A Compositional Proof System for Cylindrical Algebraic Decomposition

Jasper Nalbach Erika Ábrahám Philippe Specht Christopher W. Brown
James H. Davenport Matthew England

RWTH Aachen University

Dagstuhl Seminar 23471: The Next Generation of Deduction Systems: from Composition to Compositionality
Novemer 20–24, 2023

Satisfiability Modulo Non-linear Real Arithmetic

Is a given Boolean combination of polynomial constraints satisfiable?

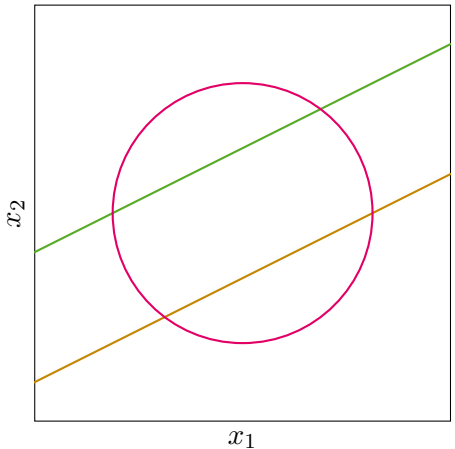
Satisfiability Modulo Non-linear Real Arithmetic

Is a given Boolean combination of polynomial constraints satisfiable?

In this talk: only conjunctions!

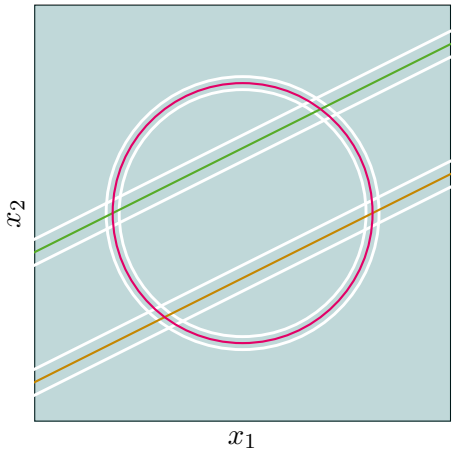
$$0.5x_1 + 0.5 - x_2 > 0 \wedge x_1^2 + x_2^2 - 1 < 0 \wedge 0.5x_1 - 0.5 - x_2 < 0$$

Cylindrical Algebraic Decomposition [Collins 1975]



$$0.5x_1 + 0.5 - x_2 > 0$$
$$\wedge x_1^2 + x_2^2 - 1 < 0$$
$$\wedge 0.5x_1 - 0.5 - x_2 < 0$$

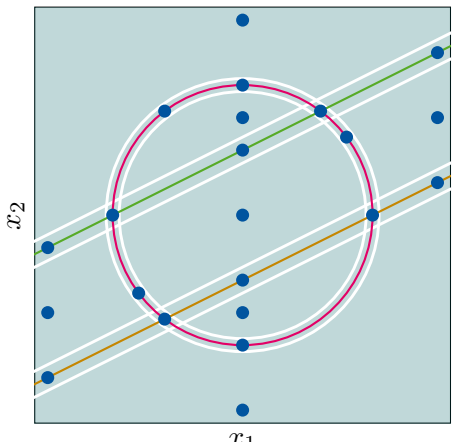
Cylindrical Algebraic Decomposition [Collins 1975]



$$0.5x_1 + 0.5 - x_2 > 0$$
$$\wedge x_1^2 + x_2^2 - 1 < 0$$
$$\wedge 0.5x_1 - 0.5 - x_2 < 0$$

Decomposition into sign-invariant cells!

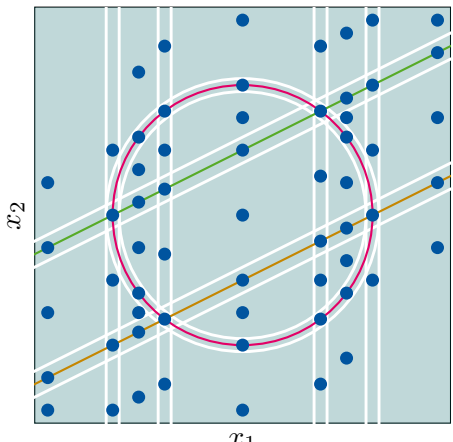
Cylindrical Algebraic Decomposition [Collins 1975]



$$0.5x_1 + 0.5 - x_2 > 0$$
$$\wedge x_1^2 + x_2^2 - 1 < 0$$
$$\wedge 0.5x_1 - 0.5 - x_2 < 0$$

Decomposition into sign-invariant cells!

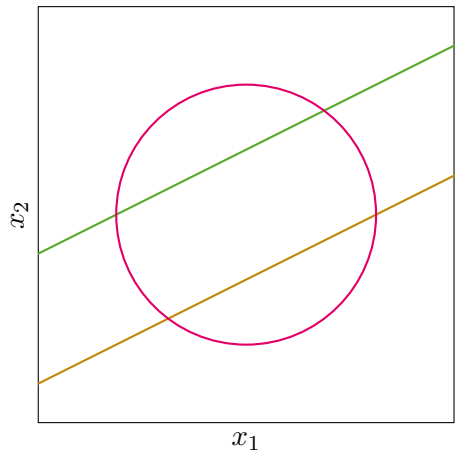
Cylindrical Algebraic Decomposition [Collins 1975]



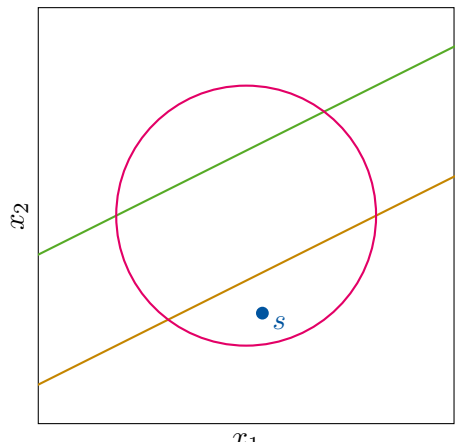
$$0.5x_1 + 0.5 - x_2 > 0$$
$$\wedge x_1^2 + x_2^2 - 1 < 0$$
$$\wedge 0.5x_1 - 0.5 - x_2 < 0$$

Decomposition into sign-invariant cells!

