Quantifier Elimination for Presburger Arithmetic

Yu-Fang Chen

based on the slides of Isil Dillig

Quantifier Elimination

- A theory T admits quantifier elimination if for every quantified formula, there exists an equivalent quantifier-free formula.
- A quantifier elimination procedure is an algorithm that computes an equivalent, quantifier-free formula for any quantified formula
- Quantifier elimination algorithm for a theory T allows deciding satisfiability of any quantified T-formula. Why?

Quantifier Elimination

- A theory T admits quantifier elimination if for every quantified formula, there exists an equivalent quantifier-free formula.
- A quantifier elimination procedure is an algorithm that computes an equivalent, quantifier-free formula for any quantified formula
- Quantifier elimination algorithm for a theory T allows deciding satisfiability of any quantified T-formula. Why?
- Because we can use quantifier elimination algorithm to obtain equivalent quantifier-free formula and use decision procedure for quantifier-free fragment

A Simplification

- For developing a quantifier elimination (QE) algorithm, sufficient to consider formulas of the form $\exists x.F$ where F is quantifier free

A Simplification

- For developing a quantifier elimination (QE) algorithm, sufficient to consider formulas of the form $\exists x.F$ where F is quantifier free
- Why is this the case?
- Given arbitrary formula G, first look at innermost quantified formula
- This innermost formula is either of the form $\forall x.F$ or $\exists x.F$
- If it is of the form $\exists x.F$, apply QE algorithm

A Simplification,cont

- If innermost quantified formula is of the form $\forall x.F$, equivalent to $\neg(\exists x.\neg F)$
- In this case, apply QE algorithm to $\exists x.\neg F$ to obtain quantifier free formula F'
- Since F' is equivalent to $\exists x.\neg F$, $\forall x.F$ equivalent to $\neg F'$
- Thus, result of eliminating quantifier from $\forall x.F$ is $\neg F'$
- In either case, formula contains one less quantifier
- Repeat this process, removing innermost quantifier at each step

Example

- Suppose we have a procedure for eliminating quantifier from formula $\exists x.F$ where F is quantifier-free
- Let us see how to use it to eliminate quantifiers from formula

$$\exists x.\forall y.\exists z.F_1[x, y, z]$$

- Start with innermost quantified formula $\exists z.F_1[x, y, z]$
- Suppose QE elimination procedure returns $F_2[x, y]$
- Now, the formula is $\exists x.\forall y.F_2[x, y]$

Example, cont

- Current formula: $\exists x. \forall y. F_2[x, y]$
- Continue with innermost quantified formula $\forall y. F_2[x, y]$
- Rewrite it as $\neg \exists y. \neg F_2[x, y]$
- Apply QE algorithm to $\exists y. \neg F_2[x, y]$
- Suppose result is F_3; now formula is $\exists x. \neg F_3[x]$
- Now, apply QE procedure one last time to obtain quantifier-free formula

Summary

- As example illustrates, sufficient to have quantifier elimination procedure for $\exists x. F$
- Because this also allows us to eliminate universal quantifiers
- Thus, our QE procedure will only deal with existential quantifiers
- Furthermore, only talk about quantifier elimination in linear integer arithmetic

Theory of Integers

- Earlier we talked about theory of integers $T_{\mathbb{Z}}$ with signature:

$$\Sigma_{\mathbb{Z}} : \{..., -2, -1, 0, 1, 2, ..., +, -, =, <\}$$

- In this theory, we can write formulas such as: $\exists x.2x = y$
- What does this formula imply about y?

Theory of Integers

- Earlier we talked about theory of integers $T_{\mathbb{Z}}$ with signature:

$$\Sigma_{\mathbb{Z}} : \{..., -2, -1, 0, 1, 2, ..., +, -, =, <\}$$

- In this theory, we can write formulas such as: $\exists x. 2x = y$
- What does this formula imply about y? y is even

- Earlier we talked about theory of integers $T_{\mathbb{Z}}$ with signature:

$$\Sigma_{\mathbb{Z}} : \{..., -2, -1, 0, 1, 2, ..., +, -, =, <\}$$

- In this theory, we can write formulas such as: $\exists x.2x = y$
- What does this formula imply about y? y is even
- Similarly, $\exists w.3w = z$ expresses z is evenly divisible by 3
- Unfortunately, without additional divisibility predicate, we cannot write equivalent quantifier-free formula!
- Thus, this formulation of theory of integers does not admit quantifier elimination

Augmented Theory of Integers

- To admit quantifier elimination, we will add an additional divisibility predicates $k|\cdot$ to $T_{\mathbb{Z}}$ (k positive integer)
- Intended interpretation: $k|x$ is true if k evenly divides x
- According to this interpretation, is $x > 1 \wedge y > 1 \wedge 2|x + y$ satisfiable?

