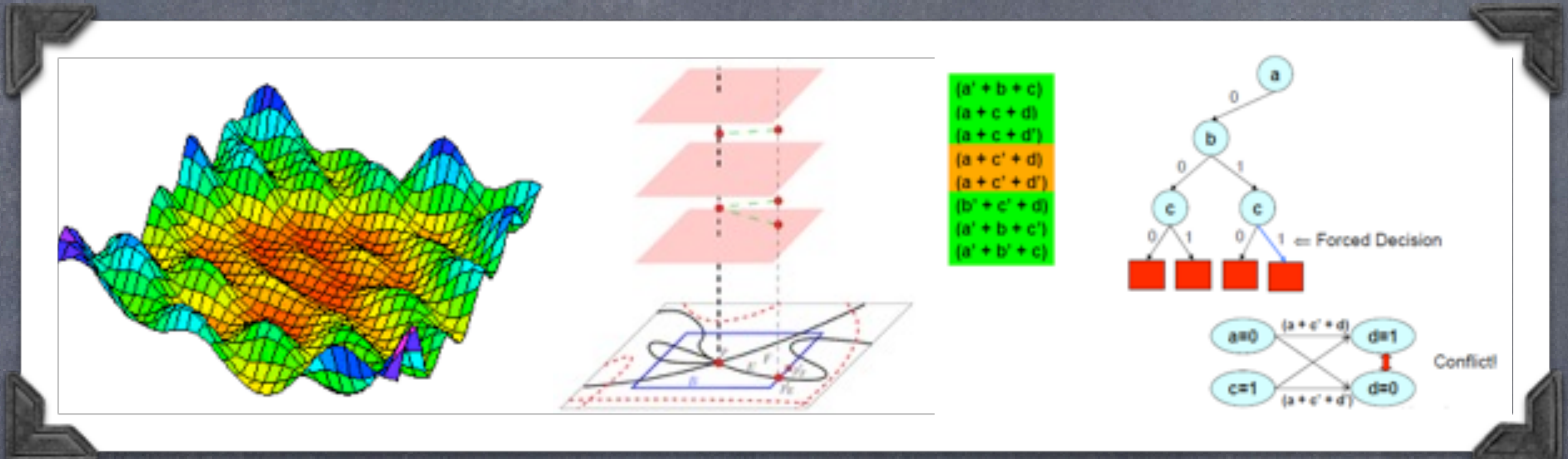
Exact Global Optimization on Demand



Leonardo de Moura & Grant Olney Passmore

Microsoft Research, Redmond, USA

LFCS, Edinburgh, and Clare Hall, Cambridge, UK

COST Action Meeting, IMDEA Software, Madrid

October, 2013

Big Picture

- A *lazy* (CDCL-like) approach to *exact* nonlinear global optimization over the real numbers

Big Picture

- A *lazy* (CDCL-like) approach to *exact* nonlinear global optimization over the real numbers
- Three main conceptual ingredients:

Big Picture

- A *lazy* (CDCL-like) approach to *exact* nonlinear global optimization over the real numbers
- Three main conceptual ingredients:

CAD-based approach to optimization

eager method for nonlinear optimization, in Mathematica v9.x

Big Picture

- A *lazy* (CDCL-like) approach to *exact* nonlinear global optimization over the real numbers
- Three main conceptual ingredients:

CAD-based approach to optimization

eager method for nonlinear optimization, in Mathematica v9.x

nlsat/mcsat - existential CAD `on demand'

lazy CDCL-like approach to Exists RCF, in Z3 (Jovanović - de Moura, 2012)

Big Picture

- A *lazy* (CDCL-like) approach to *exact* nonlinear global optimization over the real numbers
- Three main conceptual ingredients:

CAD-based approach to optimization

eager method for nonlinear optimization, in Mathematica v9.x

nlsat/mcsat - existential CAD `on demand'

lazy CDCL-like approach to Exists RCF, in Z3 (Jovanović - de Moura, 2012)

computable nonstandard RCFs

computable RCFs containing infinitesimals (de Moura - Passmore, 2013)

Big Picture

- A *lazy* (CDCL-like) approach to *exact* nonlinear global optimization over the real numbers
- Three main conceptual ingredients:

CAD-based approach to optimization

eager method for nonlinear optimization, in Mathematica v9.x

nlsat/mcsat - existential CAD `on demand'

lazy CDCL-like approach to Exists RCF, in Z3 (Jovanović - de Moura, 2012)

computable nonstandard RCFs

computable RCFs containing infinitesimals (de Moura - Passmore, 2013)

- A *practical application* of nonstandard models!

Exact Global Optimization

$$\begin{aligned} &\underset{\vec{x}}{\text{minimize}} && f(\vec{x}) \\ &\text{subject to} && \bigwedge_{i=1}^{m} f_i(\vec{x}) \leq b_i \end{aligned}$$

Many classes of optimization problems, based on restrictions of f's and b's

Exact Global Optimization

$$\begin{array}{ll} \underset{\vec{x}}{\text{minimize}} & f(\vec{x}) \\ \text{subject to} & \bigwedge_{i=1}^{m} f_i(\vec{x}) \leq b_i \end{array}$$

Many classes of optimization problems, based on restrictions of f's and b's

nonlinear, computable

What is a Real Closed Field?

$$\mathrm{RCF} = Th(\langle \mathbb{R}, +, *, <, 0, 1 \rangle)$$

What is a Real Closed Field?

$$\mathrm{RCF} = Th(\langle \mathbb{R}, +, *, <, 0, 1 \rangle)$$

Examples:

- The reals: $\langle \mathbb{R}, +, *, <, 0, 1 \rangle$
- The algebraic reals: $\langle \mathbb{R}_{alg}, +, *, <, 0, 1 \rangle$
- The (a!) Hyperreals: $\left(\prod_{\mathbb{N}} \langle \mathbb{R}, +, *, <, 0, 1 \rangle \right) / \, \mathcal{U}$
- Real closures: $\widetilde{\mathbb{K}}$ s.t. $\mathbb{K} = \mathbb{Q}(t_1, \ldots, t_n, \epsilon_1, \ldots, \epsilon_m)$

Optimization using RCF QE - I

RCF admits quantifier elimination (QE)

Optimization using RCF QE - I

RCF admits quantifier elimination (QE)

In theory, one can exploit RCF QE to solve nonlinear optimization problems over the reals:
Let's see how! In the next slide...

Optimization using RCF QE - I

RCF admits quantifier elimination (QE)

In theory, one can exploit RCF QE to solve nonlinear optimization problems over the reals:
Let's see how! In the next slide...

In practice, this is not a viable solution:
RCF QE is infeasible: O(2^2^(Omega(n)))

Optimization using RCF QE - II

$$\begin{aligned} & \underset{\vec{x}}{\text{minimize}} && f(\vec{x}) \\ & \text{subject to} && \bigwedge_{i=1}^{m} f_i(\vec{x}) \leq b_i \end{aligned}$$

Optimization using RCF QE - II

$$\begin{array}{ll} \underset{\vec{x}}{\text{minimize}} & f(\vec{x}) \\ \text{subject to} & \bigwedge_{i=1}^{m} f_i(\vec{x}) \leq b_i \end{array}$$

Step 1: New coordinate function y

$$F(\vec{x}, y) \triangleq \left(y = f(\vec{x}) \wedge \bigwedge_{i=1}^{m} f_i(\vec{x}) \leq b_i \right)$$

Optimization using RCF QE - II

$$\begin{array}{ll} \underset{\vec{x}}{\text{minimize}} & f(\vec{x}) \\ \text{subject to} & \bigwedge_{i=1}^{m} f_i(\vec{x}) \le b_i \end{array}$$

Step 1: New coordinate function y

$$F(\vec{x}, y) \triangleq \left(y = f(\vec{x}) \wedge \bigwedge_{i=1}^{m} f_i(\vec{x}) \le b_i \right)$$

Step 2: QE (project onto y)

Use RCF QE to eliminate $\exists \vec{x}$ from $\exists \vec{x} F(\vec{x}, y)$, obtaining $\varphi(y)$ s.t.

$$\varphi(y) \triangleq \bigvee_{i} \bigwedge_{j} (p_{i,j}(y) \bowtie_{i,j} 0), \qquad \bowtie_{i,j} \in \{<, \le, =, \ge, >\}, \quad p_{i,j} \in \mathbb{Z}[y].$$

Step 2: QE (project onto y)

Use RCF QE to eliminate $\exists\vec{x}$ from $\exists\vec{x}F(\vec{x},y)$, obtaining $\varphi(y)$ s.t.

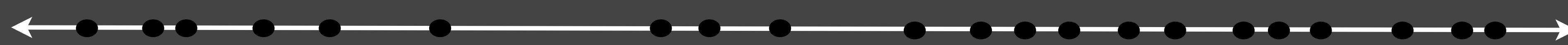
$$\varphi(y) \triangleq \bigvee_i \bigwedge_j (p_{i,j}(y) \bowtie_{i,j} 0), \qquad \bowtie_{i,j} \in \{<, \leq, =, \geq, >\}, \quad p_{i,j} \in \mathbb{Z}[y].$$

Step 3: Real Root Isolation
(note sign invariance: IVT!)

Use univariate real root isolation (e.g., via Sturm sequences) to isolate all roots of $p_{i,j}(y) \in \mathbb{Z}[y]$. This partitions $\mathbb{R}$ into $2k+1$ connected components.

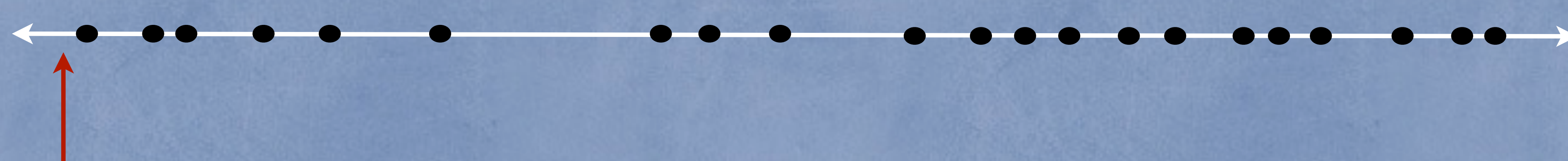
Step 3: Real Root Isolation
(note sign invariance: IVT!)

Use univariate real root isolation (e.g., via Sturm sequences) to isolate all roots of $p_{i,j}(y) \in \mathbb{Z}[y]$. This partitions $\mathbb{R}$ into $2k+1$ connected components.



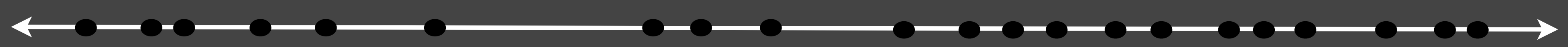
Step 4: Search!

