Adapting Cylindrical Algebraic Decomposition for Proof Specific Tasks

Erika Abraham[1] and Tudor Jebelean[2]

[1] RWTH Aachen University, Germany
[2] RISC-Linz, Johannes Kepler University, Austria

Abstract. We develop a case study on using quantifier elimination and cylindrical algebraic decomposition for the purpose of finding specific terms for the automation of proving mathematical properties in elementary analysis.
Real–life proofs in specific mathematical domains are difficult to automate because of the high number of assumptions necessary for the prover to succeed. In particular, in elementary analysis, it is allmost impossible to find automatically the appropriate terms for the instantiation of universal assumptions and for witnessing the existential goals. We aim at developing such proofs in natural style, in the frame of the *Theorema* system.
Finding such terms is actually possible by using quantifier elimination (QE) based on cylindrical algebraic decomposition (CAD). However, the current straightforward approach lacks in efficiency, because several redundant calls may be necessary, and because the nature of the problem is slightly different.
We study some natural–style proofs, the necessary special terms, and the corresponding usage of the QE/CAD, and identify specific techniques for adapting these algorithms in order to increase their efficiency. The experiments are performed partially in *Mathematica* and partially in `SMT-RAT`, the latter used as "white–box", which allows to inspect the intermediate results and to adapt certain parts of the algorithms.

Keywords: Cylindrical Algebraic Decomposition, Automated Reasoning, Satisfiability Modulo Theories

MSC: 68T15, 68W30

1 Introduction

Cylindrical Algebraic Decomposition (CAD) [2] is a quite successfull, albeit expensive technique for quantifier elimination in the theory of real closed fields. Previous research [5] shows that this technique can be used for finding automatically witnesses in mechanical proofs for simple theorems in elementary analysis. Given a real-algebraic sentence whose truth we want to prove, CAD–based quantifier elimination is used to find for each existentially quantified variable a suitable term (witness) such that substituting that term for that variable leads to the satisfaction of the statement. Unfortunately, the approach [5] requires the invocation of the CAD procedure once for each existential quantifier block and can therefore be time–cost prohibitive, especially if the problems contain polynomials of higher degree.

The research reported here proposes improvements to the previous approach to increase the efficiency of finding witness terms. Instead of calling the CAD as a black box, our improvements are based on white-box CAD computations, where we make

use of influencing the decomposition process and extracting CAD-internal information from the decomposition which cannot be extracted from the standard black-box output of the CAD (as implemented in, e.g., *Mathematica*). The main ideas are (1) to apply only partial CAD computations that are necessary for finding proper witness terms for the existentially quantified variables and (2) to reduce the effort to computing a single CAD and exploit its internal representation to construct witnesses for all existentially quantified variables in one step. Our technique can also be applied to strengthen Satisfiability Modulo Theories (SMT) solvers, originally designed for deciding the satisfiability of quantifier-free logical problems, with abilities to provide symbolic solutions for satisfiable problem instances even for quantified real-algebraic problems.

2 Proofs in Natural Style

We illustrate our approach to natural–style[3] proving by an example. Consider the notion of *convergence of real sequences*, which is defined by the following definition with alternating quantifiers (f is a real function of natural argument, ϵ is real, m, n are naturals):

$$\text{IsConvergent}(f) \;\Leftrightarrow\; \exists_a\, \forall_{\epsilon>0}\, \exists_m\, \forall_{n\geq m}\, \text{Abs}[f(n)-a] < \epsilon$$

The natural–style proof of the statement: "The sum of two convergent sequences is convergent" is presented in the sequel. (For space reasons we do not exhibit the treatment of the type information about the variables – real functions of real value, real numbers, natural numbers – but this is easily handled by the same logical approach.)

Stage 1: The presentation of the proof begins by listing the *local assumptions* and the goal. The local assumptions are the formulae which are assumed only for the purpose of this proof. Other assumptions will also be necessary for the success of the proof, which typically express certain properties of the domains involved and are used in several proofs. It is important to note that the shape of the assumptions and of the goal are very similar, this is the basis for the specific proof strategy illustrated here. Namely, these two assumptions and the goal will be transformed in parallel, one quantifier at a time, until the quantifiers are removed.