Augmented Theory of Integers

- To admit quantifier elimination, we will add an additional divisibility predicates $k|\cdot$ to $T_{\mathbb{Z}}$ (k positive integer)
- Intended interpretation: $k|x$ is true if k evenly divides x
- According to this interpretation, is $x > 1 \wedge y > 1 \wedge 2|x + y$ satisfiable? Yes, e.g., $x = 2, y = 2$

Augmented Theory of Integers

- To admit quantifier elimination, we will add an additional divisibility predicates $k|\cdot$ to $T_{\mathbb{Z}}$ (k positive integer)
- Intended interpretation: $k|x$ is true if k evenly divides x
- According to this interpretation, is $x > 1 \wedge y > 1 \wedge 2|x + y$ satisfiable? Yes, e.g., $x = 2, y = 2$
- What about $\neg(2|x) \wedge 4|x$?

Augmented Theory of Integers

- To admit quantifier elimination, we will add an additional divisibility predicates $k|\cdot$ to $T_{\mathbb{Z}}$ (k positive integer)
- Intended interpretation: $k|x$ is true if k evenly divides x
- According to this interpretation, is $x > 1 \wedge y > 1 \wedge 2|x + y$ satisfiable? Yes, e.g., $x = 2, y = 2$
- What about $\neg(2|x) \wedge 4|x$? No

- To admit quantifier elimination, we will add an additional divisibility predicates $k|\cdot$ to $T_{\mathbb{Z}}$ (k positive integer)
- Intended interpretation: $k|x$ is true if k evenly divides x
- According to this interpretation, is $x > 1 \land y > 1 \land 2|x + y$ satisfiable? Yes, e.g., $x = 2, y = 2$
- What about $\neg(2|x) \land 4|x$? No
- We will write $\widehat{T_{\mathbb{Z}}}$ to denote T with additional divisibility predicate and additional axiom:

$$\forall.x.k|x \leftrightarrow \exists y.x = ky$$

- Is $x|y$ well-formed formula in $\widehat{T_{\mathbb{Z}}}$?

Augmented Theory of Integers

- To admit quantifier elimination, we will add an additional divisibility predicates $k|\cdot$ to $T_{\mathbb{Z}}$ (k positive integer)
- Intended interpretation: $k|x$ is true if k evenly divides x
- According to this interpretation, is $x > 1 \land y > 1 \land 2|x + y$ satisfiable? Yes, e.g., $x = 2, y = 2$
- What about $\neg(2|x) \land 4|x$? No
- We will write $\widehat{T_{\mathbb{Z}}}$ to denote T with additional divisibility predicate and additional axiom:

$$\forall .x.k|x \leftrightarrow \exists y.x = ky$$

- Is $x|y$ well-formed formula in $\widehat{T_{\mathbb{Z}}}$? No!

Quantifier Elimination for $\widehat{T_{\mathbb{Z}}}$

- Fortunately, $\widehat{T_{\mathbb{Z}}}$ admits quantifier elimination Z
- Which quantifier-free formula is equivalent to $\exists x. 3x = y$?

- Fortunately, $\widehat{T_{\mathbb{Z}}}$ admits quantifier elimination Z
- Which quantifier-free formula is equivalent to $\exists x.3x = y$? $3|y$

Quantifier Elimination for $\widehat{T_{\mathbb{Z}}}$

- Fortunately, $\widehat{T_{\mathbb{Z}}}$ admits quantifier elimination Z
- Which quantifier-free formula is equivalent to $\exists x. 3x = y$? $3 | y$
- The quantifier elimination method for $\widehat{T_{\mathbb{Z}}}$ was given by Cooper in 1972 in a paper called Theorem Proving in Arithmetic without Multiplication
- Thus, known as Cooper's method
- Rest of lecture: Learn about Cooper's method

