NASA/TM-20205010644

Improving Automated Strategies for Univariate Quantifier Elimination

Katherine Cordwell
Carnegie Mellon University, Pittsburgh, Pennsylvania

César A. Muñoz and Aaron M. Dutle
Langley Research Center, Hampton, Virginia

January 2021

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- Help desk contact information:

https://www.sti.nasa.gov/sti-contact-form/
and select the "General" help request type.

NASA/TM-20205010644

Improving Automated Strategies for Univariate Quantifier Elimination

Katherine Cordwell
Carnegie Mellon University, Pittsburgh, Pennsylvania

César A. Muñoz and Aaron M. Dutle
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

January 2021

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

This report discusses improved support for univariate quantifier elimination in the Prototype Verification System (PVS). Previously, PVS had three strategies for quantifier elimination—`hutch`, `tarski`, and `sturm`. Of these, only `hutch` is able to decide queries in any input format—`sturm` only works on queries regarding a single polynomial on an interval and `tarski` resolves queries in the universal existential fragment. This paper describes an extended version of `tarski`. The extension is accomplished by formally verifying a disjunctive normal form transformation in PVS and using `tarski` on each conjunctive clause. Additionally, a preprocessing step is added to the decision procedure underlying `tarski`. This preprocessing is designed to exploit properties of polynomial structure to quickly resolve queries that have certain formats. The preprocessing produces dramatic speedup when it succeeds in resolving a query, and seems to introduce negligible overhead when it does not resolve a query. Finally, testing reveals some ways to improve the `hutch` and `tarski` strategies.

1 Introduction

Quantifier elimination (QE) refers to the process of transforming a quantified formula into a logically equivalent quantifier-free formula. Although Tarski proved that quantifier elimination is decidable in 1951 [16], it was not until 1975 that George Collins developed the first practical algorithm for QE: cylindrical algebraic decomposition (CAD) [3]. In general, CAD is doubly exponential in the degree of the polynomials. Although many people have improved CAD over the years, development of QE methods continues to be an active area of research (see, e.g., [1, 4]).

QE is especially significant because queries involving real-valued polynomials often arise in formal proofs of safety-critical systems that interact with the physical environment, i.e., cyber-physical systems. For example, the polynomial constraints that arise in path planning and obstacle avoidance algorithms for autonomous systems are often amenable to QE techniques. The formal approach provides behavioral guarantees that supplement experimental testing, and these guarantees are crucially important when dealing with safety-critical systems. However, QE is often a significant computational bottleneck in the verification process.

This work focuses on improving support for quantifier elimination in the Prototype Verification System (PVS) [13]. Currently, PVS implements support for univariate QE in three *strategies*—`sturm` [12], `tarski` [10], and `hutch` [11]. At the heart of these strategies are formally verified decision procedures based on Sturm's and Tarski's theorems. Thus, the soundness of the strategies depends only on the soundness of the PVS internal logic [10].

Of the PVS strategies for QE, only `hutch` is able to work on arbitrary queries—`sturm` is primarily designed to test the satisfiability of a single polynomial within an interval, and `tarski` can only be used to test the satisfiability of formulas in the existential conjunctive fragment. This work extends `tarski` so that it is able to handle arbitrary queries by formally verifying a disjunctive normal form (DNF) transformation in PVS. In this paper, this extension is called `dnftarski`. However, the improved strategy replaced the current strategy `tarski` in the current release of the NASA PVS Library.[1] Further, because queries in the existential conjunctive fragment often contain polynomial structure that is amenable to preprocessing, a preprocessing step is added to the improved strategy. This step significantly speeds up queries when it succeeds and introduces minimal overhead when it fails.

This report is structured as follows. Section 2 discusses related work. Section 3 motivates the approach, describes the preprocessing, and explains the DNF construction. In Section 4, `dnftarski` is compared to `tarski` and `hutch` on existing benchmarks, and is then tested on new examples for a more targeted analysis of the modifications. The targeted analysis reveals some ways to improve `tarski` and `hutch`, which are also discussed in Section 4. Concluding remarks are made in Section 5, with future work discussed in 6. Appendices A, B, C, D, and E list the benchmarks used in this report.

2 Related Work

Many tools such as Mathematica [7], QEPCAD [4], Z3 [5], and REDLOG [6] provide substantial support for QE (including, in some cases, implementations of CAD). The tool RAHD [14] combines various methods for quantifier elimination and, among other things, makes use of polynomial structure in very deep and extensive ways. However, the support of these tools is unverified and may contain bugs. Using a tool like Mathematica or Z3 in a formal methods proof as a *trusted oracle* is undesirable, because the correctness of the proof remains predicated on the soundness of the oracle. Since theorem proving is challenging, sound support for QE is much more limited than unverified support.

In 2007, Mahboubi implemented CAD in CoQ [8], but no one has yet succeeded in formally verifying CAD. Cohen and Mahboubi formalized a procedure for multivariate QE in CoQ that is based on Tarski's Theorem [2]. However, this procedure is mainly implemented for theoretical interest, as a stepping stone towards a formalization of CAD rather than a practical quantifier elimination procedure. Other QE procedures have been implemented in Isabelle/HOL, HOL Light, and CoQ [10–12].

Disjunctive normal form and conjunctive normal form (CNF) transformations have been formally verified in other theorem provers. In particular, Seidl and Sickert formalized a DNF construction for linear temporal logic formulas in Isabelle/HOL [15], and Maric formalized a CNF construction as part of Isabelle/HOL's SAT solver library [9].

3 Approach

The approach is pictured in Figure 1. It involves the following steps: First, assume that every QE query has been transformed into an equivalent existential QE query. Then, as every Boolean formula is logically equivalent to a Boolean formula in *disjunctive normal form*, i.e., a disjunction of conjunctive clauses, translate the Boolean

[1] `https://github.com/nasa/pvslib`.

formula in the query into an equivalent DNF formula. Next, run `tarski` with preprocessing on each conjunctive clause. Finally, return "true" if any one of the conjunctive clauses is "true". Return "false" if they are all false. In particular, if `tarski` resolves any of the clauses, then "true" can be returned at that stage without considering the rest of the clauses. This procedure is sound since the authors have formally verified that the DNF transformation produces a logically equivalent formula and that the preprocessing is sound.

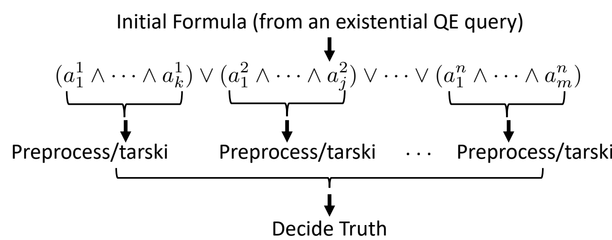

Figure 1. The approach for `dnftarski`

The technical details of this approach are discussed below.

3.1 DNF Transformation

The DNF transformation works as follows: First, every Boolean expression is put into *negation normal form* (NNF), so that negations only occur on polynomial relations (and not on Boolean combinations of polynomial relations). Then, conjunctions are recursively distributed over disjunctions, i.e., expressions of the form $(a \vee b) \wedge c$ and $c \wedge (a \vee b)$ are transformed into $(a \wedge c) \vee (b \wedge c)$ and $(c \wedge a) \vee (c \wedge b)$, respectively, until no such transformations are possible.

This transformation relies on a *deep embedding* of Boolean expressions, i.e., an encoding of Boolean expressions as mathematical objects in the PVS language. The `dnftarski` strategy parses Boolean expressions into an object having the type `PolyRelExpr`, which is a datatype with the following form:

```
PCONST(pb:bool) : PCONST?
PREL(pn:[nat->rat],d:nat,rel:TarskiRel,r:rat) : PREL?
PABS(pn:[nat->rat],d:nat,rel:TarskiRel,r:rat) : PABS?
PAND(pe1,pe2:PolyRelExpr) : PAND?
POR(pe1,pe2:PolyRelExpr) : POR?
PNOT(pe:PolyRelExpr) : PNOT?
PIMPLIES(pe1,pe2:PolyRelExpr) : PIMPLIES?
PIFF(pe1,pe2:PolyRelExpr) : PIFF?
PWHEN(pe1,pe2:PolyRelExpr) : PWHEN?
PITE(pe1,pe2,pe3:PolyRelExpr) : PITE?
```

Here, `PCONST` encodes `TRUE` or `FALSE`, `PREL` encodes a single polynomial relation, `PABS` encodes a polynomial relation with an absolute value, `PAND` encodes conjunction, `POR` encodes disjunction, `PNOT` encodes negation, `PIMPLIES` encodes implication, PIFF encodes equivalence, PWHEN encodes reverse implication, and PITE encodes conditional statements. In PREL and PABS, pn represents a polynomial $p(x) = a_n x^n + \cdots + a_0$ as a function from $\mathbb{N}$ to $\mathbb{R}$, so that $\texttt{pn}(i) = a_i$ if $i \leq n$ and $\texttt{pn}(i) = 0$ otherwise. The degree of this polynomial is represented by d, and rel represents the relation between p and 0, which is one of $>, \geq, <, \leq, =$, and $\neq$. Constants (including the zero polynomial) are represented with degree 0. As an example, the corresponding PolyRelExpr for $(x > 0 \vee 0 \geq 0) \wedge x^2 + 1 < 3$ is $\texttt{PAND}(\texttt{PREL}(f_1, 1, r_1), \texttt{POR}(\texttt{PREL}(f_2, 0, r_2), \texttt{PREL}(f_3, 2, r_3)))$ where r_1, r_2, and r_3 represent $>$, $\geq$, and $<$ respectively and f_1, f_2, and f_3 represent x, 0, and $x^2 - 2$, respectively.