We assume :

(1) $\exists_a \forall_\epsilon\, (\epsilon > 0 \Rightarrow \exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[f_1[n] - a] < \epsilon))$

(2) $\exists_a \forall_\epsilon\, (\epsilon > 0 \Rightarrow \exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[f_2[n] - a] < \epsilon))$

and we prove :

(3) $\exists_a \forall_\epsilon\, (\epsilon > 0 \Rightarrow \exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[(f_1[n] + f_2[n]) - a] < \epsilon))$

Stage 2: When the assumptions and the goal are existential formulae, the proof proceeds by first introducing Skolem constants (here a_1 and a_2) for the existential variables of the assumptions, and then by using a witness (which typically depends on the recent Skolem constants) for proving the existential goal. The crucial difficulty at the latter step consists in finding the witness (here $a_1 + a_2$).

[3] This is not the same as natural deduction.

By (1), (2) we can take a_1, a_2 such that :

(4) $\forall_\epsilon\,(\epsilon > 0 \Rightarrow \exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[f_1[n] - a_1] < \epsilon))$

(5) $\forall_\epsilon\,(\epsilon > 0 \Rightarrow \exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[f_2[n] - a_2] < \epsilon))$

For proving (3) it is sufficient to prove :

(6) $\forall_\epsilon\,(\epsilon > 0 \Rightarrow \exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[(f_1[n] + f_2[n]) - (a_1 + a_2)] < \epsilon))$

Stage 3: When the assumptions and the goal are universal formulae, the proof proceeds by first introducing a Skolem constant (here ϵ_0) for the universal variable of the goal (which is also assumed to satisfy the condition associated to the quantifier), and then by instantiating the assumptions with certain terms: The crucial difficulty at this step consists in finding the appropriate terms (here $\epsilon_0/2$). Note also that before the actual instantiation, one proves first that the respective terms satisfy the condition associated to the universal quantifier. (This is typically derived from the condition satisfied by the Skolem constant introduced before.)

For proving (6) we take ϵ_0 arbitrary but fixed, we assume :

(7) $\epsilon_0 > 0$

and we prove :

(8) $\exists_M \forall_n (n \geq M \Rightarrow \mathsf{Abs}[(f_1[n] + f_2[n]) - (a_1 + a_2)] < \epsilon_0)$

First we prove :

(9) $\frac{\epsilon_0}{2} > 0$

This follows from (7) (using elementary properties of $\mathbb{R}$).
Using (9), from (4) and (5) we obtain :

(10) $\exists_M \forall_n \left(n \geq M \Rightarrow \mathsf{Abs}[f_1[n] - a_1] < \frac{\epsilon_0}{2}\right)$

(11) $\exists_M \forall_n \left(n \geq M \Rightarrow \mathsf{Abs}[f_2[n] - a_2] < \frac{\epsilon_0}{2}\right)$

Stage 4: Here again we see an illustration of the strategy in case of parallel existential formulae. The difficulty consists in finding the witness $\mathrm{Max}[M_1, M_2]$.

By (10) and (11) we can take M_1, M_2 such that :

(12) $\forall_n \left(n \geq M_1 \Rightarrow \mathsf{Abs}[f_1[n] - a_1] < \frac{\epsilon_0}{2}\right)$

(13) $\forall_n \left(n \geq M_2 \Rightarrow \mathsf{Abs}[f_2[n] - a_2] < \frac{\epsilon_0}{2}\right)$

In order to prove (8) it suffices to prove:

(14) $\forall_n (n \geq \mathsf{Max}[M_1, M_2] \Rightarrow \mathsf{Abs}[(f_1[n] + f_2[n]) - (a_1 + a_2)] < \epsilon_0)$

Stage 5: Here again we have a situation with universal quantifiers, and the choice of the instantiation terms is n_0. Note that, analoguous to the previous similar situation, a sub–proof is needed for the fact that the instantiation terms satisfy the respective conditions.

For proving (14) we take n_0 arbitrary but fixed, we assume :

(15) $n_0 \geq \mathsf{Max}[M_1, M_2]$

and we prove :

(16) $\mathsf{Abs}[(f_1[n_0] + f_2[n_0]) - (a_1 + a_2)] < \epsilon_0$

First we prove :

(17) $(n_0 \geq M_1) \wedge (n_0 \geq M_2)$

This follows from (15) (using elementary properties of $\mathbb{R}$).
Using (17), from (12) and (13) we obtain :

(18) $\mathsf{Abs}[f_1[n_0] - a_1] < \frac{\epsilon_0}{2}$

(19) $\mathsf{Abs}[f_2[n_0] - a_2] < \frac{\epsilon_0}{2}$

Stage 6: The next two stages apply a simple but effective heuristics of replacing equal terms by constants. First $f_1[n_0]$ is replaced by x_1 and similarly for f_2.