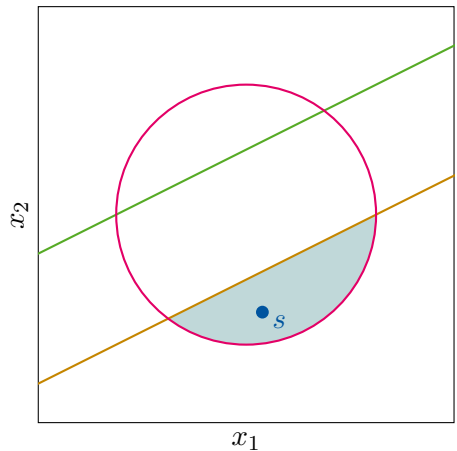
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



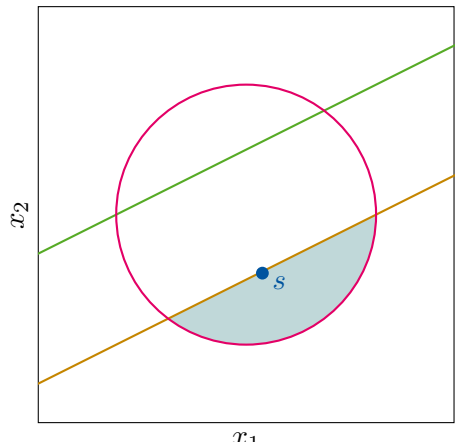
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



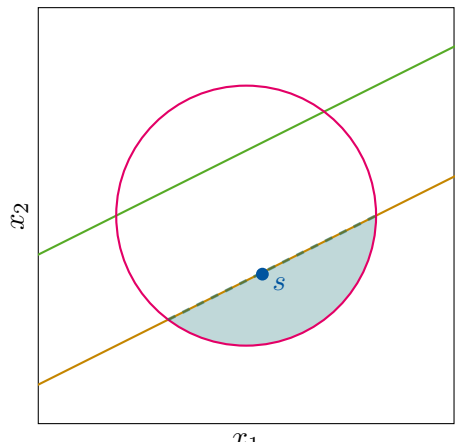
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



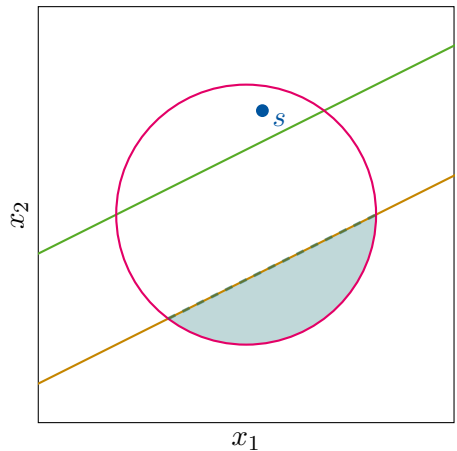
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



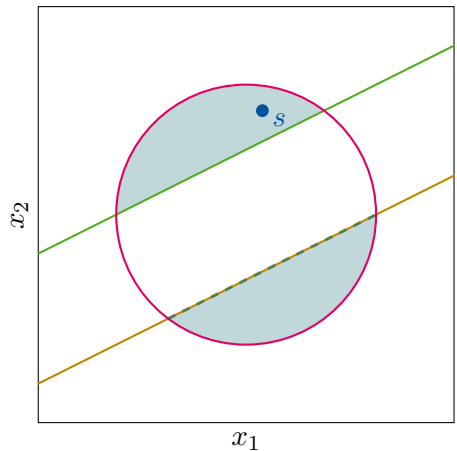
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



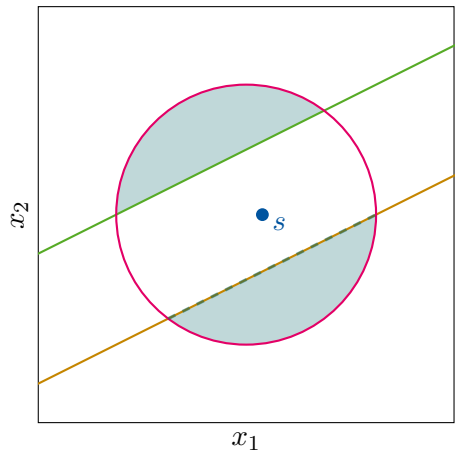
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



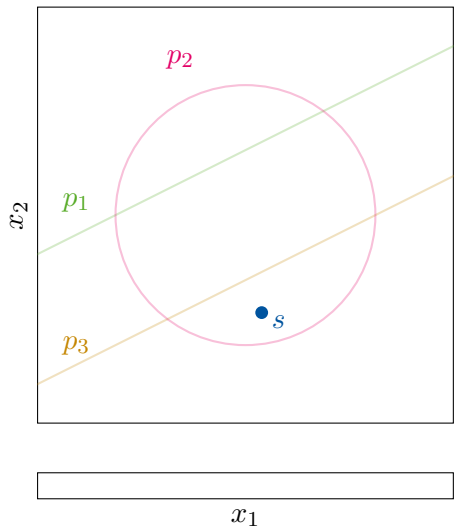
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



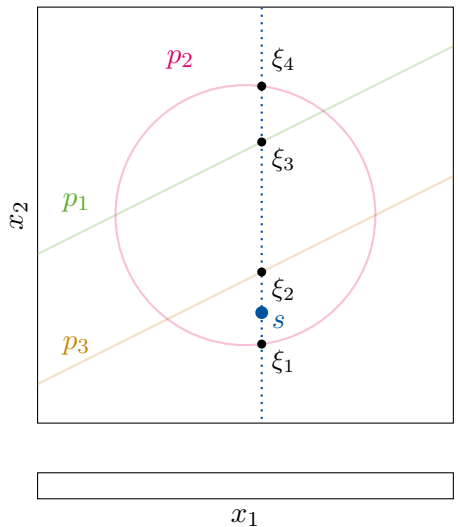
CAD-based Sampling Algorithms: NLSAT [Jovanović, De Moura 2012], NuCAD [Brown 2015], CAIC [Ábrahám et al. 2021], Single Cells [Brown, Košta 2015]