The DNF transformation takes a PolyRelExpr as input and returns an object of type DNF. The DNF objects are defined as lists of lists of DNF_Atoms. Each DNF_Atom is a record with three fields that encode a polynomial (as a function from $\mathbb{N}$ to $\mathbb{R}$), the degree of the polynomial, and the relation between the polynomial and 0. A list of DNF_Atoms encodes a conjunction of polynomial inequalities. Therefore, it evaluates to TRUE if and only if every atom in the list evaluates to TRUE. An object of type DNF evaluates to TRUE if and only if at least one of its lists of DNF_Atoms evaluates to true.

The representation of polynomials as functions was chosen to maintain consistency between dnftarski and tarski, as tarski and its underlying theories represent polynomials as functions. Although representing polynomials as lists is computationally more efficient in general, maintaining consistency with legacy code is vital for efficiency—this will be discussed further in Section 4.

The theory dnf_polynomials contains the formalization of the DNF transformation. This theory proves, in particular, the lemma dnf_preserves_truth, which states that for each PolyRelExpr p, the evaluation of the DNF associated to p is logically equivalent to the evaluation of p. The theory dnf_strategy relates the evaluation of a DNF object to tarski, showing in particular that a DNF object evaluates to true if and only if tarski evaluates one of its lists of DNF_atoms to true. This result is verified in the lemma rel_to_tarski_sound, which is the key lemma in the dnftarski strategy.

As an important note, DNF transformations can greatly increase formula size. However, formulas that tarski is currently capable of handling are (almost) already in DNF format. Therefore, there is no formula size increase on these. Further, when formula size increase does occur, evaluating the lists of DNF_Atoms in parallel could help speed up the computation. As discussed in Section 4, most of the time in a dnftarski computation seems to be spent on the calls to tarski.

3.2 Preprocessing

All of the preprocessing methods are designed to target polynomial structure to quickly resolve QE queries in the existential conjunctive fragment. Preprocessing is introduced in an attempt to partially automate human intuition—the ultimate goal would be for PVS to be able to quickly resolve queries including those that humans can quickly resolve.

Towards this, the following properties are formally verified: Given the input query $F \equiv \exists x \in \mathbb{R} : f_1(x) \sim_1 0 \wedge \cdots \wedge f_n(x) \sim_n 0$, where each $\sim_i \in \{\geq, >, =, \leq, <, \neq\}$, then:

1. If for all i the constant term c_i of $f_i(x)$ satisfies $c_i \sim_i 0$, then F is `TRUE`

2. If for all i the leading coefficient k_i of f_i satisfies $k_i \sim_i 0$, then F is `TRUE`

3. F resolves to `TRUE` if for all i either: the degree of f_i is odd and the leading coefficient k_i of f_i satisfies $\neg(k_i \sim_i 0)$ or the degree of f_i is even and the leading coefficient k_i of f_i satisfies $k_i \sim_i 0$

These properties are equivalent to testing the sign of each polynomial at $x = 0$, $x = \infty$, and $x = -\infty$. The checks at $-\infty$ and ∞ were already occurring in `tarski`, but not in a preprocessing step. The preprocessing provides dramatic speedup on queries on which it succeeds (including resolving some queries on which `tarski` would otherwise hang), and minimal overhead when it fails. These properties are combined into `preprocessingStepConj` in `preprocessing_univariate`. In `preprocessingConjTheorem`, these properties are proven sound.

The main challenge in preprocessing is not proving the polynomial properties, but rather integrating them into `tarski` while maintaining soundness. The natural place to integrate preprocessing in the existing PVS development is in the `compute_solvable` function. However, the proof of this function is extremely complicated and does not easily lend itself to the addition of preprocessing. Instead, a function `compute_solvable_new` is defined with the preprocessing in place, and this is shown to be equivalent to the old `compute_solvable` function. The soundness proof of `preprocessingConjTheorem` is extremely modular, and thus it would be very easy to modify `preprocessingStepConj` to incorporate additional preprocessing to resolve input formulas F to `TRUE`.

4 Experimental Results

This section makes use of the benchmarks tested in [11]. Additionally, in order to more accurately pinpoint tradeoffs between `dnftarski` and `hutch`, and to more comprehensively test `dnftarski`, the following new sets of examples are used:

1. `adversarial_dnf_examples` — This theory contains a set of examples on which `hutch` runs very quickly but `dnftarski` runs quite slowly.

2. `adversarial_hutch_examples` — Conversely, this theory contains examples on which `hutch` runs more slowly than `dnftarski`.

3. `tarski_examples_preprocess` — This theory contains many examples on which the preprocessing simplifies the original expression.

4. `examples_for_parallelism` — This theory contains examples on which the strategy `dnftarski` is slow, but on which it would be much faster with parallelism.

All experiments are run on a 2018 Macbook Pro with 16 GB of memory and a 2.2 GHz Intel Core i7. The results are now discussed in more detail. All examples are listed in the appendices.

4.1 Performance on Benchmarks

The performance of the various strategies on the benchmarks from [11] is shown in Table 1. These examples are listed in Appendix A. The `hutch` strategy comes with an optional `sos`? flag that changes the underlying computations [11]. By default, this flag is set to true; this default behavior is referred to as `hutch`, and if instead the flag set to `nil`, the resulting strategy is referred to as `hutch` : `sos`? `nil`. An entry of "—" indicates that the strategy did not return an answer within 5 minutes. The number in parentheses is the time that it took to run the underlying decision procedure. The number outside the parentheses is the total time that it took to close the proof. The difference in the two numbers is largely due to syntactic manipulations, e.g., showing that different representations of polynomials are equivalent. Note in particular that both the DNF transformation and the preprocessing are taking place in the underlying decision procedure.

Problem	`hutch`	`hutch` : `sos`? `nil`	`tarski`(orig.)	`tarski`(prep.)	`dnftarski`
Ex1	3.01 (0.02)	3.04 (0.017)	2.15 (0.089)	2.74 (0.086)	3.10 (0.096)
Ex2	3.02 (0.06)	3.25 (0.3)	3.00 (1.52)	4.34 (1.52)	4.27 (1.58)
Ex3	27.92 (22.65)	6.61 (1.25)	2.03 (0.19)	5.48 (0.19)	8.06 (0.2)
Ex4	4.14 (0.0057)	4.35 (0.038)	7.82 (5.70)	10.51 (5.95)	10.61 (5.88)
Ex5	5.68 (0.0068)	5.88 (0.12)	166.85 (164.33)	169.48 (163.58)	173.83 (166.87)
Ex6	68.50 (2.40)	—	—	—	—
Ex7	69.75 (43.10)	—	—	—	—
quads_2	1.73 (0.0014)	1.71 (0.0015)	1.10 (0.005)	1.71 (0.0046)	1.37 (0.0052)
quads_3	2.09 (0.0021)	2.14 (0.0038)	1.34 (0.028)	2.19 (0.027)	1.80 (0.029)
quads_4	2.52 (0.0026)	2.59 (0.0097)	1.77 (0.19)	2.78 (0.18)	2.40 (0.19)
quads_5	3.10 (0.0034)	3.12 (0.028)	3.26 (1.41)	4.52 (1.38)	4.12 (1.48)
quads_6	3.71 (0.004)	3.71 (0.069)	13.03 (10.95)	14.58 (10.78)	14.70 (11.46)
quads_7	4.10 (0.0047)	4.47 (0.17)	90.29 (87.94)	89.18 (84.84)	94.71 (91.05)
quads_8	5.37 (0.0056)	5.69 (0.43)	—	—	—
quads_9	5.98 (0.0068)	6.86 (0.88)	—	—	—
quads_10	6.69 (0.0094)	7.97 (1.53)	—	—	—

Table 1. Strategies Performance in Seconds

The numbers overall reflect much faster runtimes than those in [11], likely due to the difference in machines. Most notably, `hutch` is able to close two problems on which it previously hung.