We replace the constant terms $f_1[n_0], f_2[n_0]$ by the constants x_1, x_2, respectively.
The assumptions (18), (19) become :

(20) $\mathsf{Abs}[x_1 - a_1] < \frac{\epsilon_0}{2}$

(21) $\mathsf{Abs}[x_2 - a_2] < \frac{\epsilon_0}{2}$

The goal (16) becomes :

(22) $\mathsf{Abs}[(x_1 + x_2) - (a_1 + a_2)] < \epsilon_0$

Stage 7: Now $x_1 - a_1$ is replaced by y_1 and similarly for $x_2 - a_2$.

We replace the constant terms $x_1 - a_1, x_2 - a_2$ by the constants y_1, y_2, respectively.
The assumptions (20), (21) become :

(23) $\mathsf{Abs}[y_1] < \frac{\epsilon_0}{2}$

(24) $\mathsf{Abs}[y_2] < \frac{\epsilon_0}{2}$

The goal (22) becomes (using elementary properties of $\mathbb{R}$) :

(25) $\mathsf{Abs}[y_1 + y_2] < \epsilon_0$

Stage 8: Finally the proof succeeds by some elementary manipulation of the assumptions and by applying a well–known property of the absolute value.

From (23), (24) we obtain (using elementary properties of $\mathbb{R}$) :

(26) $\mathsf{Abs}[y_1] + \mathsf{Abs}[y_2] < \epsilon_0$

By (26), for proving (25) (using elementary properties of $\mathbb{R}$) it suffices to prove :

(28) $\mathsf{Abs}[y_1 + y_2] \leq \mathsf{Abs}[y_1] + \mathsf{Abs}[y_2]$

This follows from elementary properties of $\mathbb{R}$.

3 The Use of Algebraic Techniques

Technically (in the background), the main goal reduces (by a proof technique in natural style described in [4]) to the formulae (1) (for the sequence arguments) and (3) (for the sequence values), which can be proven using QE/CAD.

3.1 Sequence Arguments

The proof of the formula:

$$\forall_{M_1,M_2}\exists_{M_0}\forall_n((n \geq M_0) \Rightarrow ((n \geq M_1) \wedge (n \geq M_2))) \tag{1}$$

can be performed in a "black-box" manner using the functions `Resolve` and `FullSimplify` of *Mathematica*, which return `True`. However, this does not help the production of the natural–style proof, because we do not have a witness for the existential variable M_0. In order to obtain a witness, one can call the same *Mathematica* functions on the formula:

$$\forall_n((n \geq M_0) \Rightarrow ((n \geq M_1) \wedge (n \geq M_2))) \tag{2}$$

This returns $(M_0 > M_1) \wedge (M_0 > M_2)$, which allows to construct a witness for M_0 as $\mathrm{Max}[M_1, M_2]$. However, this result can be extracted already from the first run of QE/CAD, with appropriate modifications of the algorithm.

We illustrate the procedure on this example. The projection phase starts with the set of polynomials corresponding to the boolean part of formula (1): $\{M_0 - n, M_1 - n, M_2 - n\}$. We apply projections in reverse order of the quantifiers. Projection step 1 (eliminate n) generates the set: $\{M_0 - M_1, M_0 - M_2\}$. Projection step 2 eliminates M_0: $\{M_1 - M_2\}$. Projection step 3 eliminates M_2: $\{1\}$.

The lifting phase starts now from the last projection: since the polynomial has no solutions, we choose sample value $M_1 = 0$, which is then substituted in the polynomial from projection step 2, giving the set $\{M_2\}$ which has solution 0. We choose for M_2 the sample values $\{-2, 0, 2\}$. Since M_2 is universal, we have to check all values.

