Quantifier Elimination via Triangular Decomposition

Changbo Chen, Marc Moreno Maza
University of Western Ontario, Ontario, Canada

Quantifier Elimination (QE)

Let $F(x_1, \ldots, x_n)$ be a quantifier-ree formula in disjunctive normal form. Let $P := (Q_{k+1}x_{k+1} \cdots Q_n x_n)\, F(x_1, \ldots, x_n)$ be a prenex formula, where each Q_i is either the existential quantifier ($\exists$) or the universal quantifier ($\forall$). The goal of Quantifier Elimination (QE) is to compute a quantifier free formula $S(x_1, \ldots, x_k)$ such that $SF \Leftrightarrow PF$ holds for all $x_1, \ldots, x_k$.
As an example, consider

$$(\exists x)(\forall y)(ax^2 + bx + c) - (ay^2 + by + c) \geq 0$$

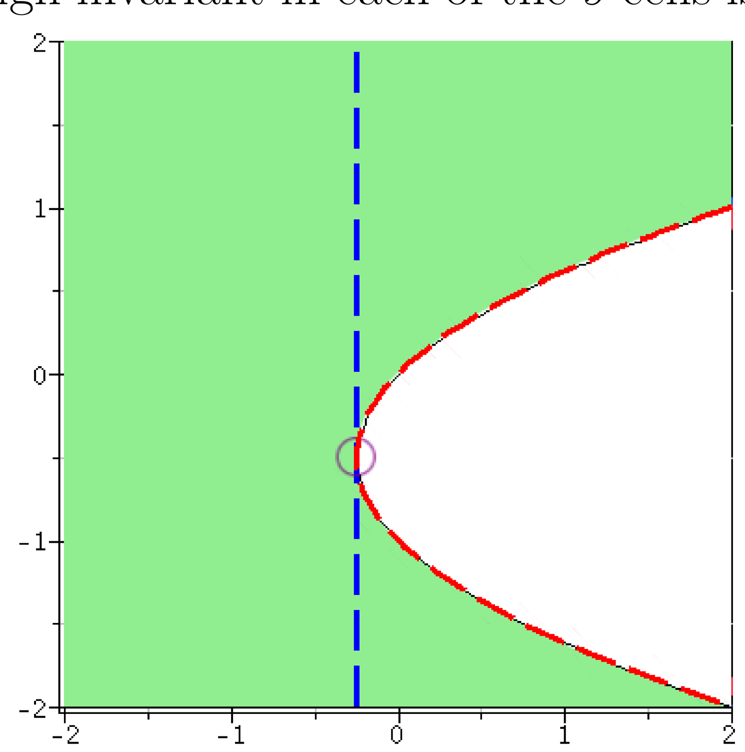
for which QE yields

$$(a < 0) \vee (a = b = 0).$$

In many research articles and computer software (QEPCAD, Mathematica, Reduce) QE is performed via cylindrical algebraic decomposition.

Cylindrical Algebraic Decomposition

A cylindrical algebraic decomposition (CAD) of $\mathbb{R}^2$ such that $y^2 + y - x$ is sign invariant in each of the 9 cells is shown below.



Motivation

In 1973, Collins introduced the first algorithm for computing CAD via a projection and lifting scheme. The projection phase relies on a global projection operator, which is a function defined independently of the input system, to eliminate variables. A strong projection operator usually produces much more polynomials than needed while a weak projection operator may fail for non-generic cases.

At ISSAC'09, the authors (with B. Xia and L. Yang) proposed a new scheme for computing CAD based on triangular decomposition (TCAD), which replaces the projection phase by computing a complex cylindrical tree. It is a natural question to ask how QE can be done via TCAD.

Main Contribution

We propose an operation which refines a given complex cylindrical tree with a new polynomial: this yields an incremental algorithm for computing the cylindrical decomposition of the complex space. An existing CAD can be refined through this operation. Moreover, QE is achieved based on refined CADs. This operation can fully take advantage of equational constraints.

Refining a Complex Cylindrical Tree

Fig.1 shows a $y^2 + x$-invariant cylindrical decomposition of $\mathbb{C}^2$, which is represented by a tree. This tree can be refined by the polynomial $y^2 + y$ into a $\{y^2 + x, y^2 + y\}$-invariant cylindrical decomposition of $\mathbb{C}^2$, see Fig. 2.