The similar run times of `tarski` (original) and `tarski` (with preprocessing) indicates that preprocessing adds negligible computational overhead. The time spent in the `dnftarski` and `tarski` decision procedures is almost identical in many cases (although this is not too surprising, given that these formulas are almost already in DNF format). In some examples, e.g., `Ex3`, `Ex5`, and `quads_7`, `dnftarski` is slightly slower than `tarski`.

As a remark, subtle choices in the strategy can greatly influence runtime—

especially in the final steps involving polynomial computations. For example, an earlier version of `dnf_tarski` represented polynomials as lists rather than as functions from $\mathbb{N}$ to $\mathbb{R}$. With this representation, there was considerable slowdown on certain examples—so that `quads_7` closed in 209.98(207.11) seconds and `Ex5` closed in 277.98(272.51) seconds. The reason for this slowdown appears to be that `dfn_tarski` depends on legacy developments where polynomials are still being represented as functions. Therefore, the list representation in `dnf_tarski` would add overhead as this representation has to be translated back and forth between the old legacy specifications and the new specifications. On other examples the slowdown was much more minimal.

4.2 New Examples

Here are some key observations from the experiments that were run on the new example sets.

4.2.1 Adversarial Examples for `dnftarski`

The examples in Appendix B suggest that the speed of `dnftarski` is largely predicated on the speed of its calls to `tarski`, i.e. the time difference between running `dnftarski` and summing the times it takes `tarski` to run on each of the conjunctive clauses in the DNF is often small.

However, when there are many clauses in the DNF, the `dnftarski` decision procedure is sometimes surprisingly slow. In `example_explode_5`, there are 144 clauses in the DNF and the `dnftarski` decision procedure takes 35.86 seconds. Although `tarski` has not been tested on each of the 144 clauses, none of them individually seems particularly complicated, so this runtime is surprisingly slow. It would be interesting to understand what is causing the slowdown, as running the DNF construction in isolation indicates that the overhead from transforming formulas into DNF is minuscule even in cases when the DNF contains many clauses. As discussed in Section 4.2.4, such slowdown could likely be elided by working on the clauses of the DNF in parallel.

4.2.2 Adversarial Examples for `hutch`

Appendix C lists a set of examples that are adversarial for `hutch`. Overall, it was more difficult to find examples that are adversarial for `hutch` than it was to find examples that are adversarial for `tarski` (and thus, by extension, `dnftarski`), which is consistent with the conclusions of [11]. Further, even for examples where `hutch` is extremely slow, `hutch : sos? nil` may be much faster—see `example_high_deg_1`, `example_high_deg_2`, and `example_high_deg_3`. However, as in `example_high_deg_4` and `example_with_equalities`, sometimes both `hutch` and `hutch : sos? nil` are very slow, whereas `dnftarski` is fast.

There are many factors which can change the performance of a given strategy. It seems that high-degree polynomials, polynomials with many roots, or polynomials with roots that are close together can slow `hutch` down. Further, `tarski` and dnftarski sometimes outperform hutch on queries that include an equality relation, such as example_high_deg_4 and example_with_equalities.

Moreover, it is sometimes the case that a single clause in the DNF of a complicated formula is easily resolved. If the first clause in the formula is easily resolved, dnftarski may be faster than hutchand hutch : sos? nil. This is the case in example_explode_formula. On this example, dnftarski takes about 10 seconds, and almost all of that is on polynomial computations. hutch is about 10 seconds slower than dnftarski, and in particular the decision procedure underlying hutch takes about 9 seconds. Interestingly, hutch : sos? nil is quite slow on this example.

4.2.3 Preprocessing

The examples in the theory tarski_examples_preprocess are listed in Appendix D. There are 15 examples total. The aggregate runtimes are given in Table 2. The time in parentheses is the aggregated time that it took to run the underlying decision procedures. The time outside the parentheses is the aggregated total time. While specific methods may be faster or slower on certain examples (for example, example_high_deg is particularly adversarial for hutch and example_conj_lc_4 is particularly adversarial for tarski) the preprocessing makes dnftarski extremely fast. Tarski hangs on two examples, i.e., it cannot return an answer within 5 minutes.

Method	Aggregated time (s)	Number Solved
Tarski (orig)	257.2 (227.46)	13
hutch	75.56 (38.37)	15
hutch : sos? nil	212.3 (175.71)	15
dnftarski	38.93 (0.073)	15

Table 2. Aggregated Times With and Without Preprocessing

4.2.4 Parallelism

As noted before, the clauses of a DNF formula could all be evaluated independently and in parallel. The examples in the theory examples_for_parallelism listed in Appendix E indicate that parallelism could be desirable not only when the DNF construction greatly increases the formula size (see, for example example_explode), but also on smaller DNFs when certain calls to tarski close very quickly and other calls take a long time (see, for example, example_many_roots_1 and example_slow). A strategy that allows parallel calls to tarski would help resolve the easier query that terminated the process without the burden of having to resolve the computationally expensive query.

On some of these examples, unless adding in parallelism were to incur significant overhead, it is likely to make dnftarski considerably faster than hutch. For instance, on example_many_roots_1, hutch takes 33.38 seconds of total time and hutch : sos? nil takes 42.64 seconds of total time. Here, dnftarski hangs because tarski is very slow on some of the initial conjunctive clauses. However, the conjunctive clause that resolves to TRUE (and thus decides the query) is very quickly resolved by tarski's preprocessing.

In another instance, on example_many_roots_2 hutch runs in 156.83 seconds of total time and hutch : sos? nil runs in 68.01 seconds of total time. Currently, dnftarski runs in 131.91 seconds of total time and requires four calls to tarski. Parallelizing could help reduce this considerably by allowing these four calls to tarski to happen simultaneously rather than sequentially.

4.3 Improvements to tarski and hutch

The performed testing uncovered some places in tarski and hutch where the strategy was unable to close some goals. First, in hutch, queries of the form $\neg\exists x \in \mathbb{R} : F(x)$ were not being discharged even when F was unsatisfiable. Similarly, queries of the form $\neg\forall x \in \mathbb{R} : F(x)$ were not being discharged even when $\neg F$ was satisfiable. This happened because hutch was handling these formulas by moving their negations to the antecedent—so that when given, for example, $\neg\exists x \in \mathbb{R} : F(x)$, it moved $\exists x \in \mathbb{R} : F(x)$ to the antecedent. Unfortunately, hutch was not storing any information to indicate that a formula had been moved to the antecedent, and so it treated the negations as if they were in the consequent. This has now been fixed.

Second, there was a subtle behavior where tarski would sometimes fail to discharge true queries, including "$\forall x \in \mathbb{R}, x > 0 \ \vee \ x+1 \leq 1$" and "$\forall x \in \mathbb{R}, x^9 + 12x^5 < 0 \ \vee \ x^2 \geq 49 \vee x^5 + 12x^2 + 32x = 0 \ \vee \ x > 0$". In these and other cases, the problem arose when PVS hid information regarding labels in the pre-assert function in pvs-strategies.[2]

In the first example, $x+1$ is labeled with a name, say $\texttt{name}_1$, so that $\texttt{name}_1 \leq 1$ is known. PVS then hides the meaning of $\texttt{name}_1$ and tries to prove $x \leq 0 \vee x > 0$, but it cannot do so from the information available. In the second example, the variable overlap occurs because when x^2 is labeled with a name, this name is substituted for the x^2 term in $x^5 + 12x^2 + 32$. When the meaning of the name for x^2 is hidden, PVS does not have enough information to close the proof.

The strategy has been edited so that the relevant information regarding labels is no longer hidden.

5 Conclusion

In this work, the PVS strategy tarski has been improved with a preprocessing step and extended in dnftarski, a new general-purpose strategy for univariate QE. Previously hutch was the only general-purpose strategy for univariate QE, and because quantifier elimination is such a computational bottleneck in proofs, it is desirable to have more than one strategy to perform quantifier elimination.

[2] Hiding *unnecessary* names in the strategies is highly desirable behavior, because the fewer formulas that PVS has to work with, the more efficient it will be. However, in the proofs of these examples, pre-assert was hiding necessary information.