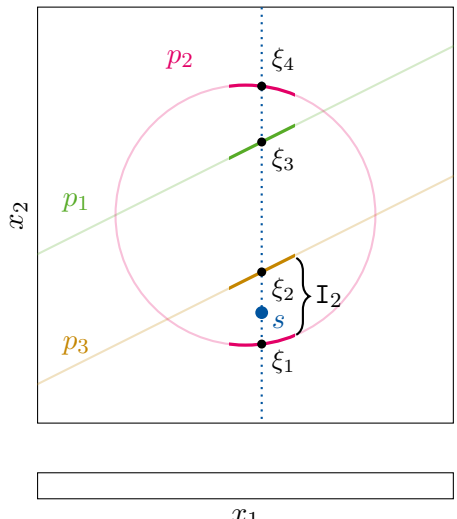
Levelwise Single Cell Construction [Nalbach et al. 2023]



Levelwise Single Cell Construction [Nalbach et al. 2023]



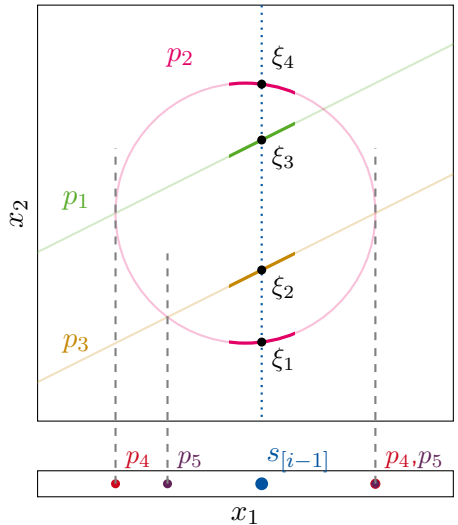
Levelwise Single Cell Construction [Nalbach et al. 2023]



p_1, p_2, p_3 sign-invariant $\Longleftarrow$

- $x_2 \in \mathtt{I}_2 = (\xi_1, \xi_2) = (\mathrm{root}_{x_2}[p_2, 1], \mathrm{root}_{x_2}[p_3, 1])$
- $\xi_1 \prec \xi_2$
- $\xi_2 \prec \xi_3$
- $\xi_2 \prec \xi_4$

Levelwise Single Cell Construction [Nalbach et al. 2023]



$\xi_1 \prec \xi_2 \impliedby$

- $\mathrm{res}_{x_2}(p_3, p_2) = p_5$ sign-invariant
- ξ_1 well-defined

$\xi_2 \prec \xi_3 \impliedby$

- $\mathrm{res}_{x_2}(p_3, p_1) \hat{=} 1$ sign-invariant (trivial)

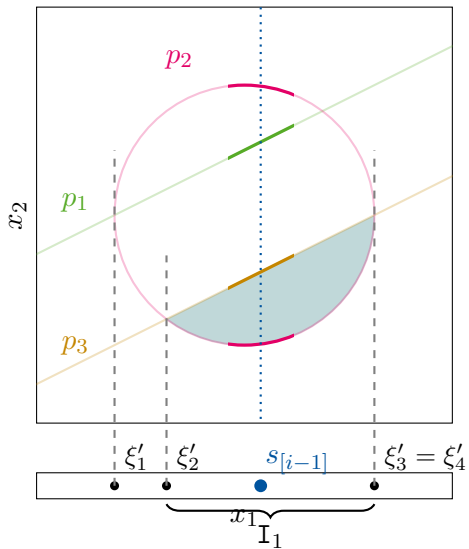
$\xi_2 \prec \xi_4 \impliedby$

- $\mathrm{res}_{x_2}(p_3, p_2) = p_5$ sign-invariant
- ξ_4 well-defined

ξ_1 well-defined, ξ_4 well-defined $\impliedby$

- $\mathrm{disc}_{x_2}(p_2) = p_4$ sign-invariant

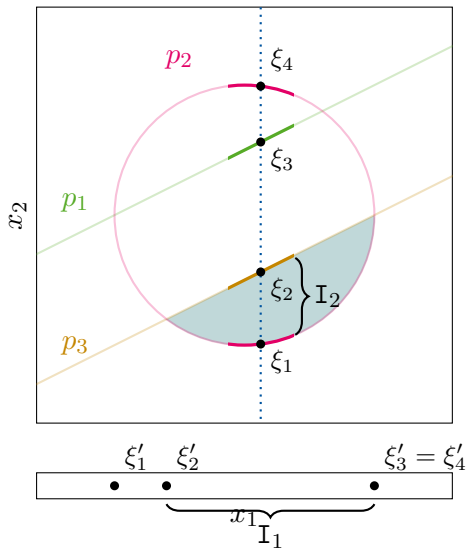
Levelwise Single Cell Construction [Nalbach et al. 2023]



p_4, p_5 sign-invariant

- $x_1 \in \mathtt{I}_1 = (\xi'_2, \xi'_3) = (\mathrm{root}_{x_2}[p_5, 1], \mathrm{root}_{x_2}[p_5, 2])$
- $\xi'_1 \prec \xi'_2$ (trivial)
- $\xi'_3 \approx \xi'_4$ (trivial)

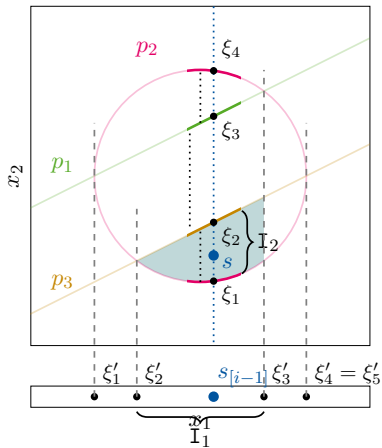
Levelwise Single Cell Construction [Nalbach et al. 2023]