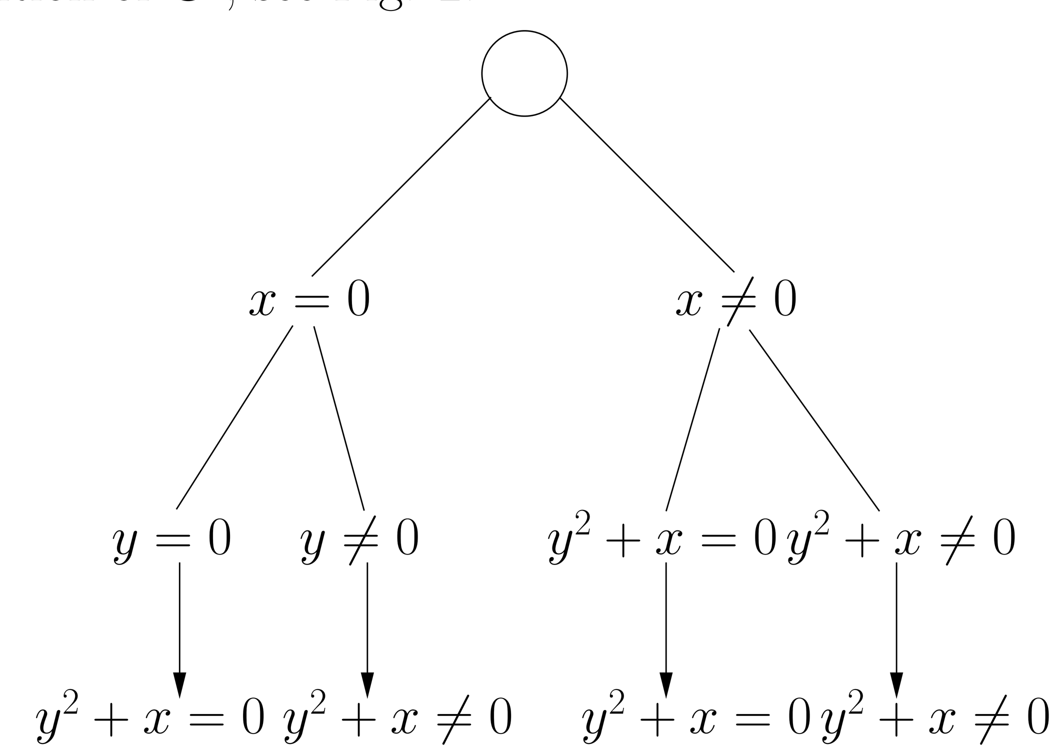


Fig.1

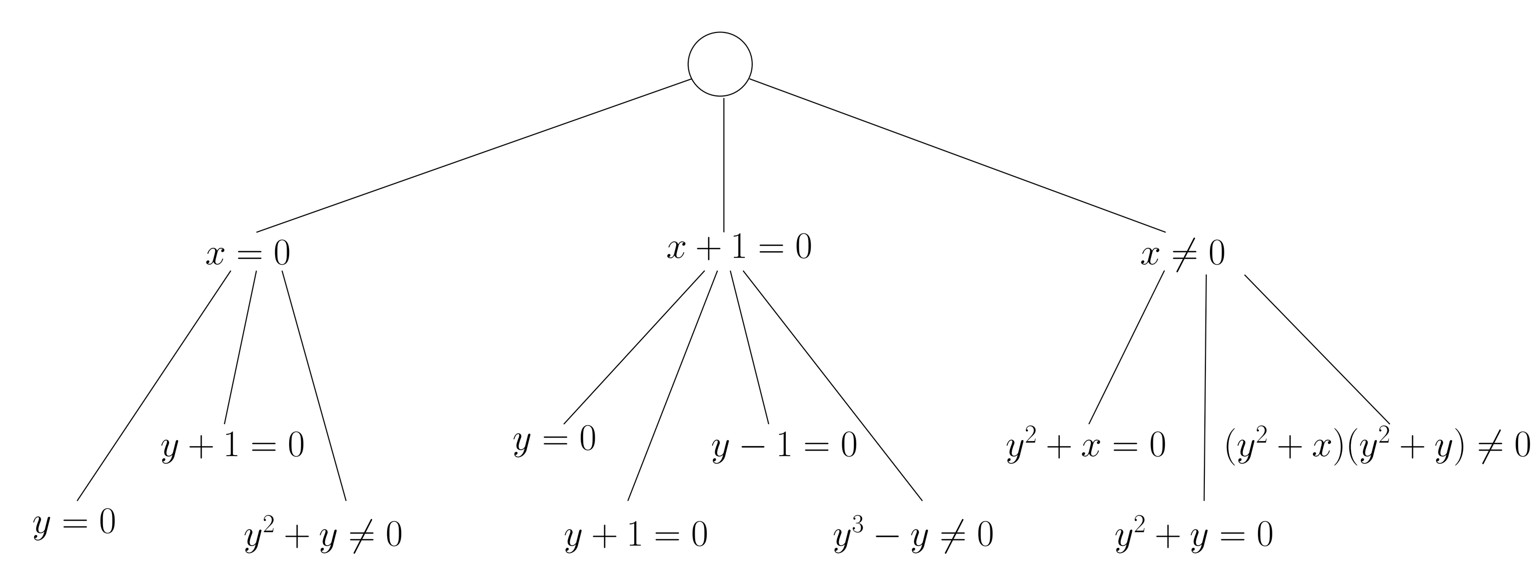


Fig. 2

Experimentation and Application

Results with a preliminary implementation in the MAPLE's RegularChains library.