Overall, `dnftarski` and `hutch` have different strengths and weaknesses. In particular, `dnftarski` performs poorly when the underlying calls to `tarski` perform poorly. This means that `hutch` and `hutch : sos? nil` often outperform `dnftarski`, as `tarski` is highly sensitive to the number of formulas in the query and somewhat sensitive to variable degree and polynomial complexity. However, various factors can hinder the performance of `hutch` and `hutch : sos? nil`, and in some cases `dnftarski` is superior. Generally, the speed of `dnftarski` seems to be predicated on the speed of its calls to `tarski`, although in cases with large DNFs, `dnftarski` can run more slowly. However, the DNF transformation itself seems to introduce minimal overhead, and the preprocessing increases the competitiveness of `dnftarski`.

6 Future Work

One could continue to extend `tarski` (and thus `dnftarski`) with additional preprocessing, as the existential conjunctive fragment lends itself very nicely to preprocessing. It would be easy to extend `preprocessingStepConj` to contain more preprocessing that can resolve formulas to "true". An ideal preprocessing routine would automate or supersede human intuition, and so a significant and challenging goal would be to implement reasoning to guess a rough range for values of x that would satisfy the formula. For example, a human can look at the formula $x^{350} - x^{90} + x^{80} - x^{60} + x^{50} - 10.5 < 9.5$ and quickly discern that the behavior is fundamentally different when $|x| < 1$ and when $|x| > 1$. In particular, as long as $|x| \leq 1$, it is easy to see that the formula is true. However once $|x| > 1$, the formula becomes unsatisfiable. So, one could approximate this formula with $-1 \leq x \wedge x \leq 1$.

The authors suggest adding preprocessing to automatically return "false" on systems that are evidently unsatisfiable. For example, experiments suggest that returning "false" on clauses that contain both some atom P and its negation $\neg P$ would be useful. Unfortunately, this preprocessing would not easily fit into the specification of `preprocessingStepConj`, and because the soundness proof of the `tarski` strategy is very complicated, the authors suggest implementing the transformation to "false" as an initial `transform_system` step, where an arbitrary query is transformed into "$\exists x : x^2 < 0$" in cases when the original system is clearly false. Further, this `transform_system` step could contain other preprocessing designed to reduce the number of polynomial relations in conjunctive clauses for which `tarski` must check satisfiability. For example, it could trim formulas by removing duplicate relations, and it could reduce linear systems with n clauses to systems with at most two clauses. Reducing linear systems would help improve `tarski`'s performance on interval computations. For example, currently `tarski` hangs on the computation "$\exists x : x^{25} - 10.28x^{39} + 6.0697x^3 + 96.6786x^2 - 125.32x - 6.50689 > 0 \wedge x \geq 8.4000001 \wedge x \leq -3.00001$", even though the two linear constraints are obviously inconsistent. (Surprisingly, `hutch` is also slow on this example.)

The authors also believe that it would be very worthwhile to change `dnftarski` to use parallelism. The DNF construction is inherently parallel, and currently the strategy is not taking advantage of this. Using parallelism would speed up the performance of the strategy on examples such as those in `examples_for_parallelism`.

It would also be possible to parallelize calls to `dnftarski` and both forms of `hutch`. This could be quite helpful—as the tradeoffs among the strategies are often very difficult to analyze a priori, it is often not clear which strategy will be fastest on a particular input problem.

References

1. C. W. Brown. Improved projection for cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 32(5):447–465, 2001.

2. C. Cohen and A. Mahboubi. Formal proofs in real algebraic geometry: From ordered fields to quantifier elimination. *Logical Methods in Computer Science*, 8(1:02):1–40, February 2012.

3. G. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Second GI Conference on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183, Kaiserslautern, 1975. Springer-Verlag.

4. G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12(3):299–328, 1991.

5. L. de Moura and G. Passmore. Computation in real closed infinitesimal and transcendental extensions of the rationals. In *Automated Deduction - CADE-24, 24th International Conference on Automated Deduction, Lake Placid, New York, June 9-14, 2013, Proceedings*, 2013.

6. A. Dolzmann and T. Sturm. REDLOG: Computer algebra meets computer logic. *Acm Sigsam Bulletin*, 31(2):2–9, 1997.

7. W. R. Inc. Mathematica, Version 12.0. Champaign, IL, 2019.

8. A. Mahboubi. Implementing the cylindrical algebraic decomposition within the coq system. *Mathematical Structures in Computer Science*, 17(1):99–127, 2007.

9. F. Maric. Formal verification of modern SAT solvers. *Archive of Formal Proofs*, July 2008.

10. A. Narkawicz, C. Muñoz, and A. Dutle. Formally-verified decision procedures for univariate polynomial computation based on Sturm's and Tarski's theorems. *Journal of Automated Reasoning*, 54(4):285–326, 2015.

11. A. Narkawicz, C. Muñoz, and A. Dutle. A decision procedure for univariate polynomial systems based on root counting and interval subdivision. *Journal of Formalized Reasoning*, 11(1):19–41, 2018.

12. A. J. Narkawicz and C. A. Muñoz. A formally-verified decision procedure for univariate polynomial computation based on Sturm's theorem. Technical Memorandum NASA/TM-2014-218548, NASA, Langley Research Center, Hampton VA 23681-2199, USA, November 2014.

13. S. Owre, J. Rushby, and N. Shankar. PVS: A prototype verification system. In D. Kapur, editor, *Proceeding of the 11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer, June 1992.
14. G. O. Passmore. *Combined Decision Procedures for Nonlinear Arithmetics, Real and Complex*. PhD thesis, The University of Edinburgh, 2011.
15. S. Sickert. Linear temporal logic. *Archive of Formal Proofs*, Mar. 2016.
16. A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, 1951.

Appendix A

Benchmarks

$$Ex1 : \forall\, x \in \mathbb{R} : x \geq -9 \,\wedge\, x < 10 \,\wedge\, x^4 > 0 \implies x^{12} > 0.$$

$$\begin{aligned}Ex2 : &\forall\, x \in \mathbb{R} : (x-2)^2 \cdot (-x+4) > 0 \,\wedge\, x^2 \cdot (x-3)^2 \geq 0 \,\wedge\, x - 1 \geq 0 \,\wedge \\ &-(x-3)^2 + 1 > 0 \implies (-(x - 11/12))^3 \cdot (x - 41/10)^3 \geq 0.\end{aligned}$$

$$\begin{aligned}Ex3 : &\exists\, x \in \mathbb{R} : x^5 - x - 1 = 0 \,\wedge\, x^{12} + 425/23 \cdot x^{11} - 228/23 \cdot x^{10} - 2 \cdot x^8 \\ &- 896/23 \cdot x^7 - 394/23 \cdot x^6 + 456/23 \cdot x^5 + x^4 + 471/23 \cdot x^3 \\ &+ 645/23 \cdot x^2 - 31/23 \cdot x - 228/23 = 0 \,\wedge\, x^3 + 22 \cdot x^2 - 31 \geq 0 \,\wedge \\ &x^{22} - 234/567 \cdot x^{20} - 419 \cdot x^{10} + 1948 > 0.\end{aligned}$$

$$\begin{aligned}Ex4 : &\forall\, x \in \mathbb{R} : x > 0 \,\vee\, -((61 \cdot x)/9) + (5 \cdot x^2)/9 + (20 \cdot x^3)/9 > -4 \,\vee \\ &1 \leq x \,\vee\, x \leq 0 \,\vee\, -((19 \cdot x)/9) + (10 \cdot x^2)/9 \leq -1 \,\vee\, -((13 \cdot x)/9) \\ &+ (31 \cdot x^2)/45 + x^3/18 \leq -(7/10) \,\vee\, -((61 \cdot x)/9) + (5 \cdot x^2)/9 \\ &+ (20 \cdot x^3)/9 \leq -4.\end{aligned}$$

$$\begin{aligned}Ex5 : &\forall\, x \in \mathbb{R} : -((5 \cdot x)/6) - (10 \cdot x^2)/3 - x^3/3 > 0 \,\vee\, (5 \cdot x)/6 \\ &+ (10 \cdot x^2)/3 + x^3/3 > 0 \,\vee\, 1 \leq x \,\vee\, x \leq 0 \,\vee\, -((19 \cdot x)/9) \\ &+ (10 \cdot x^2)/9 \leq -1 \,\vee\, -((13 \cdot x)/9) + (31 \cdot x^2)/45 + x^3/18 \leq -(7/10) \\ &\vee\, -((101 \cdot x)/30) - (64 \cdot x^2)/15 + (14 \cdot x^3)/15 \leq -(11/5) \,\vee \\ &-((61 \cdot x)/9) + (5 \cdot x^2)/9 + (20 \cdot x^3)/9 \leq -4.\end{aligned}$$