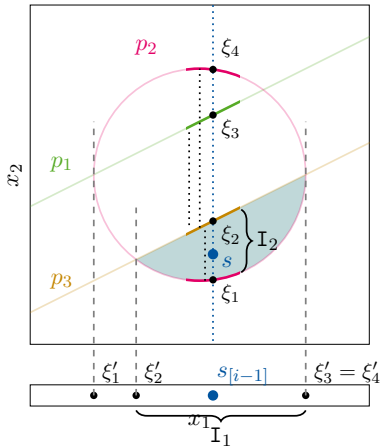
p_1, p_2, p_3 sign-invariant $\Longleftarrow$

- $x_2 \in \mathtt{I}_2$
- $x_1 \in \mathtt{I}_1$

Single Cell Construction: Root Orderings



$$\xi_1 \preceq \xi_2 \preceq \xi_3 \preceq \xi_4$$



$$\xi_1 \preceq \xi_2 \begin{matrix} \preceq \xi_3 \\ \preceq \xi_4 \end{matrix}$$

Single Cell Construction: Root Orderings

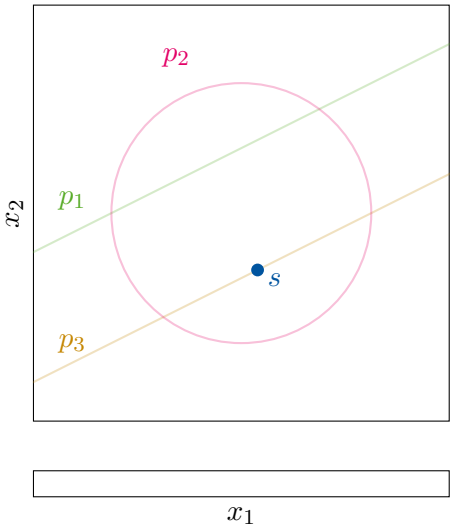
- McCallum 1998

  Theorem 2 *Let $r \geq 2$. Let $f(x, x_r)$ be a polynomial in $\mathbb{R}[x, x_r]$ of positive degree. Let $D(x)$ be the discriminant of $f(x, x_r)$ and suppose that $D(x)$ is a nonzero polynomial. Let S be a connected submanifold of $\mathbb{R}^{r-1}$ on which f is degree-invariant and does not vanish identically, and in which D is order-invariant. Then f is analytic delineable on S and is order-invariant in each f-section over S.*

- Brown 2001

  THEOREM 3.1. *Let $f \in \mathbb{R}[\overline{x}, z]$ be a $(k+1)$-variate polynomial of positive degree n in the variable z with discriminant $D(\overline{x}) \neq 0$. Let S be a connected analytic submanifold of $\mathbb{R}^k$ in which D is order-invariant, the leading coefficient of f is sign-invariant, and such that f vanishes identically at no point in S. f is degree-invariant on S.*

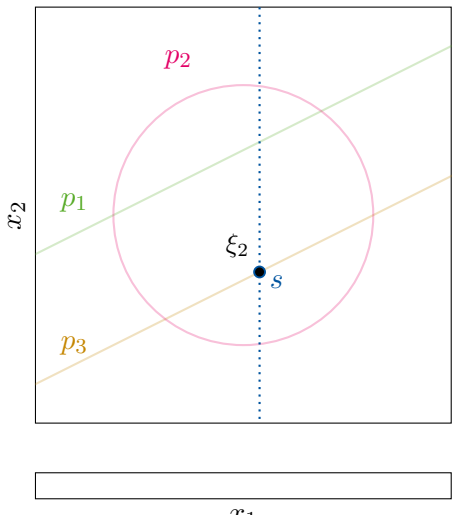
Single Cell Construction: Equational Constraints



McCallum 1999

Theorem 2.2 *Let $r \geq 2$, let $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ be real polynomials of positive degrees in the main variable x_r, let $R(x_1, \ldots, x_{r-1})$ be the resultant of f and g, and suppose that $R \neq 0$. Let S be a connected subset of $\mathbb{R}^{r-1}$ on which f is delineable and in which R is order-invariant. Then g is sign-invariant in each section of f over S.*

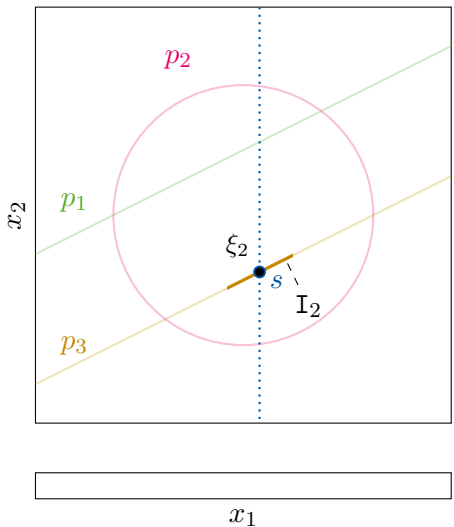
Single Cell Construction: Equational Constraints



McCallum 1999

Theorem 2.2 *Let $r \geq 2$, let $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ be real polynomials of positive degrees in the main variable x_r, let $R(x_1, \ldots, x_{r-1})$ be the resultant of f and g, and suppose that $R \neq 0$. Let S be a connected subset of $\mathbb{R}^{r-1}$ on which f is delineable and in which R is order-invariant. Then g is sign-invariant in each section of f over S.*