Overview of Cooper's Method

- Given $\widehat{T_{\mathbb{Z}}}$−formula $\exists x.F[x]$, where F is quantifier-free, Cooper's method constructs quantifier-free $\widehat{T_{\mathbb{Z}}}$-formula that is equivalent to $\exists x.F[x]$.
- Cooper's method has five main steps:
  1. Put $F[x]$ into NNF
  2. Normalize literals: $s < t, k | t$, or $\neg(k | t)$
  3. Isolate terms containing x on one side: $hx < t, s < hx$
  4. Ensure x has same coefficient d everywhere and replace dx with new variable x'
  5. Replace $F[x']$ with a disjunction of $F[j]$'s for finitiely many j

Step 1: Put Formula in Negation Normal Form

- A formula is in negation normal form (NNF) if the negation operator ($\neg$) is only applied to variables and predicates
- Recursively apply the following rules (left to right):
  - $\neg(\forall x.G) \Leftrightarrow \exists x.\neg G$
  - $\neg(\exists x.G) \Leftrightarrow \forall x.\neg G$
  - $\neg\neg G \Leftrightarrow G$
  - $\neg(G_1 \wedge G_2) \Leftrightarrow (\neg G_1) \vee (\neg G_2)$
  - $\neg(G_1 \vee G_2) \Leftrightarrow (\neg G_1) \wedge (\neg G_2)$

Example

- $\neg\forall x.\exists y.x > y \Leftrightarrow$
- $\exists x.\neg\exists y.x > y \Leftrightarrow$
- $\exists x.\forall y.\neg(x > y)$

Try it!

Convert $\neg\forall x.(x > z \vee \exists y.x > y)$ to NNF

Step 2: Normalize Literals: $s < t, k|t$, or $\neg(k|t)$

- Normalize literals so that every literal is of the form $s < t$, $k|t$, or $\neg(k|t)$
- To do this, we need to rewrite $s = t$, $\neg(s = t)$, and $\neg(s < t)$ as a boolean combination of literals of the form $s' < t'$
- Rewrite rules:
  1. $s = t \Leftrightarrow s < t + 1 \wedge t < s + 1$
  2. $\neg(s = t) \Leftrightarrow s < t \vee t < s + 1$
  3. $\neg(s < t) \Leftrightarrow t < s + 1$

Example

- Let us normalize literals in the following formula:

$$\neg(x < y) \wedge \neg(x = y + 3)$$

- $\neg(x < y) \Leftrightarrow y < x + 1$
- $\neg(x = y + 3) \Leftrightarrow x < y + 3 \vee y + 3 < x$
- Normalized formula after step 2:

$$y < x + 1 \wedge (x < y + 3 \vee y + 3 < x)$$

Step 3: Collect Terms Containing x on One Side

- After step 3, literals should be of one of the following forms:

$$hx < t, t < hx, k|hx + t, \neg(k|hx + t)$$

  where t is a term not containing x and h,k are positive

- Example: Let us apply this transformation to the formula:

$$x + x + y < z + 3z + 2y - 4x$$

- Result: $6x < 4z + y$
- Example: $5|(-7x + t)$
- After applying transformation,we get: $5|(7x - t)$

Step 4a: Ensure x Has the Same Coefficient Everywhere

- After previous step, formula is of the form $\exists x. F_3[x]$
- Compute least common multiple (lcm) of x's coefficients:

$$d = lcm\{h : h \text{ is coefficient of } x \text{ in } F_3[x]\}$$

- Now, multiply literals in $F_3[x]$ by constants so that x's coefficient is d everywhere:

$$\begin{array}{rcll} hx < t \Leftrightarrow & dx < h't & \text{where } d = hh' \\ t < hx \Leftrightarrow & h't < dx & \text{where } d = hh' \\ k \mid (hx + t) \Leftrightarrow & h'k \mid (dx + h't) & \text{where } d = hh' \\ \neg(k \mid (hx + t)) \Leftrightarrow & \neg(h'k \mid (dx + h't) & \text{where } d = hh' \end{array}$$

Example

- Consider the formula

$$2x < y \vee (2z < 3x \wedge 3|(4x+1))$$

- What is the lcm of x's coefficients in this formula? 12
- Rewrite each literal so that x has coefficient 12:

$$\begin{aligned} 2x < y &\Leftrightarrow 12x < 6y \\ 2z < 3x &\Leftrightarrow 8z < 12x \\ 3|(4x+1) &\Leftrightarrow 9|(12x+3) \end{aligned}$$

- New formula after transformation:

$$12x < 6y \vee (8z < 12x \wedge 9|(12x+3))$$

Step 4b: Replace dx with New Variable x'

- After Step 4a, variable x has the same coefficient d everywhere
- Now, we replace dx with a new variable x'
- Since x' is implicitly equal to dx, what can we say about x'?