$$
\begin{aligned}
Ex6 : \exists\, x \in \mathbb{R} : &-((51\cdot x)/10) - (267\cdot x^2)/2 - (5409\cdot x^3)/10 - (4329\cdot x^4)/5 \\
&- (2052\cdot x^5)/5 - 70\cdot x^6 > -(7/10) \ \wedge\ -((10327\cdot x)/270) \\
&- (71681\cdot x^2)/270 - (135853\cdot x^3)/810 - (57328\cdot x^4)/135 \\
&+ (77743\cdot x^5)/135 + (115774\cdot x^6)/405 + (175\cdot x^7)/18 + (49\cdot x^8)/3 \\
&+ (49\cdot x^9)/162 > -(721/90) \ \wedge\ -((2981\cdot x)/90) - (251\cdot x^2)/6 \\
&- (24217\cdot x^3)/270 + (2698\cdot x^4)/135 + (18964\cdot x^5)/135 \\
&- (595\cdot x^6)/54 + (280\cdot x^7)/27 + (7\cdot x^8)/27 > -(206/45) \ \wedge \\
&- ((799\cdot x)/90) + (169\cdot x^2)/18 - (7933\cdot x^3)/270 + (2672\cdot x^4)/135 \\
&+ (329\cdot x^5)/90 + (112\cdot x^6)/27 + (7\cdot x^7)/54 > -(103/90) \ \wedge \\
&- ((781\cdot x)/90) - (701\cdot x^2)/6 - (12217\cdot x^3)/270 + (11323\cdot x^4)/135 \\
&+ (7264\cdot x^5)/135 + (935\cdot x^6)/54 + (280\cdot x^7)/27 \\
&+ (7\cdot x^8)/27 > -(77/15) \ \wedge\ -((361\cdot x)/30) \\
&- (811\cdot x^2)/30 + (307\cdot x^3)/45 + (2353\cdot x^4)/90 - (17\cdot x^5)/6 \\
&+ (52\cdot x^6)/9 + (2\cdot x^7)/9 > -(44/15) \ \wedge\ -((33\cdot x)/10) - (2\cdot x^2)/15 \\
&+ (41\cdot x^3)/90 + (2\cdot x^4)/15 + 2\cdot x^5 + x^6/9 > -(11/15) \ \wedge \\
&- ((1339\cdot x)/405) - (70225\cdot x^2)/324 - (11549\cdot x^3)/270 \\
&+ (65378\cdot x^4)/405 + (23483\cdot x^5)/810 + (1109\cdot x^6)/27 \\
&+ (1540\cdot x^7)/81 + (49\cdot x^8)/162 > -(721/60) \ \wedge\ -((10741\cdot x)/540) \\
&- (2263\cdot x^2)/45 + (5191\cdot x^3)/180 + (7753\cdot x^4)/270 - (52\cdot x^5)/9 \\
&+ (203\cdot x^6)/18 + (7\cdot x^7)/27 > -(103/15) \ \wedge\ -((1481\cdot x)/90) \\
&- (811\cdot x^2)/180 + (2113\cdot x^3)/90 - (493\cdot x^4)/36 + (59\cdot x^5)/9 \\
&+ (2\cdot x^6)/9 > -(22/5) \ \wedge\ -((913\cdot x)/180) + (563\cdot x^2)/90 \\
&- (257\cdot x^3)/60 + (17\cdot x^4)/9 + x^5/9 > -(11/10) \ \wedge \\
&- ((91\cdot x)/18) + (10\cdot x^2)/3 - (5\cdot x^3)/2 + (20\cdot x^4)/9 > -2 \ \wedge \\
&- ((2\cdot x)/9) - (25\cdot x^2)/18 + (10\cdot x^3)/9 > -(1/2) \ \wedge \\
&- ((61\cdot x)/9) + (5\cdot x^2)/9 + (20\cdot x^3)/9 > -4 \ \wedge\ 1 > x \ \wedge\ x > 0 \ \wedge \\
&- ((19\cdot x)/9) + (10\cdot x^2)/9 > -1 \ \wedge\ -((13\cdot x)/9) + (31\cdot x^2)/45 \\
&+ x^3/18 > -(7/10) \ \wedge\ -((253\cdot x)/90) - (53\cdot x^2)/30 + (34\cdot x^3)/15 \\
&+ x^4/9 > -(11/5) \ \wedge\ -((97\cdot x)/90) - (2051\cdot x^2)/90 + (86\cdot x^3)/15 \\
&+ (82\cdot x^4)/9 + (2\cdot x^5)/9 > -(44/5) \ \wedge\ -((93307\cdot x)/1620) \\
&- (298609\cdot x^2)/810 + (30583\cdot x^3)/270 + (264373\cdot x^4)/810 \\
&- (289811\cdot x^5)/1620 + (3113\cdot x^6)/27 + (931\cdot x^7)/81 + (8\cdot x^8)/81 > \\
&- (193/5) \wedge -((4741\cdot x)/540) - (9151\cdot x^2)/90 + (6397\cdot x^3)/60 \\
&- (2686\cdot x^4)/135 + (28\cdot x^5)/9 + (38\cdot x^6)/3 + (7\cdot x^7)/27 > -(77/10).
\end{aligned}
$$

$$\begin{aligned}
Ex7: &\forall x \in \mathbb{R} : x < -1 \lor 0 > x \lor (41613 \cdot x)/2 + 26169 \cdot x^2 \\
&+ (64405 \cdot x^3)/4 + 4983 \cdot x^4 + (7083 \cdot x^5)/10 + (1207 \cdot x^6)/35 \\
&+ x^7/8 > -6435 \lor 11821609800 \cdot x + 22461058620 \cdot x^2 + 35 \cdot x^{12} \leq \\
&4171407240 \cdot x^3 + 45938678170 \cdot x^4 + 54212099480 \cdot x^5 \\
&+ 31842714428 \cdot x^6 + 10317027768 \cdot x^7 + 1758662439 \cdot x^8 \\
&+ 144537452 \cdot x^9 + 5263834 \cdot x^{10} + 46204 \cdot x^{11} \lor x \leq 0 \lor \\
&9609600 \cdot x + 45805760 \cdot x^2 + 92372280 \cdot x^3 + 102560612 \cdot x^4 \\
&+ 68338600 \cdot x^5 + 27930066 \cdot x^6 + 6857016 \cdot x^7 + 938908 \cdot x^8 \\
&+ 58568 \cdot x^9 + 753 \cdot x^{10} \leq 0 \lor 788107320 \cdot x + 1101329460 \cdot x^2 \\
&+ 10 \cdot x^{11} \leq 782617220 \cdot x^3 + 2625491260 \cdot x^4 + 2362290448 \cdot x^5 \\
&+ 1063536663 \cdot x^6 + 240283734 \cdot x^7 + 24397102 \cdot x^8 + 1061504 \cdot x^9 \\
&+ 9179 \cdot x^{10} \lor 90935460 \cdot x + 81290790 \cdot x^2 + 5 \cdot x^{10} \leq 125595120 \cdot x^3 \\
&+ 237512625 \cdot x^4 + 161529144 \cdot x^5 + 51834563 \cdot x^6 + 6846880 \cdot x^7 \\
&+ 356071 \cdot x^8 + 2828 \cdot x^9 \lor 640640 \cdot x + 2735040 \cdot x^2 + 4837448 \cdot x^3 \\
&+ 4581220 \cdot x^4 + 2505504 \cdot x^5 + 794964 \cdot x^6 + 138652 \cdot x^7 + 11237 \cdot x^8 \\
&+ 207 \cdot x^9 \leq 0 \lor 5 \cdot x^8 \leq 73920 \cdot x + 238560 \cdot x^2 + 303324 \cdot x^3 \\
&+ 192458 \cdot x^4 + 63520 \cdot x^5 + 10261 \cdot x^6 + 608 \cdot x^7 \lor 73920 \cdot x \\
&+ 278880 \cdot x^2 + 424284 \cdot x^3 + 332962 \cdot x^4 + 142928 \cdot x^5 + 32711 \cdot x^6 \\
&+ 3514 \cdot x^7 + 98 \cdot x^8 \leq 0 \lor x \leq -1.
\end{aligned}$$