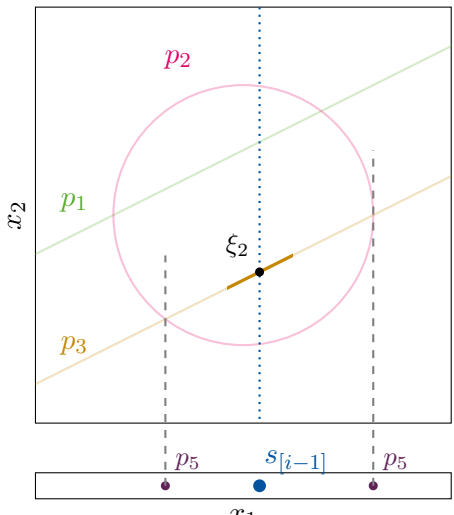
Single Cell Construction: Equational Constraints



McCallum 1999

Theorem 2.2 *Let $r \geq 2$, let $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ be real polynomials of positive degrees in the main variable x_r, let $R(x_1, \ldots, x_{r-1})$ be the resultant of f and g, and suppose that $R \neq 0$. Let S be a connected subset of $\mathbb{R}^{r-1}$ on which f is delineable and in which R is order-invariant. Then g is sign-invariant in each section of f over S.*

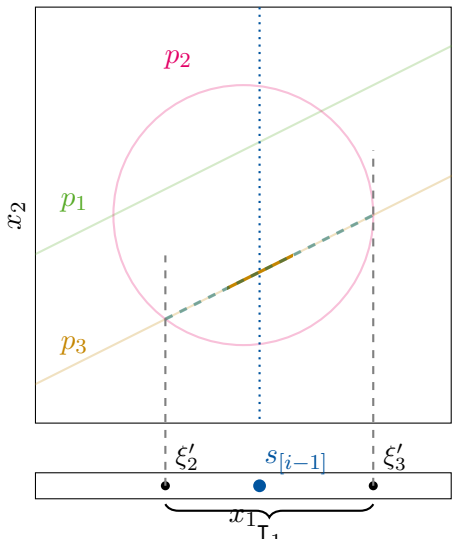
Single Cell Construction: Equational Constraints



McCallum 1999

Theorem 2.2 *Let $r \geq 2$, let $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ be real polynomials of positive degrees in the main variable x_r, let $R(x_1, \ldots, x_{r-1})$ be the resultant of f and g, and suppose that $R \neq 0$. Let S be a connected subset of $\mathbb{R}^{r-1}$ on which f is delineable and in which R is order-invariant. Then g is sign-invariant in each section of f over S.*

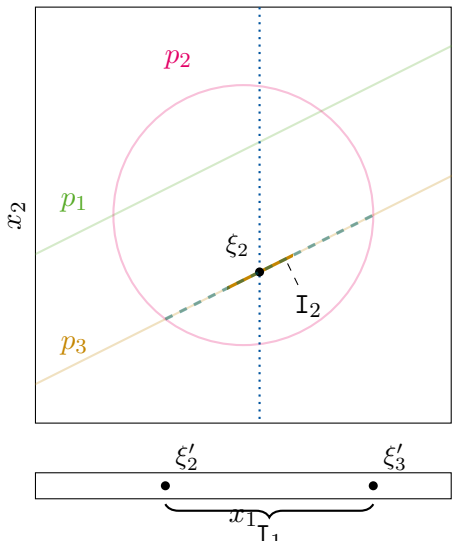
Single Cell Construction: Equational Constraints



McCallum 1999

Theorem 2.2 *Let $r \geq 2$, let $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ be real polynomials of positive degrees in the main variable x_r, let $R(x_1, \ldots, x_{r-1})$ be the resultant of f and g, and suppose that $R \neq 0$. Let S be a connected subset of $\mathbb{R}^{r-1}$ on which f is delineable and in which R is order-invariant. Then g is sign-invariant in each section of f over S.*

Single Cell Construction: Equational Constraints



McCallum 1999

Theorem 2.2 *Let $r \geq 2$, let $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ be real polynomials of positive degrees in the main variable x_r, let $R(x_1, \ldots, x_{r-1})$ be the resultant of f and g, and suppose that $R \neq 0$. Let S be a connected subset of $\mathbb{R}^{r-1}$ on which f is delineable and in which R is order-invariant. Then g is sign-invariant in each section of f over S.*

Single Cell Construction: Nullifications

- Proof system is based on McCallum's projection operator [McCallum 1985; McCallum 1992]

Single Cell Construction: Nullifications

- Proof system is based on McCallum's projection operator [McCallum 1985; McCallum 1992]
- which is incomplete:

Single Cell Construction: Nullifications

- Proof system is based on McCallum's projection operator [McCallum 1985; McCallum 1992]
- which is incomplete:
- A polynomial $p \in \mathbb{Q}[x_1, \ldots, x_{n-1}][x_n]$ (with non-zero leading coefficient) is nullified on some $s \in \mathbb{R}^{n-1}$ iff $p(s, x_n) = 0$. I.e. its roots cannot be determined!

Single Cell Construction: Nullifications

- Proof system is based on McCallum's projection operator [McCallum 1985; McCallum 1992]
- which is incomplete:
- A polynomial $p \in \mathbb{Q}[x_1, \ldots, x_{n-1}][x_n]$ (with non-zero leading coefficient) is nullified on some $s \in \mathbb{R}^{n-1}$ iff $p(s, x_n) = 0$. I.e. its roots cannot be determined!
- Lazard's operator [Lazard 1995, McCallum et al. 2019] can deal with nullifications.

Proof System for Single Cell Construction

Proof System for Single Cell Construction

- A property of level i is a function $q : \{R \mid R \subseteq \mathbb{R}^i\} \to \mathbb{B}$.

Proof System for Single Cell Construction

- A property of level i is a function $q : \{R \mid R \subseteq \mathbb{R}^i\} \to \mathbb{B}$.
  Example: Let $i \in \mathbb{N}$, $R \subseteq \mathbb{R}^i$, $p \in \mathbb{Q}[x_1, \ldots, x_i]$.
  - *connected*$(i)(R) = 1$ iff R is connected.
  - *sgn_inv*$(p)(R) = 1$ iff p is sign-invariant on R.