Step 4b: Replace dx with New Variable x'

- After Step 4a, variable x has the same coefficient d everywhere
- Now, we replace dx with a new variable x'
- Since x' is implicitly equal to dx, what can we say about x'?
  x' must be divisible by d

Step 4b: Replace dx with New Variable x'

- After Step 4a, variable x has the same coefficient d everywhere
- Now, we replace dx with a new variable x'
- Since x' is implicitly equal to dx, what can we say about x'? x' must be divisible by d
- Thus, we also add the constraint $d|x'$
- Example: Consider previous formula after Step 4a:

$$12x < 6y \vee (8z < 12x \wedge 9|(12x + 3))$$

- What is the resulting formula after this step?

$$(x' < 6y \vee (8z < x' \wedge 9|(x' + 3)))\wedge(12|x')$$

Formula after Step 4b

- After this step, formula is of the form $\exists x'.F_4[x']$
- Furthermore $\exists x'.F_4[x']$ is equivalent to $\exists x.F[x]$
- In addition, each literal in $\exists x'.F_4[x']$ is one of the following:
  1. $x' < a$
  2. $b < x'$
  3. $h | (x' + c)$
  4. $\neg(k | (x' + d))$
- Here, a, b, c, d do not contain x and h, k are positive

Step 5: Intuition

- Most involved part of Cooper's method
- Recall: We want to eliminate x' from the formula $\exists x'.F_4[x']$
- There are two possibilities:
  1. Either infinitely many small numbers n satisfying $F_4[n]$
  2. Or there exists a least integer n that satisfies $F_4[n]$
- Step 5 of Cooper's method is a case analysis on these two possibilities

Step 5a: Left Infinite Projection

- We want to eliminate x' from $\exists x'.F_4[x']$ under the assumption there are infinitely many small numbers n satisfying $F_4[n]$
- Thus, define left infinite projection $F_{-\infty}[x']$ for formula $F_4[x']$
- $F_{-\infty}[x']$ corresponds to projection of F that is only satisfied by very small values of x'
- Called left infinite projection because very small numbers correspond to left part of number line approaching infinity
- To compute left infinite projection:
  1. Replace literals $x' < a$ by

Step 5a: Left Infinite Projection

- We want to eliminate x' from $\exists x'.F_4[x']$ under the assumption there are infinitely many small numbers n satisfying $F_4[n]$
- Thus, define left infinite projection $F_{-\infty}[x']$ for formula $F_4[x']$
- $F_{-\infty}[x']$ corresponds to projection of F that is only satisfied by very small values of x'
- Called left infinite projection because very small numbers correspond to left part of number line approaching infinity
- To compute left infinite projection:
  1. Replace literals $x' < a$ by $\top$
  2. Replace literals $b < x'$ by

Step 5a: Left Infinite Projection

- We want to eliminate x' from $\exists x'.F_4[x']$ under the assumption there are infinitely many small numbers n satisfying $F_4[n]$
- Thus, define left infinite projection $F_{-\infty}[x']$ for formula $F_4[x']$
- $F_{-\infty}[x']$ corresponds to projection of F that is only satisfied by very small values of x'
- Called left infinite projection because very small numbers correspond to left part of number line approaching infinity
- To compute left infinite projection:
  1. Replace literals $x' < a$ by $\top$
  2. Replace literals $b < x'$ by $\bot$

Step 5a, cont

- In $F_{-\infty}[x']$, no literals of the form $x' < a$ and $b < x'$ because for very small numbers they evaluate to true or false
- But we still have divisibility predicates of the form

$$h | (x' + c) \text{ and } \neg(k | x' + d)$$

- Unfortunately, can't just replace these with $\top$ or $\perp$. Why?