$$\begin{aligned}
quads_2: &\forall x \in \mathbb{R} : x > 0 \land x < 2 \implies ((x-0)\cdot(x-1) \leq 0 \lor \\
&(x-1)\cdot(x-2) \leq 0).
\end{aligned}$$

$$\begin{aligned}
quads_3: &\forall x \in \mathbb{R} : x > 0 \land x < 3 \implies ((x-0)\cdot(x-1) \leq 0 \lor \\
&(x-1)\cdot(x-2) \leq 0 \lor (x-2)\cdot(x-3) \leq 0).
\end{aligned}$$

$$\begin{aligned}
quads_4: &\forall x \in \mathbb{R} : x > 0 \land x < 4 \implies ((x-0)\cdot(x-1) \leq 0 \lor \\
&(x-1)\cdot(x-2) \leq 0 \lor (x-2)\cdot(x-3) \leq 0 \lor (x-3)\cdot(x-4) \leq 0).
\end{aligned}$$

$$\begin{aligned}
quads_5: &\forall x \in \mathbb{R} : x > 0 \land x < 5 \implies ((x-0)\cdot(x-1) \leq 0 \lor \\
&(x-1)\cdot(x-2) \leq 0 \lor (x-2)\cdot(x-3) \leq 0 \lor (x-3)\cdot(x-4) \leq 0 \lor \\
&(x-4)\cdot(x-5) \leq 0).
\end{aligned}$$

$$\begin{aligned}
quads_6: &\forall x \in \mathbb{R} : x > 0 \land x < 6 \implies ((x-0)\cdot(x-1) \leq 0 \lor \\
&(x-1)\cdot(x-2) \leq 0 \lor (x-2)\cdot(x-3) \leq 0 \lor (x-3)\cdot(x-4) \leq 0 \lor \\
&(x-4)\cdot(x-5) \leq 0 \lor (x-5)\cdot(x-6) \leq 0).
\end{aligned}$$

$$
\begin{aligned}
\mathit{quads\_7}: \forall x \in \mathbb{R} : x > 0 \wedge x < 7 \implies ((x-0)\cdot(x-1) \le 0 \vee \\
(x-1)\cdot(x-2) \le 0 \vee (x-2)\cdot(x-3) \le 0 \vee (x-3)\cdot(x-4) \le 0 \vee \\
(x-4)\cdot(x-5) \le 0 \vee (x-5)\cdot(x-6) \le 0 \vee (x-6)\cdot(x-7) \le 0).
\end{aligned}
$$

$$
\begin{aligned}
\mathit{quads\_8}: \forall x \in \mathbb{R} : x > 0 \wedge x < 8 \implies ((x-0)\cdot(x-1) \le 0 \vee \\
(x-1)\cdot(x-2) \le 0 \vee (x-2)\cdot(x-3) \le 0 \vee (x-3)\cdot(x-4) \le 0 \vee \\
(x-4)\cdot(x-5) \le 0 \vee (x-5)\cdot(x-6) \le 0 \vee (x-6)\cdot(x-7) \le 0 \vee \\
(x-7)\cdot(x-8) \le 0).
\end{aligned}
$$

$$
\begin{aligned}
\mathit{quads\_9}: \forall x \in \mathbb{R} : x > 0 \wedge x < 9 \implies ((x-0)\cdot(x-1) \le 0 \vee \\
(x-1)\cdot(x-2) \le 0 \vee (x-2)\cdot(x-3) \le 0 \vee (x-3)\cdot(x-4) \le 0 \vee \\
(x-4)\cdot(x-5) \le 0 \vee (x-5)\cdot(x-6) \le 0 \vee (x-6)\cdot(x-7) \le 0 \vee \\
(x-7)\cdot(x-8) \le 0 \vee (x-8)\cdot(x-9) \le 0).
\end{aligned}
$$

$$
\begin{aligned}
\mathit{quads\_10}: \forall x \in \mathbb{R} : x > 0 \wedge x < 10 \implies ((x-0)\cdot(x-1) \le 0 \vee \\
(x-1)\cdot(x-2) \le 0 \vee (x-2)\cdot(x-3) \le 0 \vee (x-3)\cdot(x-4) \le 0 \vee \\
(x-4)\cdot(x-5) \le 0 \vee (x-5)\cdot(x-6) \le 0 \vee (x-6)\cdot(x-7) \le 0 \vee \\
(x-7)\cdot(x-8) \le 0 \vee (x-8)\cdot(x-9) \le 0 \vee (x-9)\cdot(x-10) \le 0).
\end{aligned}
$$

Appendix B

Adversarial DNF Examples

$$\begin{aligned}\mathit{example\_ta\_ors\_1} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor \\ & x^3 \geq 8 \lor x^5 + 12{\cdot}x^2 + 32{\cdot}x = 0 \lor x > 0 \lor x < 2.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ta\_ors\_2} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor \\ & x^3 \geq 8 \lor x^5 + 12{\cdot}x^2 + 32{\cdot}x = 0 \lor x > 0.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ta\_ors\_3} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor \\ & x^3 \geq 8 \lor x = 0 \lor x > 0 \lor x < 2.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ta\_ors\_4} :\forall x \in \mathbb{R} : {}& x < 2 \lor x > 0 \lor x = 0 \lor \\ & x^3 \geq 8 \lor x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor \\ & x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ta\_ors\_5} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor \\ & x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor \\ & x^3 \geq 8 \lor x^5 + 12{\cdot}x^2 + 32{\cdot}x = 0 \lor x > 0.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ta\_ors\_6} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor \\ & x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor x^3 \geq 8 \lor x = 0 \lor x < 2.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ta\_ors\_7} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor \\ & x^3 \geq 8 \lor x = 0 \lor x > 0.\end{aligned}$$

$$\begin{aligned}\mathit{example\_ors\_8} :\forall x \in \mathbb{R} : {}& x^9 + 12{\cdot}x^5 < 0 \lor x^2 \geq 49 \lor \\ & x^3 + 5{\cdot}x^8 + 32{\cdot}x^6 + x > 1/2 \lor x^3 \geq 8 \lor x = 0 \lor (x > 0 \land x < 2).\end{aligned}$$

$$\begin{aligned}\mathit{example\_explode\_1} :\forall x \in \mathbb{R} : {}& (x < 0 \land x^2 > 0) \lor (x^2 \geq 49 \land x \geq 7) \lor \\ & (x > 0 \land x + 2 > 2 \land x + 5 > 3) \lor (x = 0 \land x^2 = 0 \land x^3 = 0).\end{aligned}$$

$$\begin{aligned}\mathit{example\_explode\_2} :\forall x \in \mathbb{R} : {}& (x^2 \neq 2 \land x^3 \neq 3 \land x = 0) \lor (x < 0 \land x^2 > 0) \\ & \lor (x^2 \geq 49 \land x \geq 7) \lor (x > 0 \land x + 2 > 2 \land x + 5 > 3).\end{aligned}$$

$$\begin{aligned} \mathit{example\_explode\_4} : \exists\, x \in \mathbb{R} : &(x < 0 \ \vee\ x^2 > 0) \ \wedge\ (x^2 \geq 49 \ \vee\ x \geq 10) \wedge \\ &(x > 1/2 \ \vee\ x + 2 > 300 \ \vee\ x + 5 > 20) \wedge \\ &(x^3 \geq 8 \ \vee\ x > 1) \ \wedge\ (x < -20 \ \vee\ x < -12 \ \vee\ x^3 \neq 35) \ \wedge\ (x < 2 \ \vee\ x > 0). \end{aligned}$$

$$\begin{aligned} \mathit{example\_explode\_5} : \exists\, x \in \mathbb{R} : &(x < 0 \ \vee\ x^2 > 0) \ \wedge\ (x^2 \geq 49 \ \vee\ x \geq 10) \wedge \\ &(x > 1/2 \ \vee\ x + 2 > 300 \ \vee\ x + 5 > 20) \ \wedge\ (x^3 \geq 8 \ \vee\ x > 1) \wedge \\ &(x^3 \neq 35 \ \vee\ x < -20 \ \vee\ x < -12) \ \wedge\ (x > 0 \ \vee\ x < 2). \end{aligned}$$

$$\mathit{example\_explode\_6} : \exists\, x \in \mathbb{R} : x^2 > 0 \ \wedge\ x^2 \geq 49 \ \wedge\ x + 2 > 300 \ \wedge\ x^3 \geq 8 \ \wedge\ x^3 \neq 35 \ \wedge\ x > 0.$$