Proof System for Single Cell Construction

- A property of level i is a function $q : \{R \mid R \subseteq \mathbb{R}^i\} \to \mathbb{B}$.
  Example: Let $i \in \mathbb{N}$, $R \subseteq \mathbb{R}^i$, $p \in \mathbb{Q}[x_1, \ldots, x_i]$.
  - *connected*$(i)(R) = 1$ iff R is connected.
  - *sgn_inv*$(p)(R) = 1$ iff p is sign-invariant on R.
- A rule of inference is of the form $P_1, \ldots, P_k \vdash Q$, where $P_1, \ldots, P_k$ are the antecedents and Q is the consequent.

Proof System for Single Cell Construction

- A property of level i is a function $q : \{R \mid R \subseteq \mathbb{R}^i\} \to \mathbb{B}$.
  Example: Let $i \in \mathbb{N}$, $R \subseteq \mathbb{R}^i$, $p \in \mathbb{Q}[x_1, \ldots, x_i]$.
  - *connected*$(i)(R) = 1$ iff R is connected.
  - *sgn_inv*$(p)(R) = 1$ iff p is sign-invariant on R.
- A rule of inference is of the form $P_1, \ldots, P_k \vdash Q$, where $P_1, \ldots, P_k$ are the antecedents and Q is the consequent.
  Example: Let $i \in \mathbb{N}_{>0}$, $R \subseteq \mathbb{R}^i$, $p \in \mathbb{Q}[x_1, \ldots, x_i]$, $\text{factors}(p) = \{q_1, \ldots, q_k\}$.

$$\textit{sgn\_inv}(q_1)(R),\ \ldots,\ \textit{sgn\_inv}(q_k)(R) \qquad \vdash \textit{sgn\_inv}(p)(R)$$

Proof System for Single Cell Construction

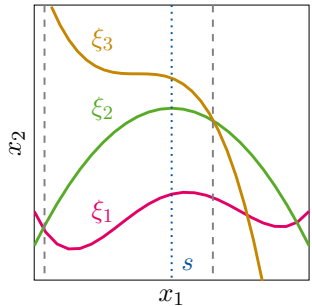
- A property of level i is a function $q : \{R \mid R \subseteq \mathbb{R}^i\} \to \mathbb{B}$.
  Example: Let $i \in \mathbb{N}$, $R \subseteq \mathbb{R}^i$, $p \in \mathbb{Q}[x_1, \ldots, x_i]$.
  - *connected*$(i)(R) = 1$ iff R is connected.
  - *sgn_inv*$(p)(R) = 1$ iff p is sign-invariant on R.
- A rule of inference is of the form $P_1, \ldots, P_k \vdash Q$, where $P_1, \ldots, P_k$ are the antecedents and Q is the consequent.
  Example: Let $i \in \mathbb{N}_{>0}$, $R \subseteq \mathbb{R}^i$, $p \in \mathbb{Q}[x_1, \ldots, x_i]$, $\text{factors}(p) = \{q_1, \ldots, q_k\}$.

$$\textit{sgn\_inv}(q_1)(R),\ \ldots,\ \textit{sgn\_inv}(q_k)(R) \qquad \vdash \textit{sgn\_inv}(p)(R)$$

- We require that the $P_1, \ldots, P_k$ are smaller than Q w.r.t. an ordering $\lhd$.

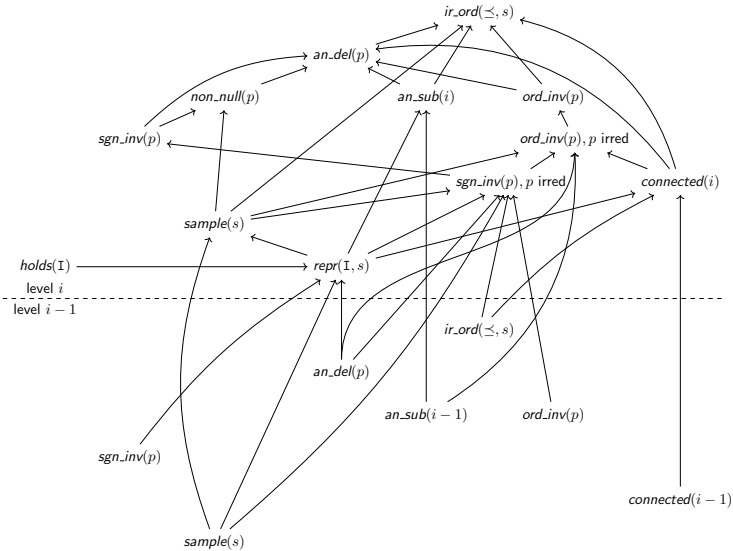
Proof System: Root Orderings

$i \in \mathbb{N}$, $R \subseteq \mathbb{R}^i$, $s \in \mathbb{R}^i$,
$\preceq$ an indexed root ordering of level $i+1$,
$\xi.p$ is irreducible for all $\xi \in \mathrm{dom}(\preceq)$,
$\preceq$ matches s



$$\begin{aligned} &\mathit{sample}(s)(R),\ \mathit{an\_sub}(i)(R),\ \mathit{connected}(i)(R),\\ &\forall \xi \in \mathrm{dom}(\preceq).\ \mathit{an\_del}(\xi.p)(R),\\ &\forall (\xi, \xi') \in \preceq .\ \mathit{ord\_inv}(\mathrm{res}_{x_{i+1}}(\xi.p, \xi'.p))(R) \end{aligned} \qquad \vdash \mathit{ir\_ord}(\preceq, s)(R)$$

Proof System: Dependencies of Properties by Proof Rules



Proof System: The Algorithm

Algorithm 1: `construct_interval`(i, Q, s)

Input : set of properties Q of level i, $s \in \mathbb{R}^i$
Output: interval `I` of level i, and properties Q' of level $i-1$ which imply Q

