Using CDCAC for SMT inquiries with special constraints

AmirHosein Sadeghimanesh
(Joint work with Matthew England)

Coventry University

a.h.sadeghimanesh@gmail.com
ad6397@coventry.ac.uk

Maple Conference 2023

Authors supported by EPSRC grant EP/T015748/1 (The DEWCAD Project)

Introduction

- Recalling what SMT solvers do.
- Recalling CDCAC.
- Can we guide the search process in CDCAC?

SMT solvers solve questions such as the following

Let

- $x_1, \ldots, x_n$ be variables,
- $\mathbb{A}$ the set of real algebraic numbers ($\mathbb{Q} \subset \mathbb{A} \subset \mathbb{R}$),
- $f_1, \ldots, f_m \in \mathbb{A}[x_1, \ldots, x_n]$ some polynomials,
- $\prec_1, \ldots, \prec_m$ some relations from the set $\{=, \neq, <, \leq, >, \geq\}$,
- $\Phi(B_1, \ldots, B_m)$ a Boolean formula.

Then

$$\overset{?}{\exists}(x_1, \ldots, x_n) \in \mathbb{R}^n \text{ such that } \Phi(B_1, \ldots, B_m) \,|_{B_1 = f_1 \prec_1 0, \ldots, B_m = f_m \prec_m 0} \,.$$

Example

$$\overset{?}{\exists} x, y \in \mathbb{R} \text{ such that } x > 0 \wedge y > 0 \wedge x^2 + y^2 \geq 1$$

SMT solvers solve questions such as the following

Let

- $x_1, \ldots, x_n$ be variables,
- $\mathbb{A}$ the set of real algebraic numbers ($\mathbb{Q} \subset \mathbb{A} \subset \mathbb{R}$),
- $f_1, \ldots, f_m \in \mathbb{A}[x_1, \ldots, x_n]$ some polynomials,
- $\prec_1, \ldots, \prec_m$ some relations from the set $\{=, \neq, <, \leq, >, \geq\}$,
- $\Phi(B_1, \ldots, B_m)$ a Boolean formula.

Then

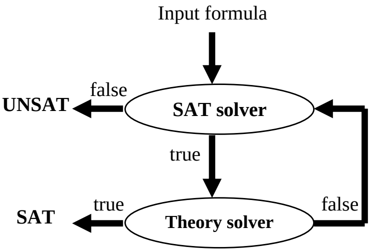
$$\stackrel{?}{\exists}(x_1, \ldots, x_n) \in \mathbb{R}^n \text{ such that } \Phi(B_1, \ldots, B_m)\,|_{B_1 = f_1 \prec_1 0, \ldots, B_m = f_m \prec_m 0}\ .$$

Example

$$\stackrel{?}{\exists} x, y \in \mathbb{R} \text{ such that } x > 0 \wedge y > 0 \wedge x^2 + y^2 \geq 1$$

$$B_1 \wedge B_2 \wedge B_3|_{B_1=(x>0), B_2=(y>0), B_3=(x^2+y^2\geq 1)}.$$

Lazy SMT



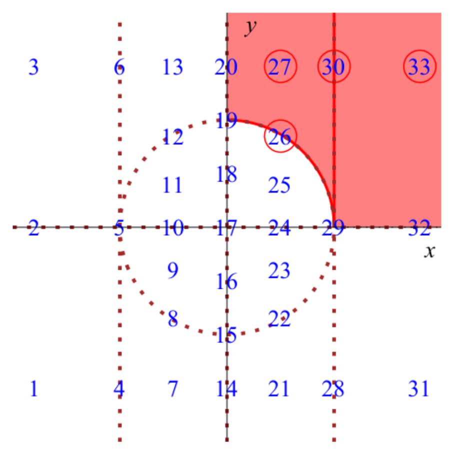
Example

Input formula: $(x^2 + y^2 < 1 \wedge y > 2) \vee (x^2 + y^2 < 1 \wedge x > 0)$
Lazy SMT sent two enquires to the theory solver, all in the form of conjunction of polynomial constraints.
1- $x^2 + y^2 < 1 \wedge y > 2 \wedge x \leq 0$ which is UNSAT.
2- $x^2 + y^2 < 1 \wedge y \leq 2 \wedge x > 0$ which is SAT with $(x, y) = (\frac{1}{2}, 0)$.