Sweep from L to R, looking for first connected component satisfying $\varphi(y)$



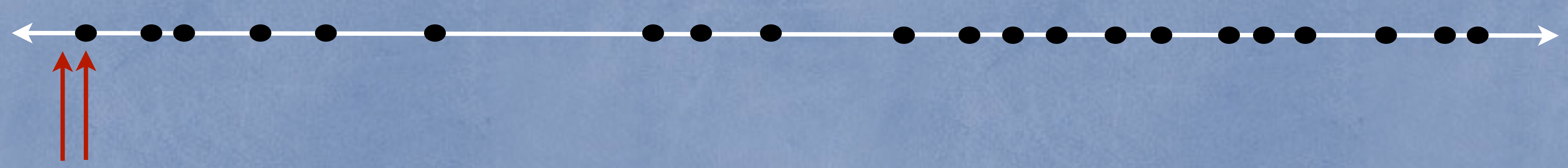
Step 3: Real Root Isolation
(note sign invariance: IVT!)

Use univariate real root isolation (e.g., via Sturm sequences) to isolate all roots of $p_{i,j}(y) \in \mathbb{Z}[y]$. This partitions $\mathbb{R}$ into $2k+1$ connected components.



Step 4: Search!

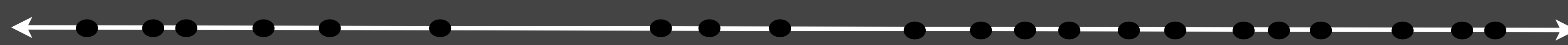
Sweep from L to R, looking for first connected component satisfying $\varphi(y)$



Monday, October 21, 13

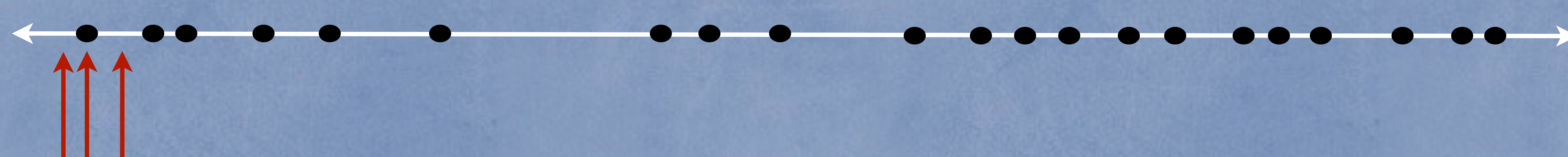
Step 3: Real Root Isolation
(note sign invariance: IVT!)

Use univariate real root isolation (e.g., via Sturm sequences) to isolate all roots of $p_{i,j}(y) \in \mathbb{Z}[y]$. This partitions $\mathbb{R}$ into $2k+1$ connected components.



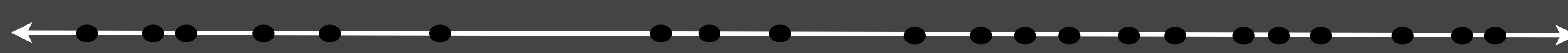
Step 4: Search!

Sweep from L to R, looking for first connected component satisfying $\varphi(y)$



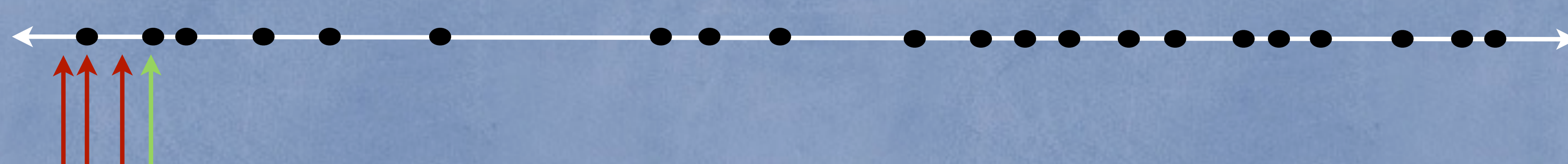
Step 3: Real Root Isolation
(note sign invariance: IVT!)

Use univariate real root isolation (e.g., via Sturm sequences) to isolate all roots of $p_{i,j}(y) \in \mathbb{Z}[y]$. This partitions $\mathbb{R}$ into $2k+1$ connected components.



Step 4: Search!

Sweep from L to R, looking for first connected component satisfying $\varphi(y)$



exact minimum found!

Four Possible Outcomes

- No satisfying region: Infeasible
- (-inf, r) : Unbounded
- [r] : Exact minimum
- (r, _) : No minimum, but exact infimum

Five ~~Four~~ Possible Outcomes

- No satisfying region: Infeasible
- (-inf, r) : Unbounded
- [r] : Exact minimum
- (r, _) : No minimum, but exact infimum
- RUN OUT OF MEMORY AND/OR TIME!!!

Computing $\varphi(y)$ explicitly is a bad idea!

A CAD-based Approach

- Used by Mathematica
- Doesn't require explicit computation of Phi(y)
- But, it is *eager* and *pessimistic*
- Our new approach is *lazy* and *optimistic*
- First, let's understand the CAD-based approach...

Cylindrical Algebraic Decomposition

CAD: A partitioning of $\mathbb{R}^n$
into finitely many RCF-definable connected components which "behaves nicely" w.r.t. projections onto lower dimensions.

$$P \subset \mathbb{Z}[x_1, \ldots, x_n]$$

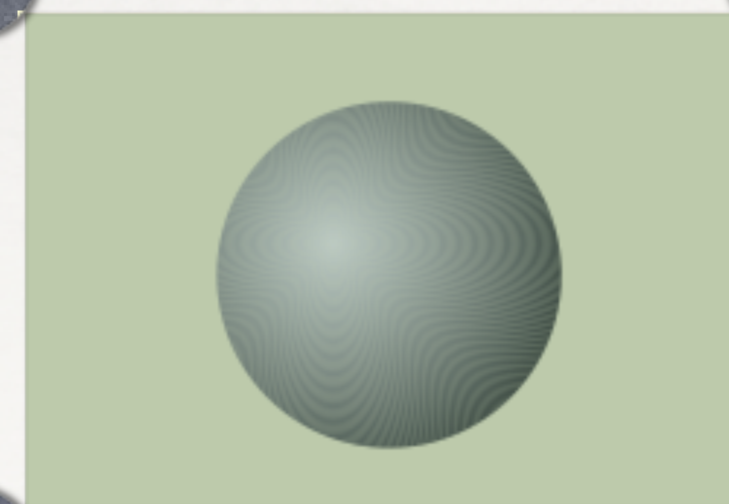
P-invariant **CAD**

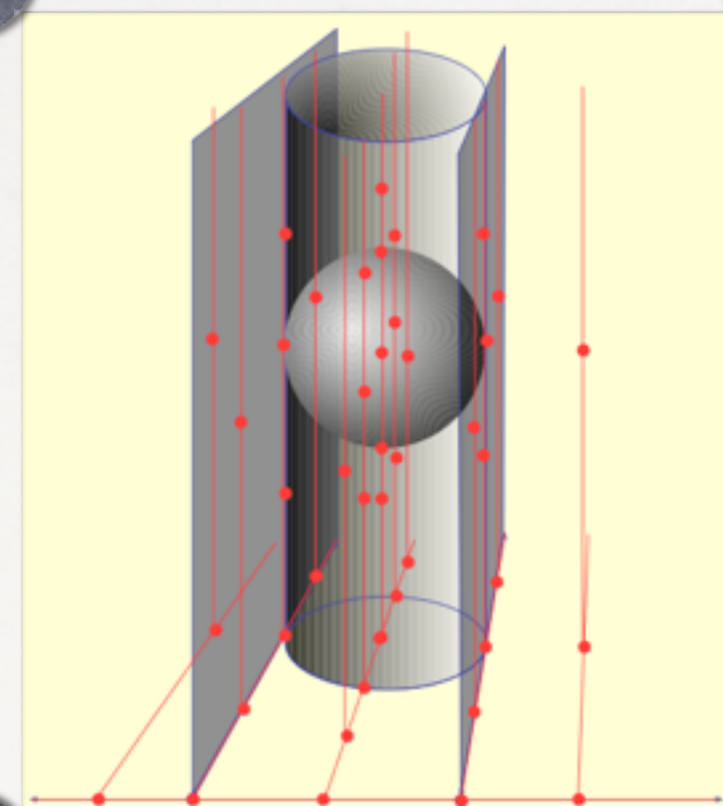
a CAD of $\mathbb{R}^n$ s.t. for all cells c_i, all $p \in P$

$$\forall \vec{r} \in c_i(p(\vec{r}) = 0) \quad \vee$$

$$\forall \vec{r} \in c_i(p(\vec{r}) > 0) \quad \vee$$

$$\forall \vec{r} \in c_i(p(\vec{r}) < 0) \quad .$$





CAD sphere diagrams: C. Brown and QEPCAD-B

CAD Phase I: Projection

$$Proj_{i+1} : \mathbb{Z}[x_1, \ldots, x_{i+1}] \to \mathbb{Z}[x_1, \ldots, x_i]$$

Inductive Property:

A (P_{i+1})-invariant CAD for R^{i+1}
can be constructed from
a (P_i)-invariant CAD of R^i.