1 **order** Q according to $\lhd$
2 **replace** $q \in Q$ using proof rules w.r.t. s until $q = \mathit{sgn\_inv}(p)$ for an irreducible p
3 $\Xi :=$ symbolic roots of $\{p \mid \mathit{sgn\_inv}(p) \in Q\}$ in x_i
4 **choose** representation $(\mathtt{I}, \preceq)$ of Ξ with respect to s
5 **replace** $q \in Q$ using proof rules w.r.t. $s, \mathtt{I}, \preceq$ until $q = \mathit{holds}(\mathtt{I})$
6 **return** `I`, $(Q \setminus \{\mathit{holds}(\mathtt{I})\})$

Proof System: The Algorithm

Algorithm 2: `construct_interval`(i, Q, s)

Input : set of properties Q of level i, $s \in \mathbb{R}^i$
Output: interval $\mathtt{I}$ of level i, and properties Q' of level $i-1$ which imply Q

1 **order** Q according to $\lhd$
2 **replace** $q \in Q$ using proof rules w.r.t. s until $q = \mathit{sgn\_inv}(p)$ for an irreducible p
3 $\Xi :=$ symbolic roots of $\{p \mid \mathit{sgn\_inv}(p) \in Q\}$ in x_i
4 **choose** representation $(\mathtt{I}, \preceq)$ of Ξ with respect to s
5 **replace** $q \in Q$ using proof rules w.r.t. $s, \mathtt{I}, \preceq$ until $q = \mathit{holds}(\mathtt{I})$
6 **return** $\mathtt{I}$, $(Q \setminus \{\mathit{holds}(\mathtt{I})\})$

Proof System: The Algorithm

Algorithm 3: `construct_interval`(i, Q, s)

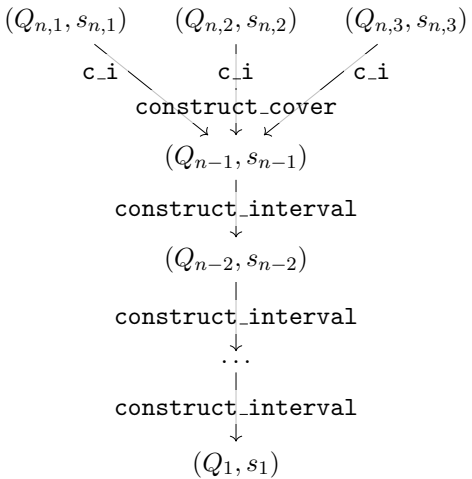
Input : set of properties Q of level i, $s \in \mathbb{R}^i$
Output: interval `I` of level i, and properties Q' of level $i-1$ which imply Q

1 **order** Q according to $\lhd$
2 **replace** $q \in Q$ using proof rules w.r.t. s until $q = \mathit{sgn\_inv}(p)$ for an irreducible p
3 $\Xi :=$ symbolic roots of $\{p \mid \mathit{sgn\_inv}(p) \in Q\}$ in x_i
4 **choose** representation $(\mathtt{I}, \preceq)$ of Ξ with respect to s
5 **replace** $q \in Q$ using proof rules w.r.t. $s, \mathtt{I}, \preceq$ until $q = \mathit{holds}(\mathtt{I})$
6 **return** `I`, $(Q \setminus \{\mathit{holds}(\mathtt{I})\})$

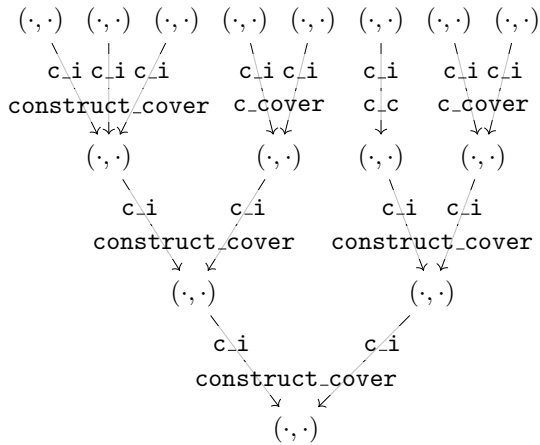
- calling `construct_interval` on $Q = \{\mathit{sgn\_inv}(p)\}$ iteratively constructs a sign-invariant cell

Applications: NLSAT and CAIC

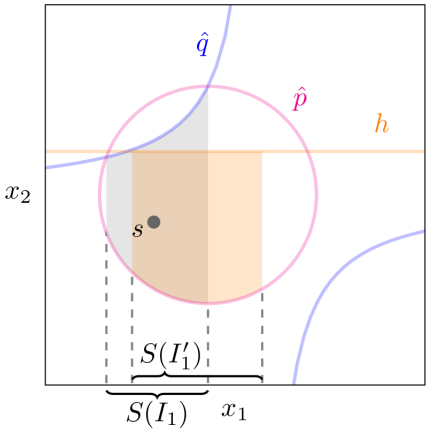
NLSAT [Jovanović, De Moura 2012]



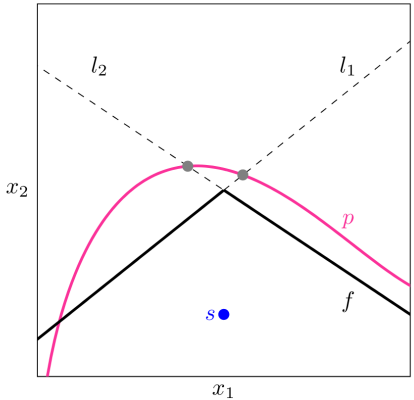
CAIC [Ábrahám et al. 2012]



Applications: Under-Approximations of Cells



Linear under-approximation
(Valentin Promies)



Piecewise linear under-approximation
(Paul Wagner)