- In $F_{-\infty}[x']$, no literals of the form $x' < a$ and $b < x'$ because for very small numbers they evaluate to true or false
- But we still have divisibility predicates of the form

$$h|(x' + c) \text{ and } \neg(k|x' + d)$$

- Unfortunately, can't just replace these with $\top$ or $\perp$. Why?
- Because for an arbitrary very small number, these divisibility predicates need not hold
- Thus, want to figure out if there exists a very small number satisfying divisibility predicates

Step 5a, cont

- Good news: If there exists a very small number satisfying divisibility constraints, there must also exist a number in a finite precomputable range $[1, \delta]$ satisfying these predicates
- This is known as peridocity property of divisibility predicates
- Periodicity property: Suppose $m|\delta$, then, $m|n$ iff $m|(n + \lambda\delta)$ for all integers λ
- In other words, divisibility by m cannot distinguish between numbers n and $n + \lambda\delta$
- Thus, if some very small number satisfies divisibility constraints in $F_{-\infty}$, there must exist a number $n \in [1, \delta]$
- But what is this δ?

Step 5a, cont

- Consider two literals of the form $k|x'$ and $m|x'$
- We want to find the smallest number δ such that both $k|\delta$ and $m|\delta$
- What number has this property?

Step 5a, cont

- Consider two literals of the form $k|x'$ and $m|x'$
- We want to find the smallest number δ such that both $k|\delta$ and $m|\delta$
- What number has this property? lcm(k, m)

- Consider two literals of the form $k|x'$ and $m|x'$
- We want to find the smallest number δ such that both $k|\delta$ and $m|\delta$
- What number has this property? lcm(k, m)
- Thus, δ should be the least common multiple of the LHS of divisibility constraints
- Specifically:

$$\delta = lcm \left(\begin{array}{l} h \text{ of literals } h|(x'+c) \\ k \text{ of literals } \neg(k|(x'+d)) \end{array} \right)$$

- Thus, to determine if there exists a very small number n satisfying $F_{-\infty}$, sufficient to numbers in the range $[0, \delta]$

Step 5a, Summary

- Assume infinitely many small numbers satisfy $\exists x'. F_4[x']$
- First compute left infinite projection $F_{-\infty}$ of F_4
- Cooper's result: $\exists x'. F_4$ is satisfiable iff there exists n in the range $[1, \delta]$ satisfying $F_{-\infty}$, i.e.,:

$$\bigvee_{j=1}^{\delta} F_{-\infty}[j]$$

- Under the assumption there are infinitely many small numbers satisfying $\exists x. F[x]$, we have the equivalence:

$$\exists x. F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F_{-\infty}[j]$$

Step 5b: Exists a Least Satisfying Number

- Now, let's consider case with a least number satisfying $F_4[x']$
- Recall: All the inequality literals are either $x' < a$ or $b < x'$
- If there is a least number satisfying $F_4[x']$, one of these inequality literals must be responsible for it
- Can a literal $x' < a$ be responsible for this least number?

Step 5b: Exists a Least Satisfying Number

- Now, let's consider case with a least number satisfying $F_4[x']$
- Recall: All the inequality literals are either $x' < a$ or $b < x'$
- If there is a least number satisfying $F_4[x']$, one of these inequality literals must be responsible for it
- Can a literal $x' < a$ be responsible for this least number? No because $x' < a$ satisfied no matter how small x' is

Step 5b: Exists a Least Satisfying Number

- Now, let's consider case with a least number satisfying $F_4[x']$
- Recall: All the inequality literals are either $x' < a$ or $b < x'$
- If there is a least number satisfying $F_4[x']$, one of these inequality literals must be responsible for it
- Can a literal $x' < a$ be responsible for this least number? No because $x' < a$ satisfied no matter how small x' is
- Thus,if there is least value of x', it is due to some $b < x'$
- Thus, disregarding divisibility constraints, least number satisfying $F_4[x']$ must be one of these b's!

Step 5b, cont

- Now, let's take the divisibility constraints into account
- Because of the divisibility constraints, least number satisfying $F_4[x']$ might not be exactly b
- It might be greater than b to satisfy divisibility constraints
- But it can't be greater than $b + \delta$ (δ same as before). Why?

- Now, let's take the divisibility constraints into account
- Because of the divisibility constraints, least number satisfying $F_4[x']$ might not be exactly b
- It might be greater than b to satisfy divisibility constraints
- But it can't be greater than $b + \delta$ (δ same as before). Why?
- Because of periodicity, if there is no number in the range $[b, b + \lambda]$, there can't be number greater than $b + \lambda$ satisfying divisibility constraints
- Thus, assuming some literal $b < x'$ is limiting factor, $\exists x'.F_4[x']$ has solution iff:

$$\bigvee_{j=1}^{\delta} F_4[b + j]$$

Step 5b, cont

- Not done yet because we don't know which literal of the form $b < x'$ is the most constraining literal
- Suppose we have n literals $b_1 < x', b_2 < x', \ldots, b_n < x'$
- We need to take into the possibility that any of them could be most constraining
- Thus, assuming there is a least number satisfying $F_4[x]$, $\exists x.F[x]$ equivalent to:

$$\bigvee_{i=1}^{n} \bigvee_{j=1}^{\delta} F_4[b_i + j]$$

Step 5, summary

- Now, let's combine the two case analysis
- Assuming $F[x]$ satisfied by infinitely many small x, we have:

$$\exists x.F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F_{-\infty}[j]$$