$$P_n = P \subset \mathbb{Z}[x_1, \ldots, x_n]$$

$$P_{n-1} = Proj(P_n) \subset \mathbb{Z}[x_1, \ldots, x_{n-1}]$$

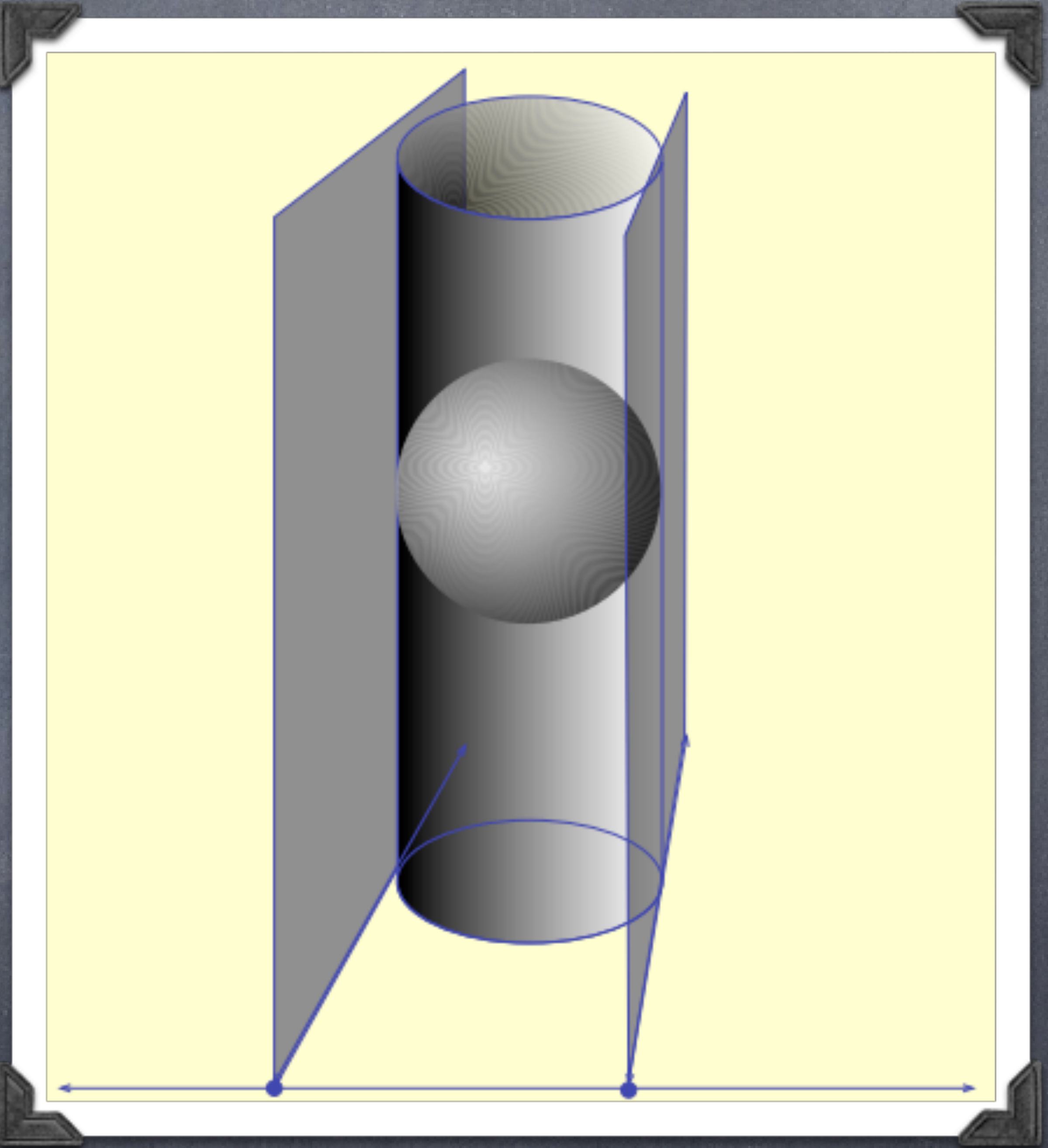
$$\vdots$$

$$P_2 = Proj(P_3) \subset \mathbb{Z}[x_1, x_2]$$

$$P_1 = Proj(P_2) \subset \mathbb{Z}[x_1]$$

Projection sets

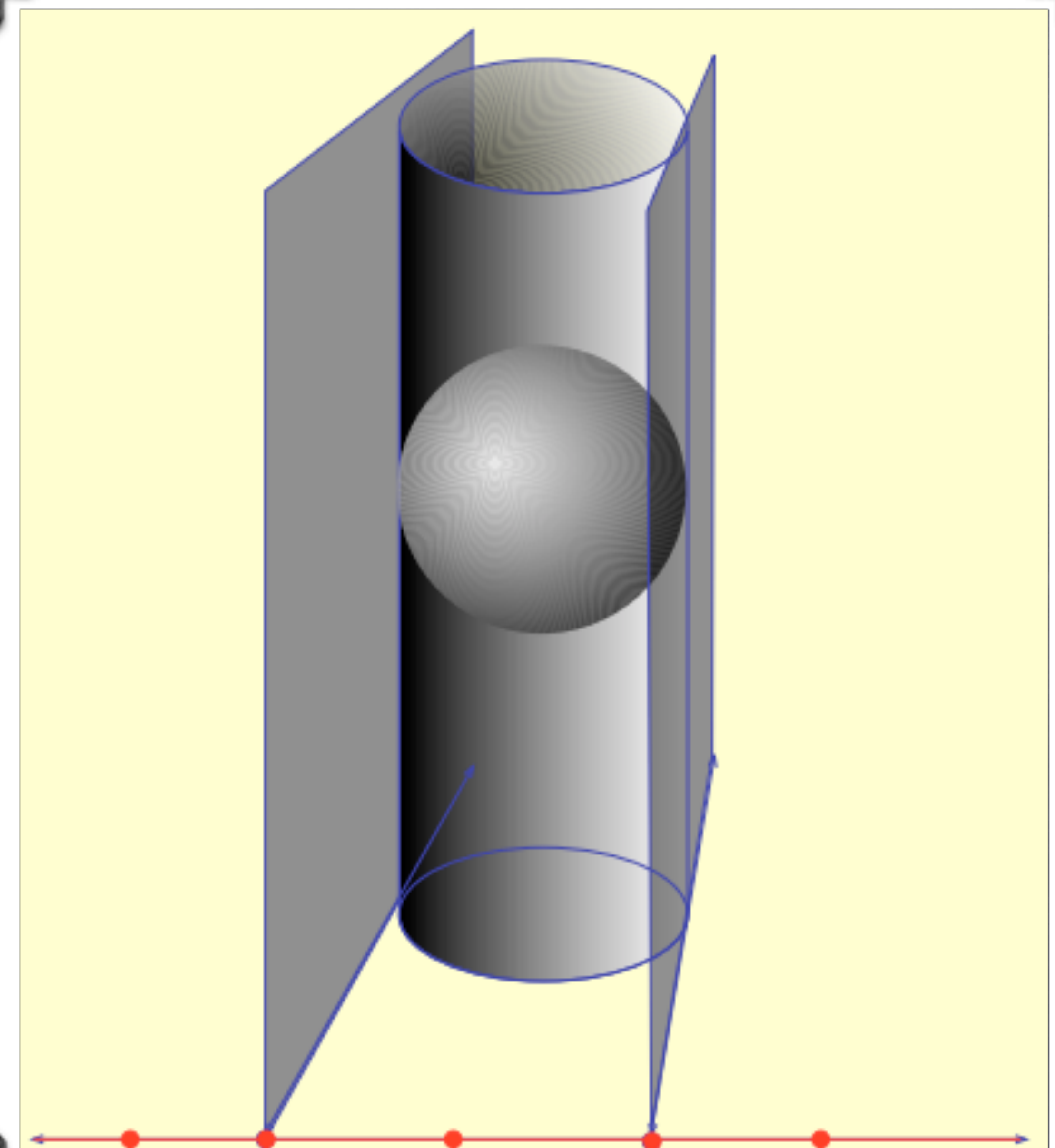
$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$



CAD sphere diagrams: C. Brown and QEPCAD-B

Projection sets

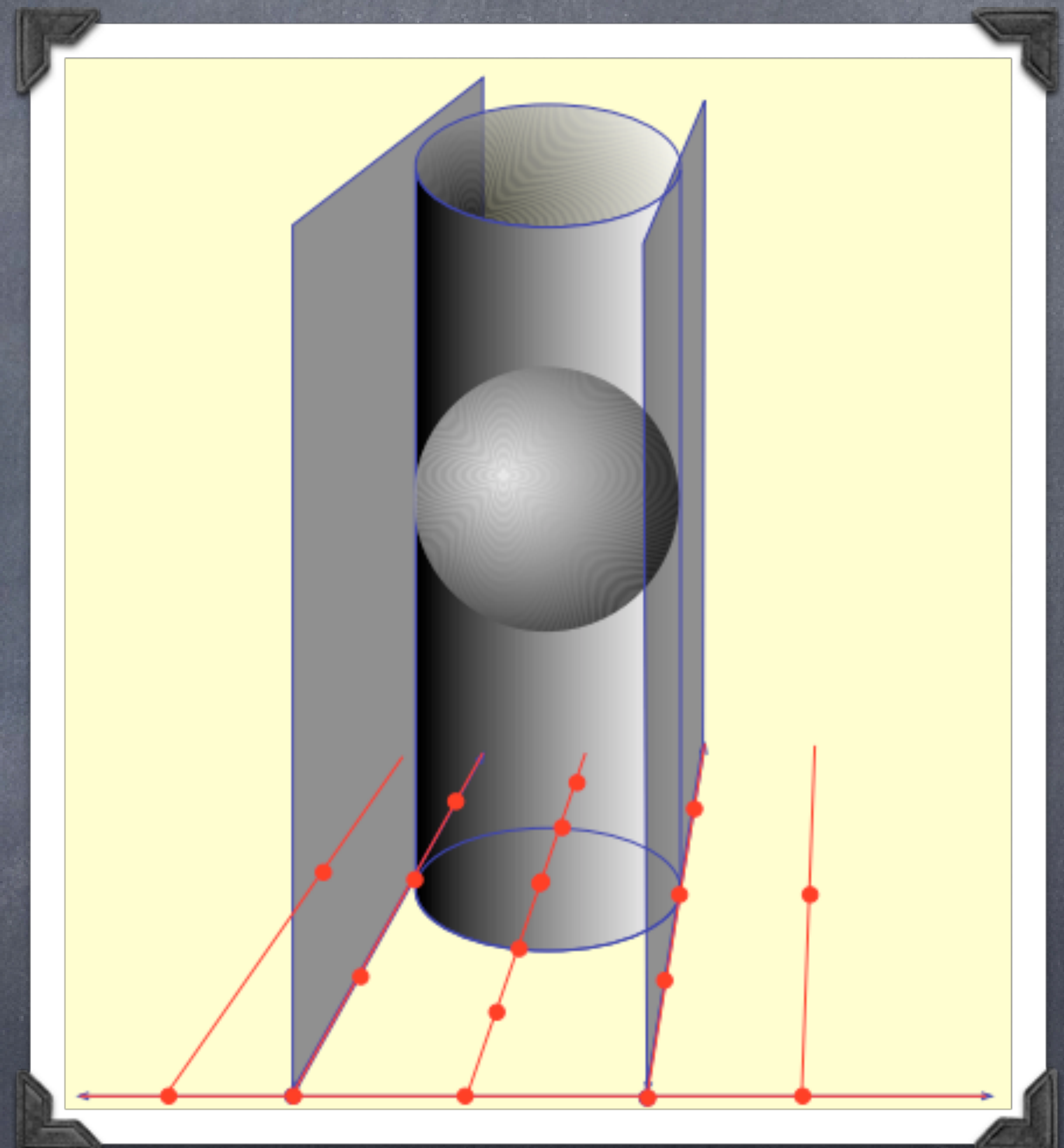
$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$