$$\begin{aligned} \mathit{example\_explode\_7} : \forall\, x \in \mathbb{R} : &(x < 0 \ \wedge\ x^2 > 0) \ \vee\ (x^2 \geq 49 \ \wedge\ x \geq 7) \vee \\ &(x > 1/2 \ \wedge\ x + 2 > 5/2 \ \wedge\ x + 5 > 3) \vee \\ &(x^3 \geq 8 \ \wedge\ x > 1) \ \vee\ (x \neq 5 \ \wedge\ x^2 \neq 25 \ \wedge\ x^3 \neq 125). \end{aligned}$$

$$\begin{aligned} \mathit{example\_slow} : \exists\, x \in \mathbb{R} : &(x^5 - 4{\cdot}x^4 + 16{\cdot}x^3 - 2348{\cdot}x^2 + 10{\cdot}x - 14 > 0 \wedge \\ &x^{12} - 4{\cdot}x^4 + 16{\cdot}x^3 - 2348{\cdot}x^2 + 10{\cdot}x - 14 < 0) \vee \\ &(589{\cdot}x^7 - 25{\cdot}x^6 - 4{\cdot}x^4 + 16{\cdot}x^3 - 2348{\cdot}x^2 + 10{\cdot}x + 14 < 0 \\ &\wedge\ x^{12} - 4{\cdot}x^4 + 16{\cdot}x^3 - 2348{\cdot}x^2 + 10{\cdot}x - 14 < 0 \ \wedge\ -35{\cdot}x^{20} < -20). \end{aligned}$$

$$\begin{aligned} \mathit{example\_slow\_tarski} : \exists\, x \in \mathbb{R} : &589{\cdot}x^7 - 25{\cdot}x^6 - 4{\cdot}x^4 + 16{\cdot}x^3 - 2348{\cdot}x^2 + 10{\cdot}x + 14 < 0 \wedge \\ &x^{12} - 4{\cdot}x^4 + 16{\cdot}x^3 - 2348{\cdot}x^2 + 10{\cdot}x - 14 < 0 \ \wedge\ -35{\cdot}x^{20} < -20. \end{aligned}$$

$$\mathit{example\_high\_degree} : \neg\, \exists\, x \in \mathbb{R} : (x^{101} - 5{\cdot}x^{100} + 10{\cdot}x - 510)^2 + (x^{11} - 11{\cdot}x^{10} + 2{\cdot}x^3 + x)^2 = 0.$$

Appendix C

Adversarial Hutch Examples

$$
\begin{aligned}
\mathit{example\_hutch\_slow\_1} :\neg \exists x \in \mathbb{R} : &\; x^5 - 11.5{\cdot}x^4 - 27.5{\cdot}x^3 + 223.5{\cdot}x^2 + 436.5{\cdot}x - 270 = 0 \wedge \\
&\; x^7 + 31.5{\cdot}x^6 - 258{\cdot}x^5 - 10007{\cdot}x^4 + 25881{\cdot}x^3 + 350312{\cdot}x^2 + 467640{\cdot}x - \\
&\; 324000.125000001 = 0 \wedge \\
&\; x^7 - 5{\cdot}x^6 - 8.75{\cdot}x^5 + 16{\cdot}x^4 + 4.75{\cdot}x^3 - 11{\cdot}x^2 + 3{\cdot}x = 0 \ \wedge\ x \neq -3.
\end{aligned}
$$

$$
\begin{aligned}
\mathit{example\_hutch\_slightly\_slow\_1} :\neg \exists x \in \mathbb{R} : &\; x^4 + 10{\cdot}x^3 + 35{\cdot}x^2 + 50{\cdot}x + 24 = 0 \wedge \\
&\; (x^5 - 35{\cdot}x^4 + 485{\cdot}x^3 - 3325{\cdot}x^2 + 11274{\cdot}x - 0.5 > 0 \vee \\
&\; - 12{\cdot}x^3 - x^5 < 0).
\end{aligned}
$$

$$\mathit{example\_high\_deg\_1} : \exists x \in \mathbb{R} : x^{240} - 5{\cdot}x^8 + 32{\cdot}x^6 + x^2 > 1/3 \ \wedge\ x < 1/2.$$

$$\mathit{example\_high\_deg\_2} :\neg \exists x \in \mathbb{R} : x^{240} - 5{\cdot}x^8 + 32{\cdot}x^6 + x^2 > 1/3 \ \wedge\ x^2 = 0 \ \wedge\ x < -1/2.$$

$$\mathit{example\_high\_deg\_3} : \exists x \in \mathbb{R} : x^{350} - x^{90} + x^{80} - x^{60} + x^{50} - 10.5 < 9.5 \ \wedge\ x^3 = 0.5.$$

$$\mathit{example\_high\_deg\_4} : \exists x \in \mathbb{R} : x^{350} - x^{90} + x^{80} - x^{60} + x^{50} - 10.5 < 9.5 \ \wedge\ x^3 - x^2 = -0.01.$$

$$
\begin{aligned}
\mathit{example\_check} : \exists x \in \mathbb{R} : &\; x^{350} - x^{90} + x^{80} - x^{60} + x^{50} - 10.5 < 9.5 \wedge \\
&\; x^2 - 0.010207{\cdot}x - 0.0101031 = 0.
\end{aligned}
$$

$$
\begin{aligned}
\mathit{example\_with\_equality} : \forall x \in \mathbb{R} : &\; x^2 \neq 0 \ \vee\ x^8 - 12{\cdot}x - 0.001 + x^{25} - 20{\cdot}x^{12} = 0 \vee \\
&\; (x^2 < 5 \ \wedge\ x \neq 1.2617199999) \ \vee\ -x^2 + x^4 + x^6 - x > 0 \vee \\
&\; x^2 + x^4 - x^6 - x - 0.0001 < 0.
\end{aligned}
$$

$$
\begin{aligned}
\mathit{example\_with\_equalities} : \forall x \in \mathbb{R} : &\; x^2 \neq 0 \ \vee\ ((x^8 - 12{\cdot}x - 0.001 + x^{25} - 20{\cdot}x^{12} = 0 \vee \\
&\; (x^2 < 5 \ \wedge\ x \neq 1.2617199999) \ \vee\ -x^2 + x^4 + x^6 - x > 0 \vee \\
&\; x^2 + x^4 - x^6 - x - 0.0001 < 0) \ \wedge\ x^{90} - x^{80} + 0.0001 < 0.002).
\end{aligned}
$$

$$
\begin{aligned}
\mathit{example\_explode\_formula} : \exists x \in \mathbb{R} : &\; (x < 0 \ \vee\ x^{100} - x^{90} < 0) \wedge \\
&\; (x^2 \geq 49 \ \vee\ x \geq 10 \ \vee\ x^3 - 9{\cdot}x^2 \geq 0) \wedge \\
&\; (x < 1/2 \ \vee\ x^{102} + 5 > 20) \wedge \\
&\; (x^3 < 8 \ \vee\ x^4 + 1.8{\cdot}x^3 - 3.59{\cdot}x^2 - 3.96{\cdot}x + 4.84 > 0) \wedge \\
&\; (x^3 \neq 35 \ \vee\ x < -20) \ \wedge\ (x < 2 \ \vee\ -0.0001{\cdot}x^3 \leq -0.0008).
\end{aligned}
$$

Appendix D

Tarski Examples with Preprocessing

$$example\_odd\_1 : \exists\, x \in \mathbb{R} : x^{27} + 312{\cdot}x^2 + 513{\cdot}x^{22} + 1200000 < 0.$$

$$example\_odd\_2 : \exists\, x \in \mathbb{R} : x^{27} + 312{\cdot}x^2 + 513{\cdot}x^{22} + 1200000 = 0.$$

$$example\_odd\_3 : \exists\, x \in \mathbb{R} : x^{27} + 312{\cdot}x^{26} - x^{25} + x^{24} - 30{\cdot}x^{23} + 153{\cdot}x^{22} + 513{\cdot}x + 12 > 0.$$