Table 1. The timings for computing cylindrical decomposition of complex space

System	Recursive(t)	Incremental(t)	System	Recursive(t)	Incremental(t)
Arnborg-Lazard-rev	2933.851	472.185	Lafferriere37	171.530	20.877
Barry	42.854	1.972	lhlp5	38.362	6.956
blood-coagulation-2	246.963	31.093	McCallum-random	> 1 **hour**	51.363
cdc2-cyclin	> 1 **hour**	120.923	nql-5-4	> 1 **hour**	9.620
circles	271.452	13.684	r-5	90.453	46.270
Collins-Johnson	20.405	4.464	r-6	1608.392	1151.847
Collision	899.116	59.347	Raksanyi	2177.672	961.672
Davenport-Heintz	12.172	2.076	Rose	> 1 **hour**	38.842
GonzalezGonzalez	166.182	105.590	Solotareff-3	315.375	23.525
hereman-2	> 1 **hour**	1574.710	X-axis-ellipse	145.505	35.358
Jirstrand22	62.095	10.448	xy-5-5-1	114.627	12.336
Jirstrand42	152.145	112.735	YangBaxterRosso	73.500	2.068

Table 2. The timings for computing CAD of real space and number of cells

System	TCAD(t)	TCAD(#)	QEPCAD(t)	QEPCAD(#)
Alonso	64.220	9939	12.060	39801
Barry	7.320	1087	**Fail**	**Fail**
blood-coagulation-2	1438.769	96093	**Fail**	**Fail**
cdc2-cyclin	801.470	435	> 1 **hour**	> 1 **hour**
Collins-Johnson	13.116	3673	10.272	3673
Collision	294.462	45979	11.880	45979
Davenport-Heintz	13.044	4949	9.868	4949
hereman-2	> 1 **hour**	> 1 **hour**	37.206	6899
Jirstrand41	2.716	755	9.76	755
Jirstrand42	> 1 **hour**	> 1 **hour**	26.761	179905
Lafferriere35	4.056	801	9.980	1005
Lafferriere37	102.450	13371	**Fail**	**Fail**
lhlp2	2.444	353	**Fail**	**Fail**
McCallum-random	> 1 **hour**	> 1 **hour**	28.585	877
nql-5-4	110.750	23347	94.273	23347
r-5	1312.090	261683	> 1 **hour**	> 1 **hour**
Rose	617.106	7951	**Fail**	**Fail**
Solotareff-3	> 1 **hour**	> 1 **hour**	14.028	66675

Consider a dynamical system with a control parameter u.
We'd like to know if there exists a control Lyapunov function of the form $ax^2 + by^2$.
The equivalent QE problem is: $\forall\,(x, y)\,\exists\,(u)\left(x \neq 0 \textbf{ or } y \neq 0 \Rightarrow V > 0 \wedge \frac{d}{dt} V < 0\right)$.

The input control system.

```
> S := {diff(x(t), t) = -x(t)+u, diff(y(t),t) = -x(t) - y(t)^3};
```

$$S := \left\{\frac{d}{dt}x(t) = -x(t) + u, \frac{d}{dt}y(t) = -x(t) - y(t)^3\right\}$$

Compute $\frac{d}{dt} V$.

```
> fx := -x+u; fy := -x-y^3; V := a*x^2+b*y^2;
  Vt := diff(V, x)*fx + diff(V, y)*fy;
```

$$fx := -x + u$$
$$fy := -x - y^3$$
$$V := y^2\, b + x^2\, a$$
$$Vt := 2\, x\, a\, (-x + u) + 2\, b\, y\, (-x - y^3)$$

Call QeTcad to find conditions on a and b.

```
> RegularChains:-QuantifierElimination([[A, x, y], [E, u]], [[x<>0], [y<>0]], [[V>0, Vt<0]]);
```

$$[[0 < b, 0 < a]]$$

Verify the result for $a = 1$, $b = 4$.
Indeed $x^2 + 4\, y^2$ is a control Lyapunov function.

```
> simplify(subs({a=1,b=4}, Vt)); subs(u=4*y, %);
```

$$2\, u\, x - 2\, x^2 - 8\, y^4 - 8\, y\, x$$
$$-2\, x^2 - 8\, y^4$$