Base Phase: R^1

Projection sets

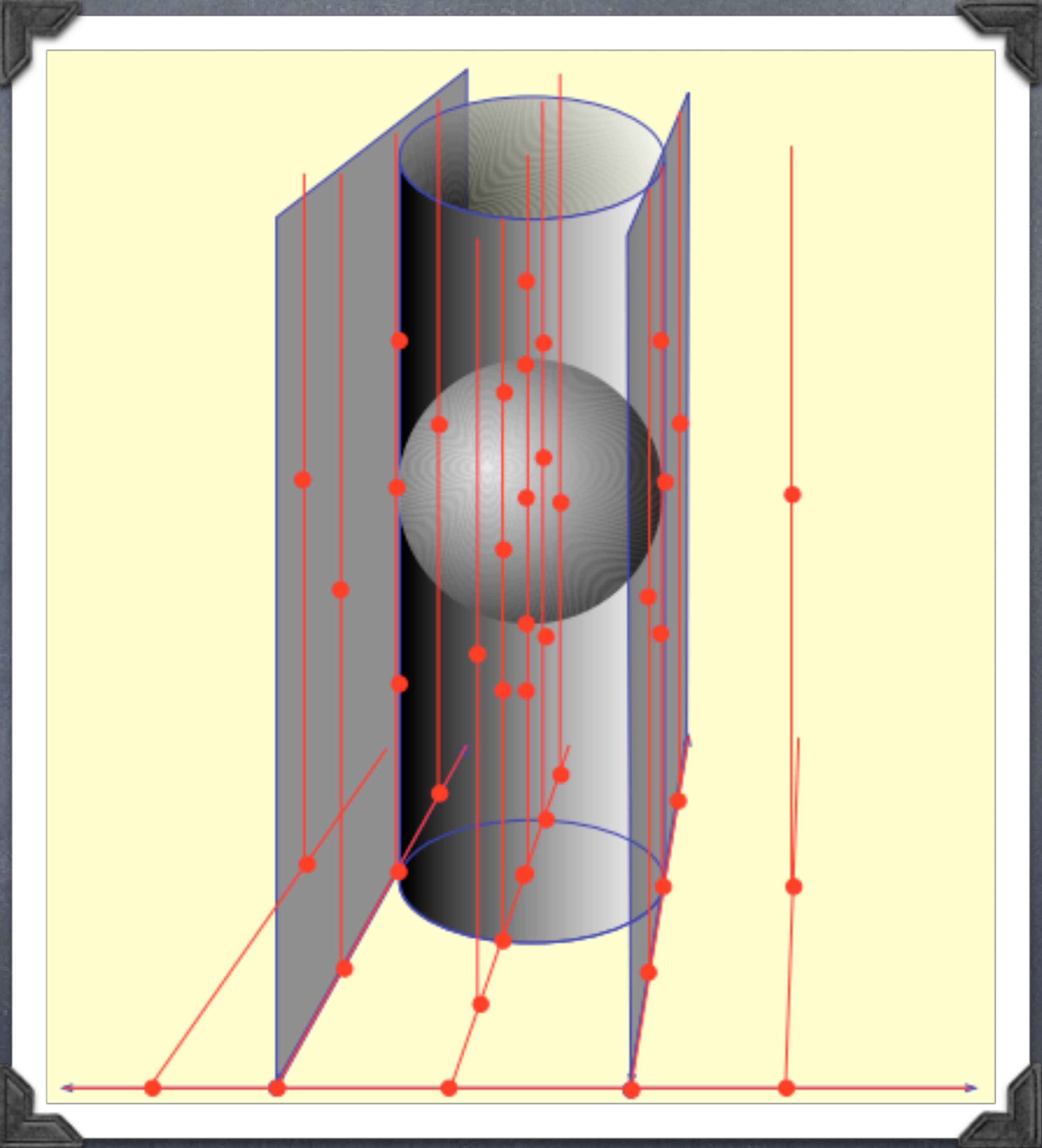
$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$



Lifting Phase: R^1 -> R^2

Projection sets

$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$



Lifting Phase: R^2 -> R^3

A CAD-based Approach to Optimization

$$\begin{array}{ll} \underset{\vec{x}}{\text{minimize}} & f(\vec{x}) \\ \text{subject to} & \bigwedge_{i=1}^{m} f_i(\vec{x}) \le b_i \end{array}$$

Step 1: New coordinate function y

$$F(\vec{x}, y) \triangleq \left(y = f(\vec{x}) \wedge \bigwedge_{i=1}^{m} f_i(\vec{x}) \le b_i \right)$$

Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$

$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$

$$\vdots$$

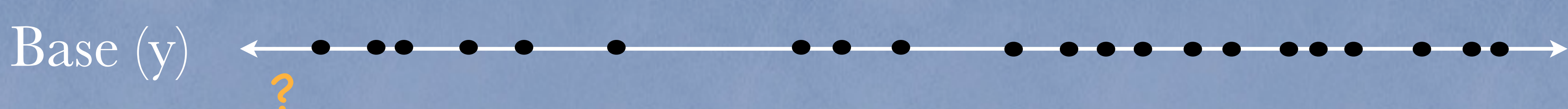
$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$

$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$
$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$
$$\vdots$$
$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$
$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

Step 3: CAD Base and Lifting (depth-first) from L to R



Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$

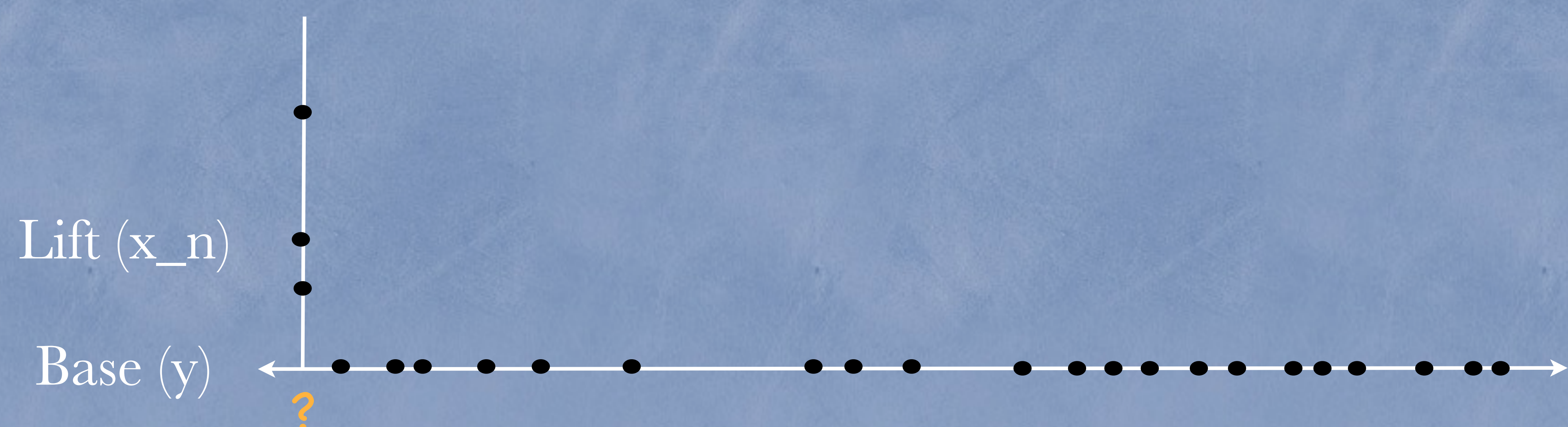
$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$

$$\vdots$$

$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$

$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

Step 3: CAD Base and Lifting (depth-first) from L to R

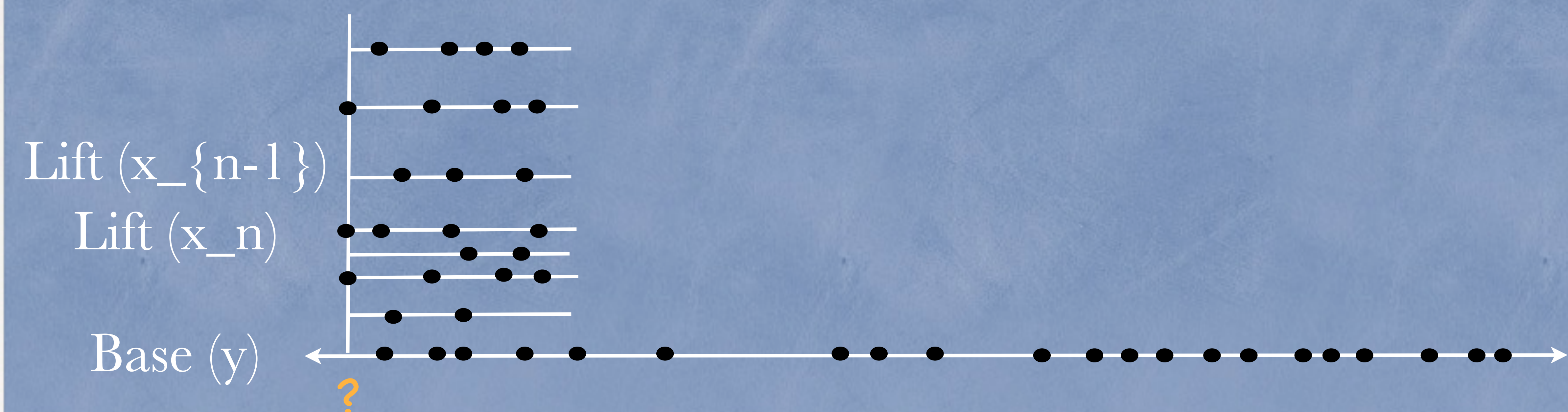


Monday, October 21, 13

Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$
$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$
$$\vdots$$
$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$
$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

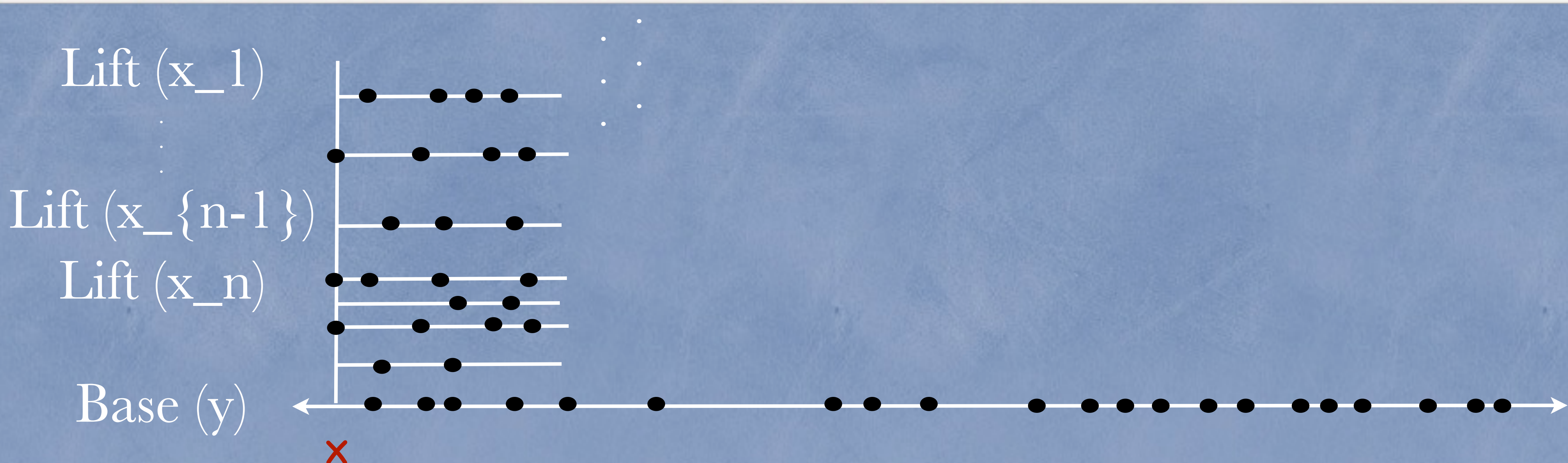
Step 3: CAD Base and Lifting (depth-first) from L to R



Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$
$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$
$$\vdots$$
$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$
$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

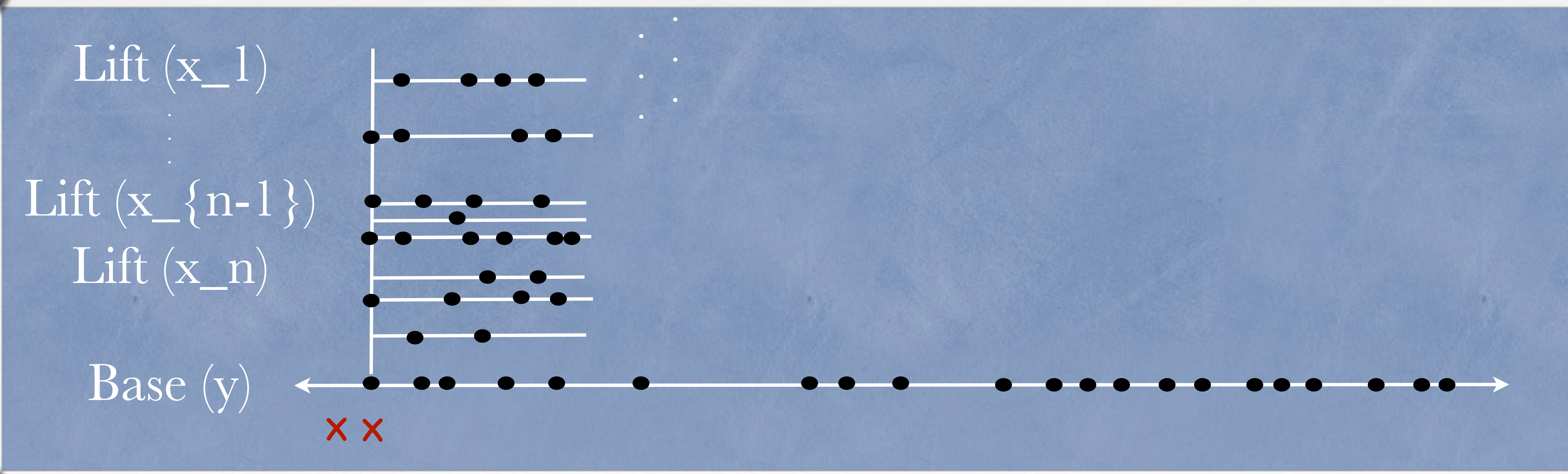
Step 3: CAD Base and Lifting (depth-first) from L to R



Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$
$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$
$$\vdots$$
$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$
$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

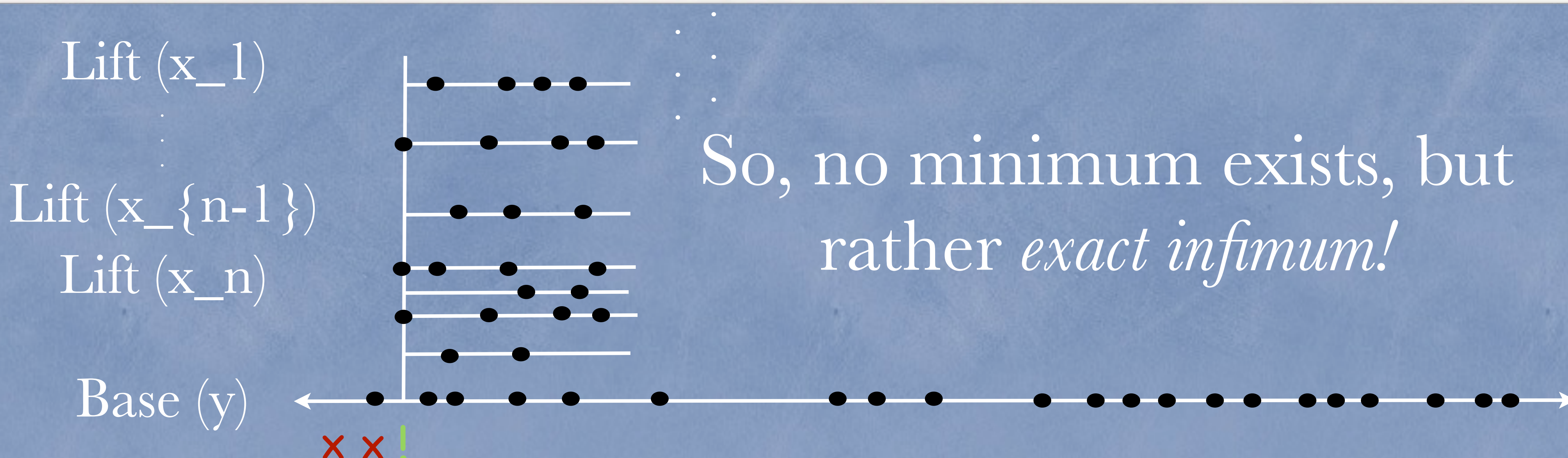
Step 3: CAD Base and Lifting (depth-first) from L to R



Step 2: CAD projection (with y lowest variable)

$$P_{n+1} = \{y - f(\vec{x}), f_1(\vec{x}) - b_1, \ldots, f_m(\vec{x}) - b_m\} \subset \mathbb{Z}[y, x_1, \ldots, x_n]$$
$$P_n = Proj(P_{n+1}) \subset \mathbb{Z}[y, x_1, \ldots, x_{n-1}]$$
$$\vdots$$
$$P_2 = Proj(P_3) \subset \mathbb{Z}[y, x_1]$$
$$P_1 = Proj(P_2) \subset \mathbb{Z}[y]$$

Step 3: CAD Base and Lifting (depth-first) from L to R



So, no minimum exists, but rather *exact infimum!*

Recap of CAD-based Approach

- Used by Mathematica
- Doesn't require explicit computation of Phi(y)
- But, it is *eager* and *pessimistic:*
  FULL CAD Projection (expensive!!!)
- Our new approach is *lazy* and *optimistic*
- We build on *nlsat/mcsat*, a CDCL-like approach to the Existential fragment of RCF

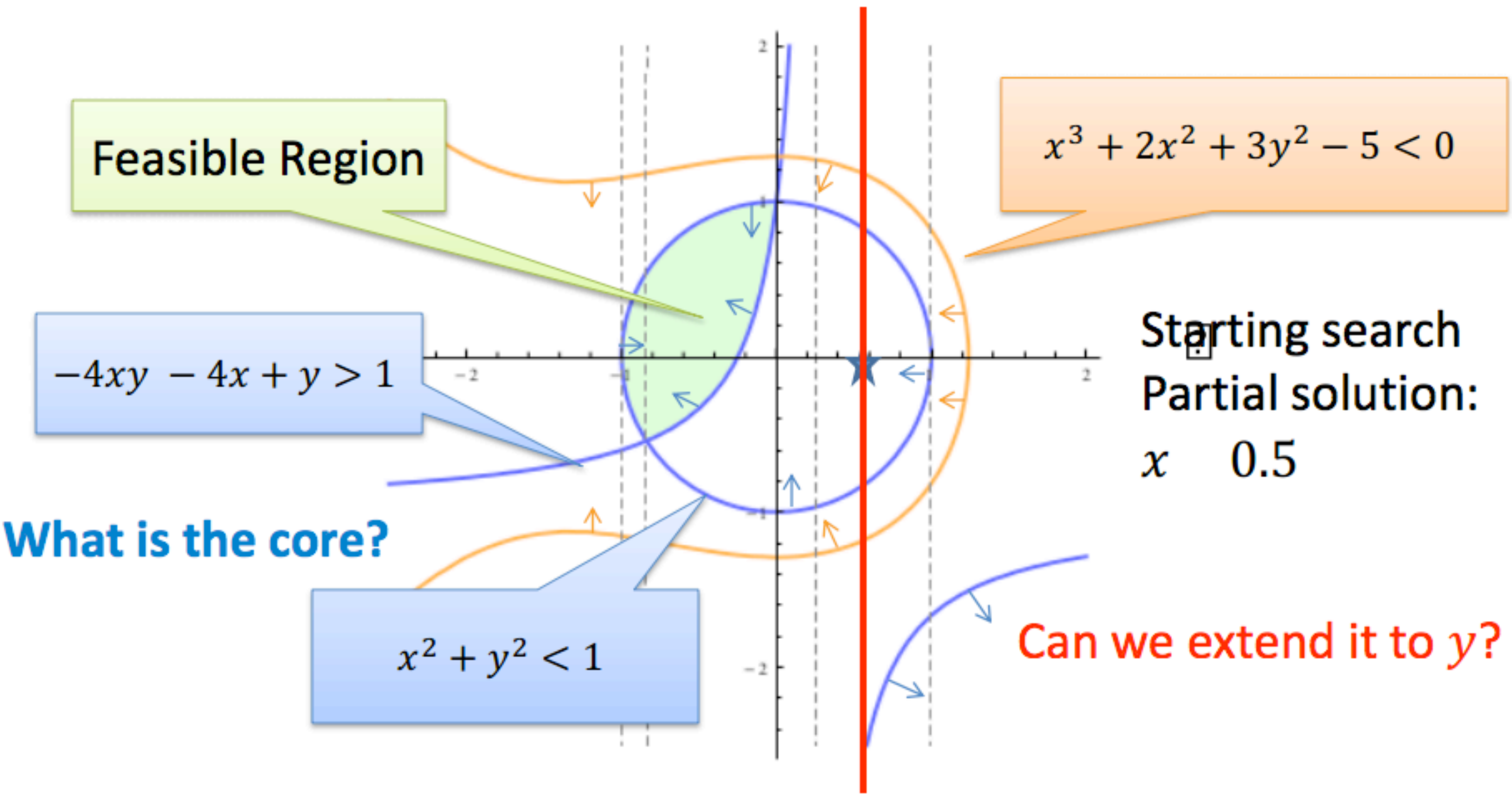
nlsat/mcsat: CDCL-like approach to ExRCF