Applications: Generation of Proofs for Real Algebra

$$
\begin{aligned}
\boxed{\mathit{sgn\_inv}(p_1)} &\to \mathit{sample}(\boxed{s}), \mathit{repr}(\boxed{\mathtt{I}_2}, s_1), \mathit{ir\_ord}(\boxed{\preceq}, s_1), \mathit{an\_del}(p_1), \mathit{an\_sub}(1), \mathit{connected}(1)\\
\boxed{\mathit{sgn\_inv}(p_2)} &\to \mathit{sample}(\boxed{s}), \mathit{repr}(\boxed{\mathtt{I}_2}, s_1), \mathit{ir\_ord}(\boxed{\preceq}, s_1), \mathit{an\_del}(p_2), \mathit{an\_sub}(1), \mathit{connected}(1)\\
\boxed{\mathit{sgn\_inv}(p_3)} &\to \mathit{sample}(\boxed{s}), \mathit{repr}(\boxed{\mathtt{I}_2}, s_1), \mathit{ir\_ord}(\boxed{\preceq}, s_1), \mathit{an\_del}(p_3), \mathit{an\_sub}(1), \mathit{connected}(1)\\
\mathit{sample}(\boxed{s}) &\to \mathit{repr}(\boxed{\mathtt{I}_2}, s_1), \mathit{sample}(s_1)\\
\mathit{repr}(\boxed{\mathtt{I}_2}, s_1) &\to \boxed{R = \mathit{setOf}(R{\downarrow_{[1]}}), \boxed{\mathtt{I}_2}}, \mathit{an\_del}(p_2), \mathit{an\_del}(p_3), \mathit{sample}(s_1)\\
\mathit{ir\_ord}(\boxed{\preceq}, s_1) &\to \mathit{an\_del}(p_1), \mathit{an\_del}(p_2), \mathit{an\_del}(p_3), \mathit{ord\_inv}(\mathrm{res}_{x_2}(p_3, p_1)), \mathit{ord\_inv}(\mathrm{res}_{x_2}(p_3, p_2)),\\
&\qquad \mathit{an\_sub}(1), \mathit{connected}(1), \mathit{sample}(s_1)\\
\mathit{an\_del}(p_1) &\to \mathit{non\_null}(p_1), \mathit{ord\_inv}(\mathrm{disc}_{x_2}(p_1)), \mathit{an\_sub}(1), \mathit{connected}(1), \mathit{sgn\_inv}(\mathrm{ldcf}_{x_2}(p_1))\\
\mathit{an\_del}(p_2) &\to \mathit{non\_null}(p_2), \mathit{ord\_inv}(\mathrm{disc}_{x_2}(p_2)), \mathit{an\_sub}(1), \mathit{connected}(1), \mathit{sgn\_inv}(\mathrm{ldcf}_{x_2}(p_2))\\
\mathit{an\_del}(p_3) &\to \mathit{non\_null}(p_3), \mathit{ord\_inv}(\mathrm{disc}_{x_2}(p_3)), \mathit{an\_sub}(1), \mathit{connected}(1), \mathit{sgn\_inv}(\mathrm{ldcf}_{x_2}(p_3))
\end{aligned}
$$

$$
\begin{aligned}
\mathit{non\_null}(p_1) &\to \mathit{trivial}\\
\mathit{non\_null}(p_2) &\to \mathit{trivial}\\
\mathit{non\_null}(p_3) &\to \mathit{trivial}\\
\mathit{ord\_inv}(\mathrm{disc}_{x_2}(p_1)) &\to \mathit{trivial}\\
\mathit{ord\_inv}(\mathrm{disc}_{x_2}(p_2)) &\to \mathit{sgn\_inv}(p_4), \mathit{sample}(s_1)\\
\mathit{ord\_inv}(\mathrm{disc}_{x_2}(p_3)) &\to \mathit{trivial}\\
\mathit{sgn\_inv}(p_4) &\to \mathit{repr}(\boxed{\mathtt{I}_1})\\
\mathit{ord\_inv}(\mathrm{res}_{x_2}(p_3, p_1)) &\to \mathit{trivial}\\
\mathit{ord\_inv}(\mathrm{res}_{x_2}(p_3, p_2)) &\to \mathit{sgn\_inv}(p_5), \mathit{sample}(s_1)
\end{aligned}
\qquad
\begin{aligned}
\mathit{sgn\_inv}(p_5) &\to \mathit{repr}(\boxed{\mathtt{I}_1})\\
\mathit{sgn\_inv}(\mathrm{ldcf}_{x_2}(p_1)) &\to \mathit{trivial}\\
\mathit{sgn\_inv}(\mathrm{ldcf}_{x_2}(p_2)) &\to \mathit{trivial}\\
\mathit{sgn\_inv}(\mathrm{ldcf}_{x_2}(p_3)) &\to \mathit{trivial}\\
\mathit{an\_sub}(1) &\to \mathit{repr}(\boxed{\mathtt{I}_1})\\
\mathit{connected}(1) &\to \mathit{trivial}\\
\mathit{sample}(s_1) &\to \mathit{repr}(\boxed{\mathtt{I}_1})\\
\mathit{repr}(\boxed{\mathtt{I}_1}) &\to \boxed{R{\downarrow_{[1]}} = \mathit{setOf}(\boxed{\mathtt{I}_1})}
\end{aligned}
$$

Conclusion

- Benefits

Conclusion

- Benefits
  - Complex algorithm is maintainable

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations
  - $\Rightarrow$ Compositional!

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations
  - $\Rightarrow$ Compositional!
- Learnings

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations
  - $\Rightarrow$ Compositional!
- Learnings
  - Subtleties of theory get visible

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations
  - $\Rightarrow$ Compositional!
- Learnings
  - Subtleties of theory get visible
  - Rigorous work, no shortcuts possible

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations
  - $\Rightarrow$ Compositional!
- Learnings
  - Subtleties of theory get visible
  - Rigorous work, no shortcuts possible
  - Not everything is automatic, properties and proof rules are carefully designed

Conclusion

- Benefits
  - Complex algorithm is maintainable
  - Modular: Heuristics are separated from the correctness, applicable in various algorithms
  - Higher trust into the code
  - During this work, two new concepts for CAD theory arose
  - Extensible: Proof system allows for many optimizations
  - $\Rightarrow$ Compositional!
- Learnings
  - Subtleties of theory get visible
  - Rigorous work, no shortcuts possible
  - Not everything is automatic, properties and proof rules are carefully designed
  - Sometimes technical, hard to present