$$\begin{aligned} example\_conj\_odd\_1 : \exists\, x \in \mathbb{R} : {} & x^{27} + 312{\cdot}x^2 + 513{\cdot}x^{22} + 10 < 0 \,\wedge \\ & 2{\cdot}x^{27} - 312{\cdot}x^2 - 3000{\cdot}x^{22} - 20 < 0 \,\wedge\, 12{\cdot}x^{85} + 1250{\cdot}x^{84} < 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_odd\_2 : \exists\, x \in \mathbb{R} : {} & -x^{27} + 312{\cdot}x^2 + 513{\cdot}x^{22} + 10 > 0 \,\wedge \\ & -12{\cdot}x^{25} + 25{\cdot}x^2 \geq 0 \,\wedge\, -x^3 + 248325{\cdot}x - 35 \geq 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_odd\_3 : \exists\, x \in \mathbb{R} : {} & -x^{27} + 312{\cdot}x^2 + 513{\cdot}x^{22} + 10 > 0 \,\wedge \\ & -12{\cdot}x^{25} + 25{\cdot}x^2 \geq 0 \,\wedge\, x^3 + 248325{\cdot}x - 35 \leq 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_odd\_4 : \exists\, x \in \mathbb{R} : {} & -x^{27} + 312{\cdot}x^2 + 513{\cdot}x^{22} + 10 > 0 \,\wedge \\ & -12{\cdot}x^{25} + 25{\cdot}x^2 \geq 0 \,\wedge\, x^3 + 248325{\cdot}x - 35 \leq 0 \,\wedge\, 30{\cdot}x^{25} - 40{\cdot}x - 350 < 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_coeff\_1 : \exists\, x \in \mathbb{R} : {} & -x^{27} + 120 > 0 \,\wedge\, -x^{27} - x^{26} - x^{25} + 1 > 0 \,\wedge \\ & -67{\cdot}x^{67} - 100{\cdot}x^{66} - 30{\cdot}x^{65} + 30 > 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_coeff\_2 : \exists\, x \in \mathbb{R} : {} & -x^{27} + 120 > 0 \,\wedge\, -x^{27} - x^{26} - x^{25} + 1 > 0 \,\wedge \\ & -67{\cdot}x^{67} - 100{\cdot}x^{66} - 30{\cdot}x^{65} + 30 > 0 \,\wedge\, x + 12 > 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_lc\_1 : \exists\, x \in \mathbb{R} : {} & -x^{26} + 12{\cdot}x^5 \leq 0 \,\wedge\, -50{\cdot}x^{27} - 10{\cdot}x^{26} + 400 < 0 \,\wedge \\ & -2{\cdot}x^2 + 100{\cdot}x + 50 \leq 0 \,\wedge\, -x < 0. \end{aligned}$$

$$\begin{aligned} example\_conj\_lc\_2 : \exists\, x \in \mathbb{R} : {} & x^{26} + 12{\cdot}x^5 \geq 0 \,\wedge\, x^{27} - 10{\cdot}x^{26} + 400 > 0 \,\wedge \\ & x^5 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 \geq 0. \end{aligned}$$

$$
\begin{aligned}
example\_conj\_lc\_3 : \exists\, x \in \mathbb{R} : {} & x^{26} + 12{\cdot}x^5 \geq 0 \;\wedge\; x^{27} - 10{\cdot}x^{26} + 400 > 0 \,\wedge \\
& x^5 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 \geq 0 \,\wedge \\
& -213{\cdot}x^6 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 < 0.
\end{aligned}
$$

$$
\begin{aligned}
example\_conj\_lc\_4 : \exists\, x \in \mathbb{R} : {} & x^{26} + 12{\cdot}x^5 \geq 0 \;\wedge\; x^{27} - 10{\cdot}x^{26} + 400 > 0 \,\wedge \\
& x^5 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 \geq 0 \,\wedge \\
& -213{\cdot}x^6 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 < 0 \;\wedge\; -2{\cdot}x^{11} + 23{\cdot}x^2 \leq 0.
\end{aligned}
$$

$$
\begin{aligned}
example\_conj\_lc\_5 : \exists\, x \in \mathbb{R} : {} & x^{26} + 12{\cdot}x^5 \geq 0 \;\wedge\; x^{27} - 10{\cdot}x^{26} + 400 > 0 \,\wedge \\
& x^5 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 \geq 0 \,\wedge \\
& -213{\cdot}x^6 - 100{\cdot}x^4 - 200{\cdot}x^3 - 100{\cdot}x - 50 < 0 \,\wedge \\
& -2{\cdot}x^{11} + 23{\cdot}x^2 \leq 0 \;\wedge\; -2{\cdot}x^{13} \leq 0.
\end{aligned}
$$

$$
example\_high\_deg : \exists\, x \in \mathbb{R} : x^{120} - 5{\cdot}x^8 + 32{\cdot}x^6 + x^2 > 1/3 \;\wedge\; x > 2.
$$

Appendix E

Examples for Parallelism

$$\begin{aligned}\mathit{example\_slow} :& \forall\, x \in \mathbb{R} : (x < 0 \iff x^{37} + 12{\cdot}x^3 - 57 < 0) \vee \\ & (x < 0 \iff x^9 + 12{\cdot}x^3 < 0 \wedge x^2 > 0).\end{aligned}$$

$$\begin{aligned}\mathit{example\_slow\_tarski} :& \exists\, x \in \mathbb{R} : x < 0 \wedge x^{37} + 12{\cdot}x^3 - 57 \geq 0 \wedge x \geq 0 \wedge \\ & x^9 + 12{\cdot}x^3 \geq 0.\end{aligned}$$

$$\begin{aligned}\mathit{example\_many\_roots\_1} :& \neg \forall\, x \in \mathbb{R} : x^{25} - 10.28{\cdot}x^{39} + 6.0697{\cdot}x^3 + 96.6786{\cdot}x^2 - \\ & 125.32{\cdot}x - 6.50689 > 0 \implies \\ & ((x < 8.4000001 \wedge x > -1.510002) \vee (x > -3.00001 \wedge x < 9)).\end{aligned}$$

$$\begin{aligned}\mathit{example\_many\_roots\_1\_tarski} :& \exists\, x \in \mathbb{R} : x^{25} - 10.28{\cdot}x^{39} + 6.0697{\cdot}x^3 + 96.6786{\cdot}x^2 - \\ & 125.32{\cdot}x - 6.50689 > 0 \wedge x \leq -1.510002 \wedge x \leq -3.00001.\end{aligned}$$

$$\begin{aligned}\mathit{example\_many\_roots\_2} :& \forall\, x \in \mathbb{R} : x^{25} - 10.28{\cdot}x^{49} + 6.0697{\cdot}x^3 + 96.6786{\cdot}x^2 - \\ & 125.32{\cdot}x - 6.50689 = 0 \implies \\ & ((x < 8.4000001 \wedge x > -1.510002) \vee (x > -3.00001 \wedge x < 9)).\end{aligned}$$

$$\begin{aligned}\mathit{example\_explode} :& \forall\, x \in \mathbb{R} : (x < 0 \wedge x^2 > 0) \vee (x^2 \geq 49 \wedge x \geq 7) \vee \\ & (x > 1/2 \wedge x + 2 > 5/2 \wedge x + 5 > 3) \vee \\ & (x^3 \geq 8 \wedge x > 1) \vee (x = 0 \wedge x^2 = 0 \wedge x^3 = 0) \vee (x > 0 \wedge x < 2)\end{aligned}$$

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD-MM-YYYY)* 01-01-2021	**2. REPORT TYPE** Technical Memorandum	**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE Improving Automated Strategies for Univariate Quantifier Elimination	**5a. CONTRACT NUMBER**	
	5b. GRANT NUMBER	
	5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Cordwell, Katherine; Munoz , Cesar A.; Dutle, Aaron M.	**5d. PROJECT NUMBER**	
	5e. TASK NUMBER	
	5f. WORK UNIT NUMBER 340428.02.20.07.01	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, Virginia 23681-2199	**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	**10. SPONSOR/MONITOR'S ACRONYM(S)** NASA	
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/TM–20205010644	

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category: Computer Programming and Software
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report discusses improved support for univariate quantifier elimination in the Prototype Verification System (PVS). Previously, PVS had three strategies for quantifier elimination—`hutch`, `tarski`, and `sturm`. Of these, only `hutch` is able to decide queries in any input format—`sturm` only works on queries regarding a single polynomial on an interval and `tarski` resolves queries in the universal existential fragment. This paper describes an extended version of `tarski`. The extension is accomplished by formally verifying a disjunctive normal form transformation in PVS and using `tarski` on each conjunctive clause. Additionally, a preprocessing step is added to the decision procedure underlying `tarski`. This preprocessing is designed to exploit properties of polynomial structure to quickly resolve queries that have certain formats. The preprocessing produces dramatic speedup when it succeeds in resolving a query, and seems to introduce negligible overhead when it does not resolve a query. Finally, testing reveals some ways to improve the `hutch` and `tarski` strategies.

15. SUBJECT TERMS
Polynomial Constraints, Quantifier Elimination, Prototype Verification System (PVS), Automated Strategies

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Information Desk (`help@sti.nasa.gov`)
U	U	U	UU	27	**19b. TELEPHONE NUMBER** *(Include area code)* (757) 864-9658

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18