**["s1-t"]},
TransformBy → **ProofSimplifier**, **by** → **UserInequl**,
TransformerOptions → {**branches** → **Proved**, **steps** → **Useful**}

TS_Out[11]= - ProofObject -

■ Conclusion and Future Work

- We have shown a possible way for exploiting quantifier elimination algorithms available in computer algebra systems for the purpose of finding witness terms
- In computer supported mathematical education we found promising application areas, where this domain-specific technique can be utilized
- A more careful and exhaustive review of the related available algorithms in Mathematica 5.1 seems to blur the traditional strict distinction between computer algebra and automatic theorem proving systems (see e.g. [Harrison 1998])

```

CylindricalDecomposition
Resolve
Reduce
FindInstance
ForAllRealQ
ExistsRealQ
ImpliesQ
...

```

- Further work is needed when the proof situation is only simplified by the real constraint solver, and we have to switch to another phase or we have to call another domain-specific constraint-checker.

$$\exists_{x \in \mathbb{R}} \left(x * x = x + x \wedge x \in \{y \mid \sin[y] > 0\} \right) \gg$$

$$\left(0 \in \{y \mid \sin[y] > 0\} \vee 2 \in \{y \mid \sin[y] > 0\} \right)$$

(Proving phase, disjunction in the goal)

We assume that

$$\neg \left(0 \in \{y \mid \sin[y] > 0\} \right)$$

and prove

$$2 \in \{y \mid \sin[y] > 0\}$$

...

■ References

- B. Buchberger: The PCS Prover in Theorema. In: LNCS 2178, pp. 469-478, Springer, Berlin, 2001.
- G. E. Collins: Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition. In: LNCS 33, pp. 134-183, 1975.
- D. Dupre: Automated Theorem Proving by Integrating Proving, Solving and Computing. Ph.D. Thesis, University of Linz, 2000.
- J. Harrison: Theorem Proving with the Real Numbers. Springer, London, 1998.
- E. Melis: Why Proof Planning for Maths Education and How? In: Mechanizing Mathematical Reasoning, pp. 364-378, 2005.
- S. Saminger -- R. Vajda -- W. Windsteiger: Using a Computer Algebra System and a Theorem Prover to Stimulate Creativity in Learning Mathematics. Linz, 2005 (preprint)
- F. Winkler: Polynomial Algorithms in Computer Algebra. Texts and Monographs in Symbolic Computation, Springer, Wien - New York, 1996.
- J. Zimmer -- E. Melis: Constraint Solving for Proof Planning. In: JAR 33(1), pp.51-88, 2004.