- $M_2 = -2$: By substitution in the polynomials from step 1, we obtain $\{M_0, M_0 + 2\}$ with solutions $\{0, -2\}$. We choose sample values for $M_0 : \{-4, -2, -1, 0, 1\}$. Since M_0 is existential, it is enough to find one value which satisfies the boolean condition of (1).
  - $M_0 = -4$: By substitution in the original polynomials, we obtain $\{n+4, n, n+2\}$ with solutions $\{-4, -2, 0\}$. We choose sample values for $n : \{-5, -4, -3, -2, -1, 0, 1\}$. Since n is universal, all values must satisfy the boolean condition. The value -5 satisfies, but the value -4 does not, therefore we can stop this branch of lifting now,a and decide that the value $M_0 = -4$ is not appropriate. Note that we save a significant amount of computation here, because the values $n : \{-3, -2, -1, 0, 1\}$ do not need to be investigated.
  - $M_0 = -2, M_0 = -1$ fail in a similar manner.
  - $M_0 = 0$ succeeds, and since we need only one successful value, we do not have to test $M_0 = 1$, thus we save some amount of computation. Note that this saving is not possible when calling QE/CAD on the formula (2), because the information about the quantifiers is not present there.
    This successful lifting branch gives us the first set of satisfying values:
    $\{M_0 = 0, M_2 = -2, M_1 = 0\}$.

- $M_2 = 0$: In a similar way we obtain the second set: $\{M_0 = 0, M_2 = 0, M_1 = 0\}$.
- $M_2 = 2$: Similarly, the third set is: $\{M_0 = 0, M_2 = 2, M_1 = 2\}$.

Now we check the signs of the polynomials from projection step 1 (after eliminating n), which corresponds to the quantifier which is eliminated when working on formula (2)), and we obtain (by some easy simplification) from the three set of satisfying values, respectively: $\{M_1 > M_2 \wedge M_0 = M_1, M_1 = M_2 \wedge M_0 = M_1, M_1 < M_2 \wedge M_0 = M_2\}$, namely the fact that M_0 is the maximum of M_1 and M_2. One can see that the witness is found in one single application of the QE/CAD method on the original formula, and that a certain amount of computation is saved by speculative evaluation corresponding to the original quantifiers.

3.2 Sequence Values

The proof of the formula:
$\forall_{a_1,a_2}\exists_{a_0}\forall_{\epsilon_0}(\epsilon_0 > 0 \Rightarrow \exists_{\epsilon_1,\epsilon_2}(\epsilon_1 > 0 \wedge \epsilon_2 > 0 \wedge$

$$\forall_{x_1,x_2}(\mathrm{Abs}[x_1 - a_1] < \epsilon_1 \wedge \mathrm{Abs}[x_2 - a_2] < \epsilon_2 \Rightarrow \mathrm{Abs}[x_1 + x_2 - a_0] < \epsilon_0))) \quad (3)$$

can be performed in a "black-box" manner using the functions `Resolve` and `FullSimplify` of *Mathematica*, which return `True`. However, this does not help the production of the natural–style proof, because we do not have witnesses for the existential variables. In order to obtain witnesses for a_0, one can call the same *Mathematica* functions on the formula:
$\forall_{\epsilon_0}(\epsilon_0 > 0 \Rightarrow \exists_{\epsilon_1,\epsilon_2}(\epsilon_1 > 0 \wedge \epsilon_2 > 0 \wedge$

$$\forall_{x_1,x_2}(\mathrm{Abs}[x_1 - a_1] < \epsilon_1 \wedge \mathrm{Abs}[x_2 - a_2] < \epsilon_2 \Rightarrow \mathrm{Abs}[(x_1 + x_2) - a_0] < \epsilon_0))) \quad (4)$$

This returns $a_0 = a_1 + a_2$, which can now be substituted in the previous formula:
$\forall_{\epsilon_0}(\epsilon_0 > 0 \Rightarrow \exists_{\epsilon_1,\epsilon_2}(\epsilon_1 > 0 \wedge \epsilon_2 > 0 \wedge$

$$\forall_{x_1,x_2}(\mathrm{Abs}[x_1 - a_1] < \epsilon_1 \wedge \mathrm{Abs}[x_2 - a_2] < \epsilon_2 \Rightarrow \mathrm{Abs}[(x_1 + x_2) - (a_1 + a_2)] < \epsilon_0))) \quad (5)$$