- Start building model for formula immediately, without first going through projection phase
- When conflict arises, use *projection on demand*
- Real-algebraic analogue of *conflict clauses* **generalize** a non-extendable partial models to rule out a *delineable* region containing them
- Non-chronological backtracking

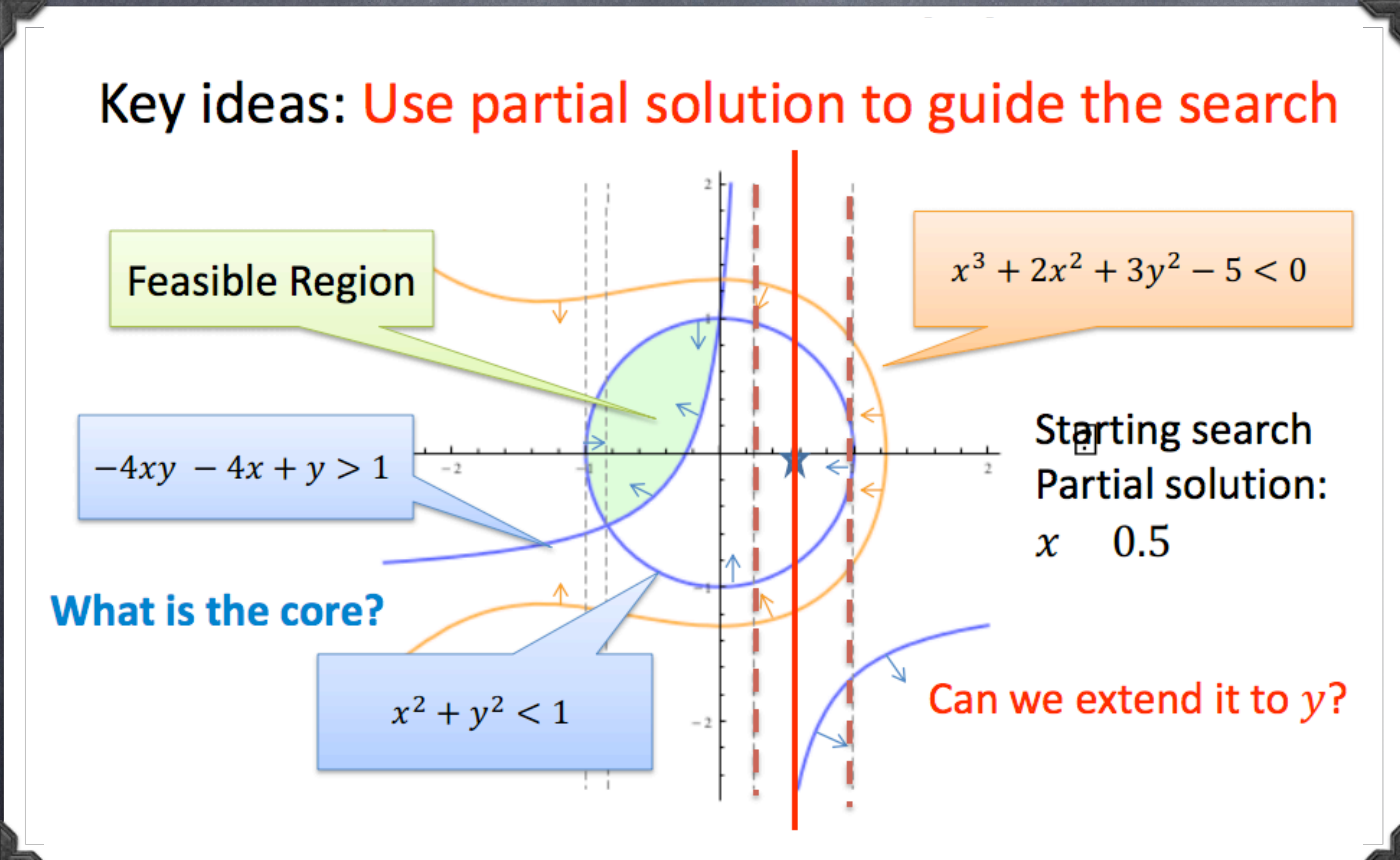
* Typically much more efficient on ExRCF SMT problems than classical (eager & pessimistic) CAD!

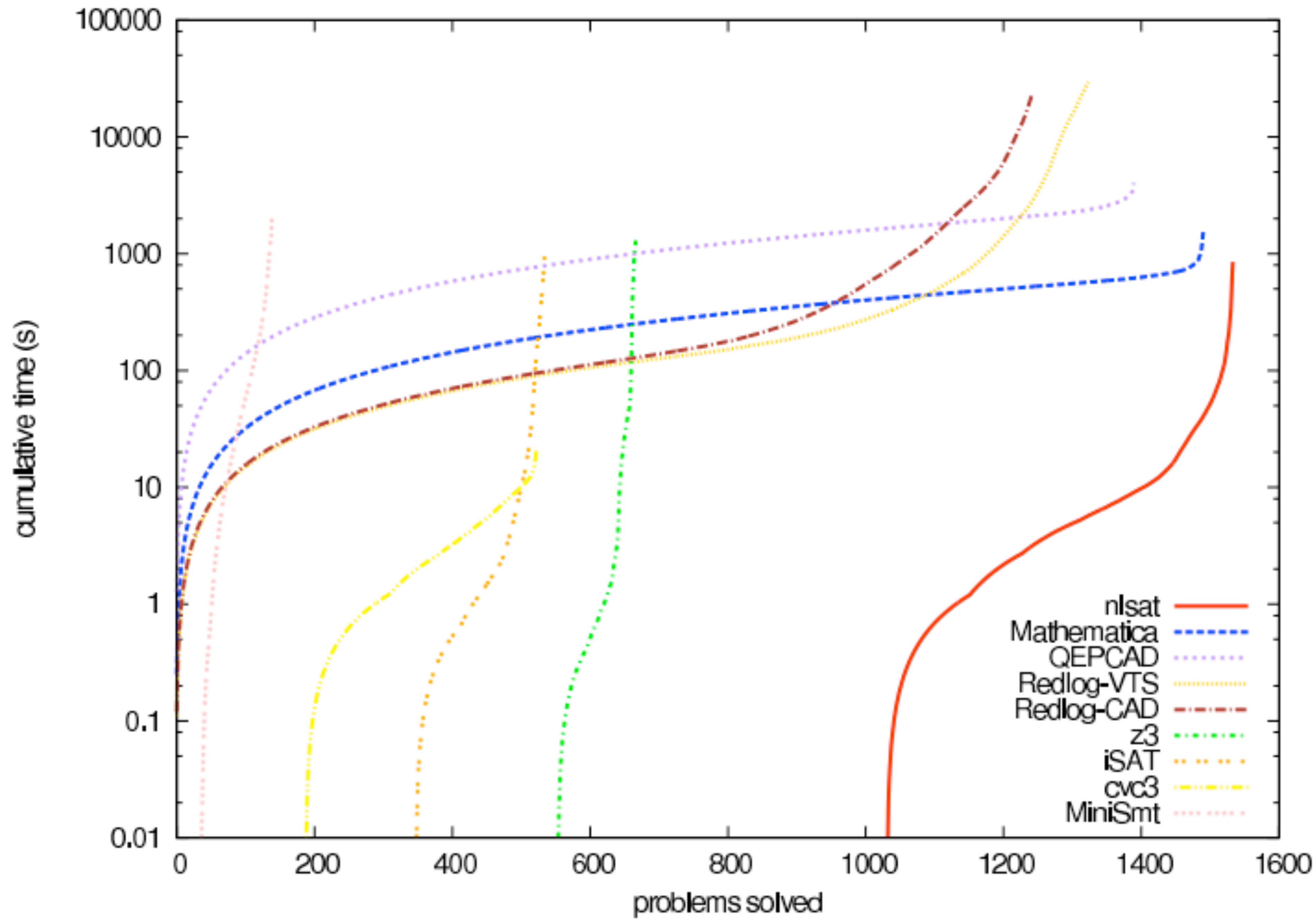
nlsat: CDCL-like approach to ExRCF

Key ideas: Use partial solution to guide the search



nlsat: CDCL-like approach to ExRCF




cumulative time (s)
problems solved
100000
10000
1000
100
10
1
0.1
0.01
0
200
400
600
800
1000
1200
1400
1600
nlsat
Mathematica
QEPCAD
Redlog-VTS
Redlog-CAD
z3
iSAT
cvc3
MiniSmt

CAD-based optimization + nlsat = ?

CAD-based Optimization uses projection eagerly and pessimistically

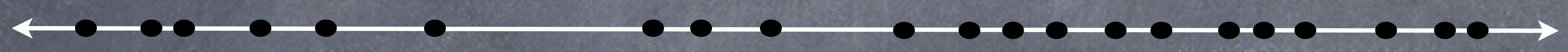
nlsat/mcsat solves Exists RCF by using projection lazily and optimistically

...but how can we combine the two?

Key difficulty: Sweeping L to R

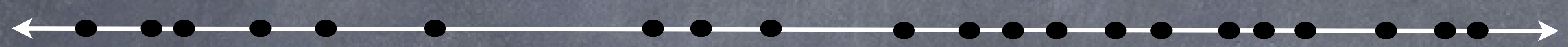
- We use coordinate function y to represent the objective function
- Then, we need to *sweep* along all possible values of y from **Left to Right**
- After CAD projection, we can do this
- But, what about with nlsat/mcsat?