CAD as the theory solver

Example

$$x > 0 \land y > 0 \land x^2 + y^2 \geq 1$$



CDCAC as the theory solver

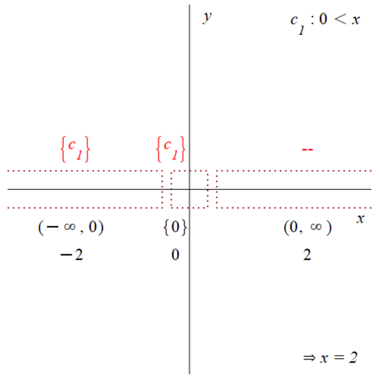
CDCAC = **C**onflict **D**riven search using **C**ylindrical **A**lgebraic **C**overings

CDCAC as the theory solver

CDCAC = **C**onflict **D**riven search using **C**ylindrical **A**lgebraic **C**overings

Example

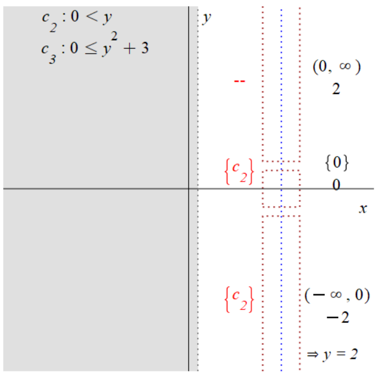
$$x > 0 \wedge y > 0 \wedge x^2 + y^2 \geq 1$$



CDCAC = **C**onflict **D**riven search using **C**ylindrical **A**lgebraic **C**overings

Example

$$x > 0 \wedge y > 0 \wedge x^2 + y^2 \geq 1$$

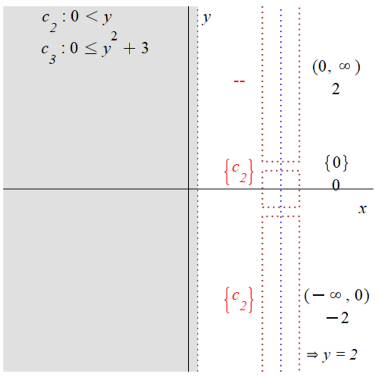


CDCAC as the theory solver

CDCAC = **C**onflict **D**riven search using **C**ylindrical **A**lgebraic **C**overings

Example

$$x > 0 \wedge y > 0 \wedge x^2 + y^2 \geq 1 \quad \text{SAT}, (x, y) = (2, 2)$$



Comparing the theory solvers

CDCAC vs CAD

```
> CodeTools:-Usage( CDCAC( [ x^2 + y^2 > 1, y > x^2 + 1 ], [ x, y ] ) );
memory used=4.32MiB, alloc change=32.00MiB, cpu time=63.00ms, real time=60.00ms, gc time=0ns
                                                  [true, [0, 3], [ ]]
```

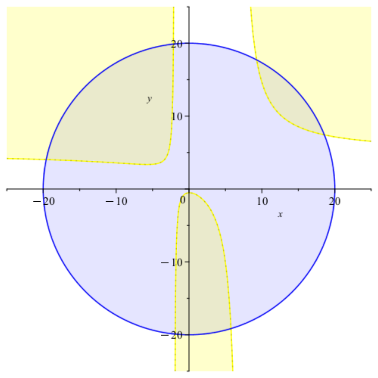
Full CAD.

```
> restart:
  with(RegularChains:-SemiAlgebraicSetTools):
  CodeTools:-Usage(CylindricalAlgebraicDecompose([x^2+(y+2)^2-1,y-x^2-1],RegularChains:-PolynomialRing([x,y]), output=cadcell)):
memory used=6.58MiB, alloc change=32.00MiB, cpu time=78.00ms, real time=81.00ms, gc time=0ns

> restart:
  with(RegularChains:-SemiAlgebraicSetTools):
  CodeTools:-Usage(CylindricalAlgebraicDecompose([x^2+(y+2)^2-1,y-x^2-1],RegularChains:-PolynomialRing([x,y]), output=piecewise)):
memory used=17.76MiB, alloc change=97.00MiB, cpu time=156.00ms, real time=278.00ms, gc time=0ns

> restart:
  with(RegularChains:-SemiAlgebraicSetTools):
  CodeTools:-Usage(CylindricalAlgebraicDecompose([x^2+(y+2)^2-1,y-x^2-1],RegularChains:-PolynomialRing([x,y]), output=list)):
memory used=6.23MiB, alloc change=32.00MiB, cpu time=63.00ms, real time=82.00ms, gc time=0ns
```