Applying QE/CAD to this formula gives "True", however without witnesses for ϵ_1 and ϵ_2, therefore we construct the formula:
$\epsilon_0 > 0 \wedge \epsilon_1 > 0 \wedge \epsilon_2 > 0 \wedge$

$$\forall_{x_1,x_2}(\mathrm{Abs}[x_1 - a_1] < \epsilon_1 \wedge \mathrm{Abs}[x_2 - a_2] < \epsilon_2 \Rightarrow \mathrm{Abs}[(x_1 + x_2) - (a_1 + a_2)] < \epsilon_0) \quad (6)$$

A call of the same functions on this formula returns: $\epsilon_1 + \epsilon_2 \leq \epsilon_0$, which allows to find appropriate witnesses $\epsilon_0/2$ for ϵ_1 and ϵ_2.

Similarly to the previous formula (but much more complicated this time), one can obtain all witnesses in only one application of the QE/CAD method, by carefully exploiting the structure of the quantifiers and of the boolean formulae involved.

The performance of the current black–box approach to QE/CAD is even less efficient in case of non–linear problems. For the same kind of statement, in which we replace "sum" by product, the *Mathematica* implementation returns after few dozens of minutes, but the answer over more then one page does not allow to infer a suitable

witness. The corresponding formulae, which we present here as a challenge for QE/CAD implementations, are the following:
$\forall_{\epsilon_0}(\epsilon_0 > 0 \Rightarrow \exists_{\epsilon_1,\epsilon_2}(\epsilon_1 > 0 \wedge \epsilon_2 > 0 \wedge$

$$\forall_{x_1,x_2}(\mathrm{Abs}[x_1 - a_1] < \epsilon_1 \wedge \mathrm{Abs}[x_2 - a_2] < \epsilon_2 \Rightarrow \mathrm{Abs}[(x_1 * x_2) - a_0] < \epsilon_0))) \quad (7)$$

$$\epsilon_0 > 0 \wedge \epsilon_1 > 0 \wedge \epsilon_2 > 0 \wedge$$
$$\forall_{x_1,x_2}(\mathrm{Abs}[x_1 - a_1] < \epsilon_1 \wedge \mathrm{Abs}[x_2 - a_2] < \epsilon_2 \Rightarrow \mathrm{Abs}[(x_1 * x_2) - (a_1 * a_2)] < \epsilon_0) \quad (8)$$

4 Conclusion

We implement the approach in the frame of the *Theorema* system [1] developed at RISC-Linz and in the frame of the `SMT-RAT` system developed at RWTH Aachen [3]. This also allows to compare the efficiency of the *Mathematica* implementation to the efficiency of a custom implementation, and to demonstrate the possibility of using external algebraic tools in *Theorema*.

Our approach can be extended to other areas, because SMT problems over reals can be approached in the same way, since unsatisfiability can be detected only by finding the appropriate instantiations which lead to contradiction.

Further work includes the developement of methods for the completion of natural style proofs after the appropriate witnesses are found, and the application of the proposed technique in further areas. For example, a relevant application would be to compute a symbolic description of the input-output behaviour of communicating processes.

Acknowledgments. Partially supported by the project "Satisfiability Checking and Symbolic Computation" (H2020-FETOPN-2015-CSA 712689).

References

1. Buchberger, B., Jebelean, T., Kutsia, T., Maletzky, A., Windsteiger, W.: Theorema 2.0: Computer-Assisted Natural-Style Mathematics. JFR 9(1), 149–185 (2016)
2. Collins, G.E.: Quantier elimination for real closed fields by cylindrical algebraic decomposition. In: Automata Theory and Formal Languages. LNCS, vol. 33, pp. 134–183. Springer (1975)
3. Corzilius, F., Kremer, G., Junges, S., Schupp, S., Abraham, E.: `SMT-RAT`: An open source C++ toolbox for strategic and parallel SMT solving. In: Proc. of SAT'15. LNCS, vol. 9340, pp. 360–368. Springer (2015)
4. Jebelean, T., Buchberger, B., Kutsia, T., Popov, N., Schreiner, W., Windsteiger, W.: Automated Reasoning. In: et al., B.B. (ed.) Hagenberg Research, pp. 63–101. Springer (2009)
5. Vajda, R., Jebelean, T., Buchberger, B.: Combining Logical and Algebraic Techniques for Natural Style Proving in Elementary Analysis. Mathematics and Computers in Simulation 79(8), 2310–2316 (April 2009)