Key difficulty: Sweeping L to R

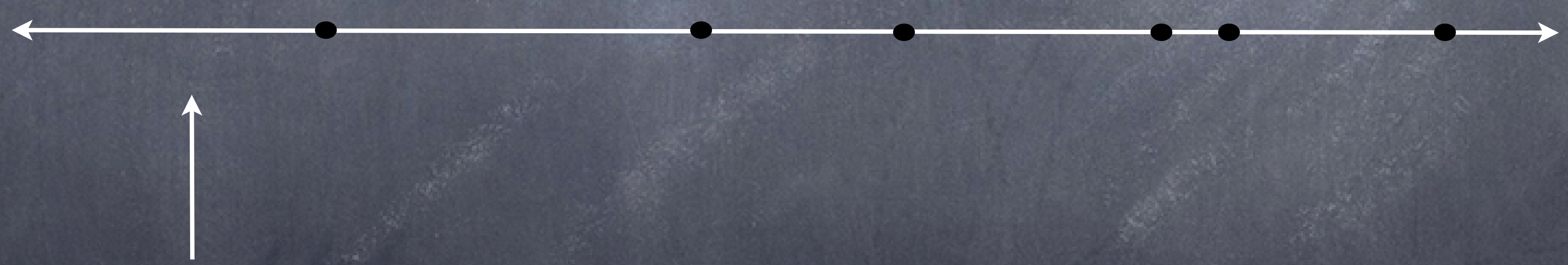


vs

Key difficulty: Sweeping L to R



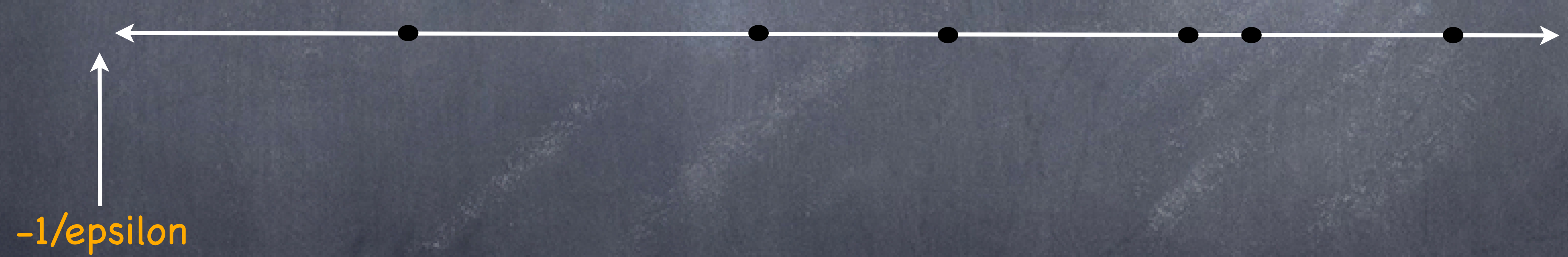
vs



where to start? how to move to `next' region?

Key difficulty: Sweeping L to R

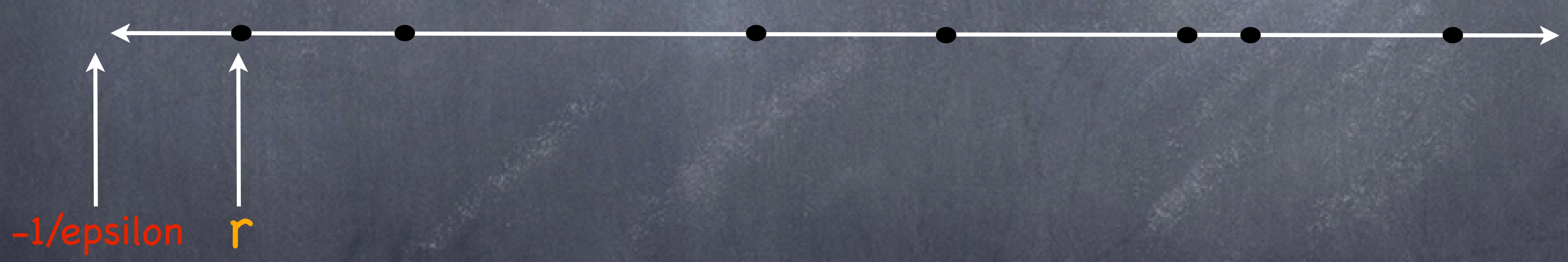
Key idea:
RCFs containing infinitesimals!



where to start? how to move to `next' region?

Key difficulty: Sweeping L to R

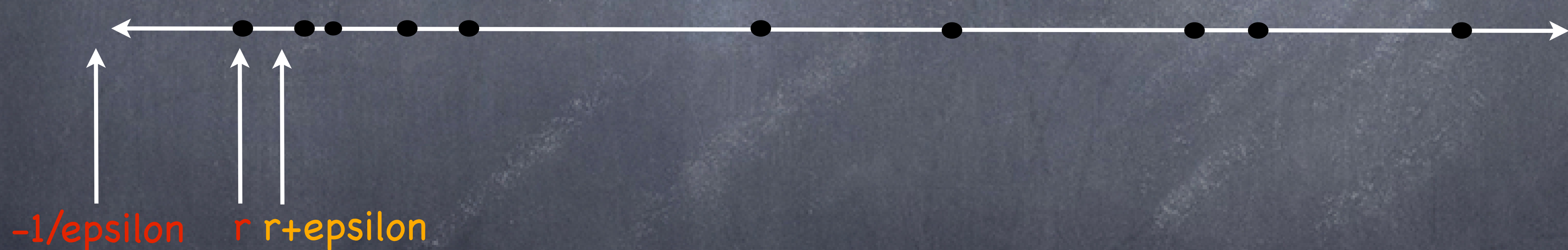
Key idea:
RCFs containing infinitesimals!



where to start? how to move to `next' region?

Key difficulty: Sweeping L to R

Key idea:
RCFs containing infinitesimals!



where to start? how to move to `next' region?

Computing with Infinitesimals & Computable Transcendentals

Technique explained in our 2013 CADE paper

Computation in Real Closed Infinitesimal and Transcendental Extensions of the Rationals

$$\sqrt{2} + \sqrt{3}$$

$$\sqrt[3]{\frac{1}{9}} - \sqrt[3]{\frac{2}{9}} + \sqrt[3]{\frac{4}{9}} = \sqrt[3]{\sqrt[3]{2} - 1}$$

Infinitesimal

$$\frac{1+\epsilon}{\epsilon^2} > 10^{100}$$

Transcendental

$$\pi + \epsilon < \pi$$

FindRoots $(1 - \sqrt{2}\, x^2 - \epsilon x^3 + \epsilon^2\, x^5)$

Real Closed Fields

$$\mathbb{R}$$

$$\rotatebox{90}{$\subseteq$}$$

$$\mathbb{R}_{alg} = \widetilde{\mathbb{Q}}$$

Real Closure of the Rational Numbers

Real Closed Fields

$\ldots, \sqrt{2}, \sqrt[3]{\pi},$
$root(-\pi - x + x^5, (1,2)), \ldots$

$\mathbb{R}$

$\cup\!\!|$

$\widetilde{K}, K = \mathbb{Q}(\pi)$

Field extension

$1, {}^{1}\!/_{3}, \pi, \pi + 1, \dfrac{\pi^2 + 1}{2}, \ldots$

$\cup\!\!|$

$\mathbb{R}_{alg} = \widetilde{\mathbb{Q}}$

Real Closed Fields

$$\mathbb{R}$$

$$\cup\!\!|$$

$$\widetilde{K_1}, K_1 = \mathbb{Q}(\pi)(e)$$

$$\cup\!\!|$$

$$\widetilde{K}, K = \mathbb{Q}(\pi)$$

$$\cup\!\!|$$

$$\mathbb{R}_{alg} = \widetilde{\mathbb{Q}}$$

$1, e, \pi + e,$
$\frac{e^2 + \pi}{2}, \ldots$

Real Closed Fields

Hyperreals

$$\begin{array}{ccc} & \mathbb{H} & \\ \rotatebox{-45}{$\subseteq$} & & \rotatebox{45}{$\subseteq$} \\ \mathbb{R} & & \widetilde{K_2}, K_2 = \mathbb{Q}(\pi)(e)(\epsilon) \\ \cup\!\shortmid & & \cup\!\shortmid \end{array}$$

Infinitesimal

$$\widetilde{K_1}, K_1 = \mathbb{Q}(\pi)(e)$$

$$\cup\!\shortmid$$

$$\widetilde{K}, K = \mathbb{Q}(\pi)$$

$$\cup\!\shortmid$$

$$\mathbb{R}_{alg} = \widetilde{\mathbb{Q}}$$

Real Closed Fields

Hyperreals

$$\mathbb{H}$$

$$\rotatebox{45}{$\subseteq$} \qquad \rotatebox{-45}{$\supseteq$}$$

$$\mathbb{R} \qquad\qquad \widetilde{K_2}, K_2 = \mathbb{Q}(\pi)(e)(\epsilon)$$

$$\cup\!| \qquad\qquad\qquad \cup\!|$$

Infinitesimal

$$\widetilde{K_1}, K_1 = \mathbb{Q}(\pi)(e)$$

$$\cup\!|$$

$$\widetilde{K}, K = \mathbb{Q}(\pi)$$

$$\cup\!|$$

$$\mathbb{R}_{alg} = \widetilde{\mathbb{Q}}$$

Our approach

Tower of extensions

Hybrid representation

Interval (arithmetic) + Thom’s lemma

Clean denominators

Non-minimal defining polynomials

Tower of extensions

Basic Idea:

Given (computable) ordered field K

Implement $K(\varsigma)$

Tower of extensions

(Computable) ordered field K

Operations: $+,\ -,\ \times,\ inv,\ sign$

$$a < b \Leftrightarrow sign(a - b) = -1$$

Binary Rational
$$\frac{a}{2^k}$$

Approximation: $approx(a) \in B_\infty$-interval

$$B_\infty = B \cup \{-\infty, \infty\}$$

$$a \neq 0 \Rightarrow 0 \notin approx(a)$$

Refine approximation

(Computable) Transcendental Extensions

$approx(\pi)(k) \in B_\infty$-interval

$$\forall n \in \mathbb{N}^+, \exists k \in \mathbb{N}, width(approx(\pi)(k)) < \frac{1}{n}$$

Elements of the extension are encoded as rational functions

$$\frac{\pi^2 + \pi - 2}{\pi + 1}$$

(Computable) Transcendental Extensions

$$\frac{1}{2}\pi + \frac{1}{\pi + 1} = \frac{\frac{1}{2}\pi^2 + \frac{1}{2}\pi + 1}{\pi + 1}$$

Standard normal form for rational functions
GCD(numerator, denominator) = 1
Denominator is a monic polynomial

(Computable) Transcendental Extensions

Refine interval

    Interval arithmetic

    Refine coefficients and extension

Zero iff numerator is the zero polynomial

If $q(x)$ is not the zero polynomial,

then $q(\pi)$ can't be zero, since π is transcendental.

Remark

$\sqrt{\pi}$ is transcendental with respect to $\mathbb{Q}$

$\sqrt{\pi}$ is not transcendental with respect to $\mathbb{Q}(\pi)$

Infinitesimal Extensions

Every infinitesimal extension is transcendental

Rational functions

$$sign(a_0 + a_1\epsilon + \ \dots + a_n\epsilon^n)$$

sign of first non zero coefficient

$$approx(\epsilon) = (0, \frac{1}{2^k})$$

Non-refinable intervals

$$approx\left(\frac{1}{\epsilon}\right) = (2^k, \infty)$$

Algebraic Extensions

$K(\alpha)$

α is a root of a polynomial with coefficients in K

Encoding α as polynomial + interval does not work

K may not be Archimedian

Roots can be infinitely close to each other.

Roots can be greater than any Real.