Special request 1

Guiding the CDCAC search

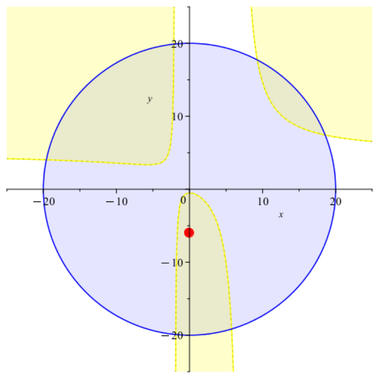
Did you notice the directions of movements in the CDCAC search steps?
Another Example: $x^2y - 5x^2 - 5xy - 2x - 14y - 7 > 0$, $x^2 + y^2 - 400 < 0$.



Special request 1

Guiding the CDCAC search

Did you notice the directions of movements in the CDCAC search steps?
Another Example: $x^2y - 5x^2 - 5xy - 2x - 14y - 7 > 0$, $x^2 + y^2 - 400 < 0$.

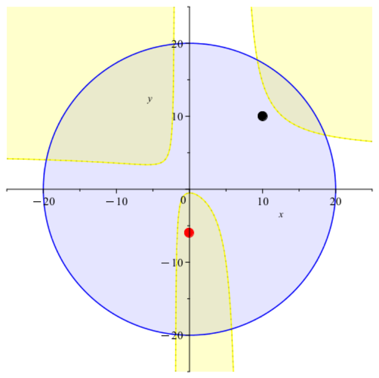


Current implementation returns $(0, -6)$.

Special request 1

Guiding the CDCAC search

Did you notice the directions of movements in the CDCAC search steps?
Another Example: $x^2y - 5x^2 - 5xy - 2x - 14y - 7 > 0$, $x^2 + y^2 - 400 < 0$.

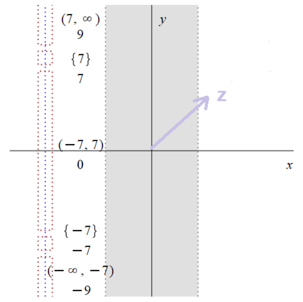


But what if we wanted a solution near a given point (the black point)?

Special request 2

Finding components of specific dimension

Did you notice the decompositions in each layer in CDCAC algorithm? There were closed (singleton sets) and open cells (open intervals). If the user wants a point from a solution component of dimension $d < n$, then we can avoid lifting up the partial points where there not enough closed cells used in its previous layers. For example we are at variable x_{d+1} and all previous variables have picked up values from open intervals, so now we can ignore the open intervals in this layer.



References

1. Erika Abrahám, James H. Davenport, Matthew England, Gereon Kremer, *Deciding the consistency of non-linear real arithmetic constraints with a conflict driven search using cylindrical algebraic coverings*. Journal of Logical and Algebraic Methods in Programming, 2021, DOI: j.jlamp.2020.100633.
2. AmirHosein Sadeghimanesh, Matthew England, *An SMT solver for non-linear real arithmetic inside Maple*. ACM Communications in Computer Algebra, 2022, DOI: 10.1145/3572867.3572880.
3. David Wilson, Russell Bradford, James Davenport, Matthew England, *Cylindrical algebraic sub-decompositions*. Mathematics in Computer Science, 2014, DOI: 10.1007/s11786-014-0191-z.

Thank you for listening.

al-Khwarizmi