Thom's Lemma

We can always distinguish the roots of a polynomial in a RCF using the signs of the derivatives

Algebraic Extensions

Roots: $-\sqrt{1/\epsilon}$, $\sqrt{1/\epsilon}$, $\sqrt[3]{1/\epsilon}$

Three roots of $\epsilon^2 x^5 - \epsilon x^3 - \epsilon x^2 + 1 \in (\mathbb{Q}(\epsilon))[x]$

$$(\epsilon^2 x^5 - \epsilon x^3 - \epsilon x^2 + 1, (-\infty, 0), \{\})$$
$$(\epsilon^2 x^5 - \epsilon x^3 - \epsilon x^2 + 1, (0, \infty), \quad \{60\epsilon^2 x^2 - 6\epsilon > 0\})$$
$$(\epsilon^2 x^5 - \epsilon x^3 - \epsilon x^2 + 1, (0, \infty), \quad \{60\epsilon^2 x^2 - 6\epsilon < 0\})$$

Algebraic Extensions

Given $H = \{h_1, \ldots, h_n\}$, $signdet(H, p, a, b)$
Feasible sign assignments of H at roots of p in (a, b)
Based on Sturm-Tarski Theorem
Ben-Or et al algorithm.

$sign\big(q(\alpha)\big)$ where $\alpha = (p, (a, b), S)$
$R = signdet(poly(S), p, (a, b))$

if $S \cup \{q = 0\} \in R$ then $q(\alpha) = 0$,
if $S \cup \{q > 0\} \in R$ then $q(\alpha) > 0$,
if $S \cup \{q < 0\} \in R$ then $q(\alpha) < 0$.

The procedure MkInfinitesimal creates a new infinitesimal extension, while Pi and E return π and e respectively. In the following example, we extract the first (and only) root of the polynomial $x^3 + \epsilon x^2 + (\sqrt{2} + \pi)\,x - \pi$.

```
eps = MkInfinitesimal("eps")
pi  = Pi()
r   = MkRoots([-pi, sqrt2 + pi, eps, 1])[0]
print(r)
>> root(x^3 + eps*x^2 + (root(x^2 + -2, (0, +oo), {}) + pi)*x +
       -1*pi, (0, +oo), {})
print(r.decimal(10))
>> 0.6337173142?
```

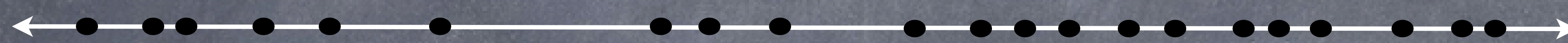
Compute with infinitesimals and transcendentals online!

1/epsilon amounts of fun! :-)

Key difficulty: Sweeping L to R

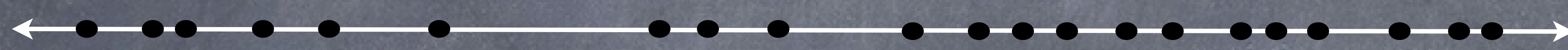
- We use coordinate function y to represent the objective function
- Then, we need to *sweep* along all possible values of y from **Left to Right**
- After CAD projection, we can do this
- But, what about with nlsat?

Key difficulty: Sweeping L to R

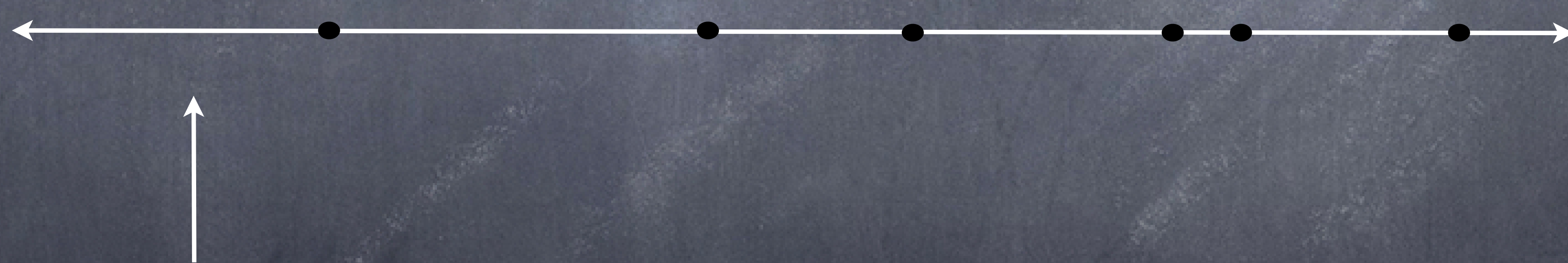


vs

Key difficulty: Sweeping L to R



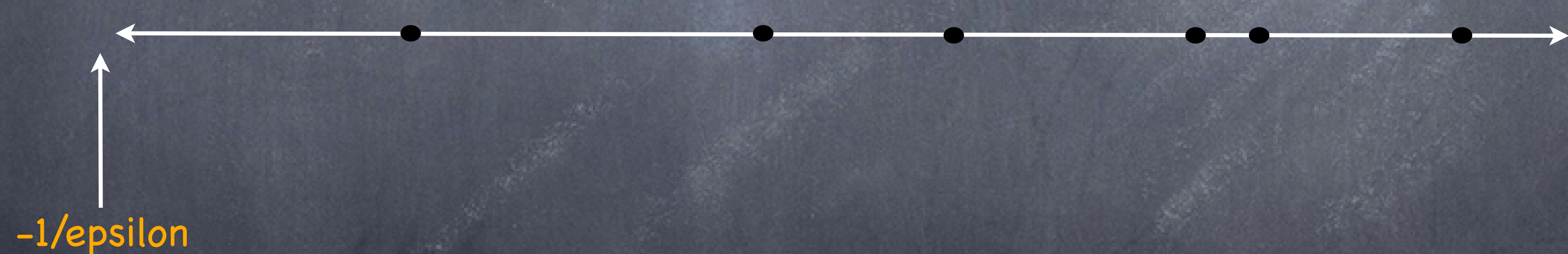
vs



where to start? how to move to `next' region?

Key difficulty: Sweeping L to R

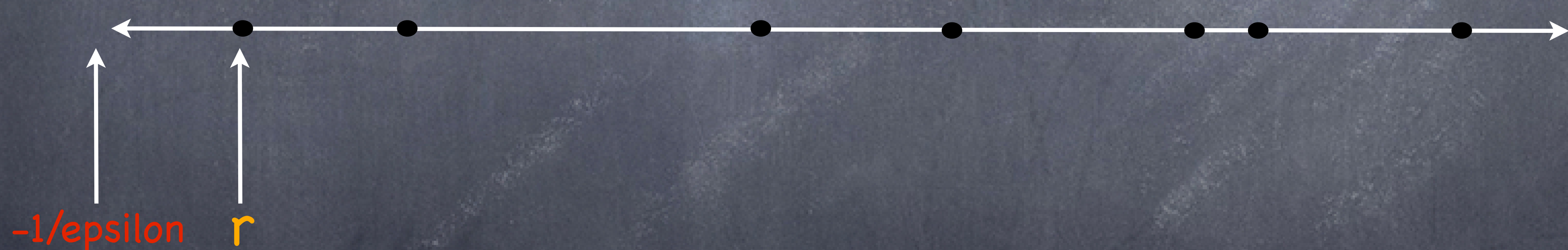
Key idea:
RCFs containing infinitesimals!



where to start? how to move to `next' region?

Key difficulty: Sweeping L to R

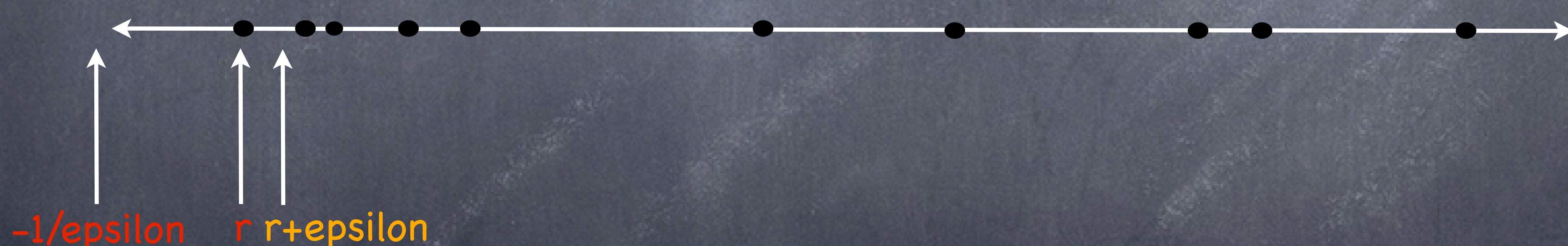
Key idea:
RCFs containing infinitesimals!



where to start? how to move to `next' region?

Key difficulty: Sweeping L to R

Key idea:
RCFs containing infinitesimals!



where to start? how to move to `next' region?

```
procedure Min(F(x⃗, y))
  G := true
  ε := MkInfinitesimal() (* create an infinitesimal value *)
  loop
    r := Min_0(G)
    case r of
      unsat ⇒ return unsat
      unbounded ⇒ v := −1/ε
      (inf, a) ⇒ v := a + ε
      (min, a) ⇒ v := a
    end
    case Check(F(x⃗, y), {y ↦ v}) of
      sat ⇒ return r
      (unsat, S) ⇒ G := G ∧ S
    end
  end
```

Min_0: Procedure for Univariate Optimization Problem

Check: Procedure for SAT Modulo Assignment Problem, with support for RCFs containing *infinitesimals*, and satisfying the *finite decomposition* property.

The RCF Optimization Problem

Input: A quantifier-free RCF formula $F(\vec{x}, y)$.

Output (with 'is (un)sat' meaning 'is (un)satisfiable over $\mathbb{R}$'):

$$\begin{cases} \mathsf{unsat}, & \text{if } F(\vec{x}, y) \text{ is unsat,} \\ \mathsf{unbounded}, & \text{if for all } v \text{ exists } w < v \text{ s.t. } F(\vec{x}, w) \text{ is sat,} \\ (\mathsf{inf}, a), & \text{if for all } v \leq a,\ F(\vec{x}, a) \text{ is unsat, and} \\ & \text{for all } \epsilon > 0 \text{ exists } v \in (a, a+\epsilon) \text{ s.t. } F(\vec{x}, v) \text{ is sat,} \\ (\mathsf{min}, a), & \text{if } F(\vec{x}, a) \text{ is sat, and for all } v < a,\ F(\vec{x}, v) \text{ is unsat.} \end{cases}$$

Conclusion

- A CDCL-like approach to exact nonlinear global optimization over the real numbers (and all RCFs)
- Three main conceptual ingredients:

CAD-based approach to optimization

eager method for nonlinear optimization, in Mathematica v9.x

nlsat/mcsat - existential CAD `on demand'

lazy CDCL-like approach to Exists RCF, in Z3 (Jovanović - de Moura, 2012)

computable nonstandard RCFs

computable RCFs containing infinitesimals (de Moura - Passmore, 2013)

Thank you!