- Assuming there is least x satisfying $F[x]$, we have:

$$\exists x.F[x] \Leftrightarrow \bigvee_{i=1}^{n} \bigvee_{j=1}^{\delta} F_4[b_i + j]$$

- Combining these two, we get the final result of step 5:

$$\exists x.F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F_{-\infty}[j] \vee \bigvee_{i=1}^{n} \bigvee_{j=1}^{\delta} F_4[b_i + j]$$

Example

- Use Cooper's method to eliminate quantifier from:

$$\exists x. -y < 3x - 2y + 1 \wedge 2x - 6 < z \wedge 2 | (x + 1)$$

- Step 1: Already in NNF
- Step 2: Already normalized
- Step 3: Collect x-terms on one side:

$$\exists x. y - 1 < 3x \wedge 2x < z + 6 \wedge 2 | (x + 1)$$

- Step 4a: Make coefficients of x equal everywhere
- What is lcm of x's coefficients?

Example

- Use Cooper's method to eliminate quantifier from:

$$\exists x. -y < 3x - 2y + 1 \land 2x - 6 < z \land 2 | (x+1)$$

- Step 1: Already in NNF
- Step 2: Already normalized
- Step 3: Collect x-terms on one side:

$$\exists x. y - 1 < 3x \land 2x < z + 6 \land 2 | (x+1)$$

- Step 4a: Make coefficients of x equal everywhere
- What is lcm of x's coefficients? 6

$$\exists x. y - 1 < 3x \wedge 2x < z + 6 \wedge 2 | (x + 1)$$

- Multiply literals so that x has coefficient 6 everywhere:

$$\exists x. 2y - 2 < 6x \wedge 6x < 3z + 18 \wedge 12 | (6x + 6)$$

- Step 4b: Replace $6x$ with x'; add divisibility constraint $6 | x'$
- Formula after step 4:

$$\exists x'. 2y - 2 < x' \wedge x' < 3z + 18 \wedge 12 | (x' + 6) \wedge 6 | x'$$

Example, cont

$$\exists x'. 2y - 2 < x' \wedge x' < 3z + 18 \wedge 12 | (x' + 6) \wedge 6 | x'$$

- Step 5a: Assume there are infinitely many small numbers satisfying formula
- Construct left infinite projection:

$$F_{-\infty} : \bot \wedge \top \wedge 12 | (x' + 6) \wedge 6 | x'$$

- This simplifies to $\bot$
- Step 5b: Assume there is least number satisfying formula
- Which inequalities could be responsible for least n?

Example, cont

$$\exists x'.2y - 2 < x' \land x' < 3z + 18 \land 12 | (x' + 6) \land 6 | x'$$

- Step 5a: Assume there are infinitely many small numbers satisfying formula
- Construct left infinite projection:

$$F_{-\infty} : \bot \land \top \land 12 | (x' + 6) \land 6 | x'$$

- This simplifies to $\bot$
- Step 5b: Assume there is least number satisfying formula
- Which inequalities could be responsible for least n? $2y - 2 < x'$

Example, cont

$$\exists x'.2y - 2 < x' \wedge x' < 3z + 18 \wedge 12|(x' + 6) \wedge 6|x'$$

- Thus, if there is solution, must lie in range $[2y - 2, 2y - 2 + \delta]$
- What is δ here?

Example, cont

$$\exists x'.2y-2<x'\wedge x'<3z+18\wedge 12|(x'+6)\wedge 6|x'$$

- Thus, if there is solution, must lie in range $[2y-2, 2y-2+\delta]$
- What is δ here? 12
- Now putting everything together, we get:

$$\bigvee_{j=1}^{12}.0<j\wedge 2y+j<3z+20\wedge 12|(2y+j+4)\wedge 6|2y-2+j$$

Example 2

- Apply Cooper's method to $\exists x. 2x = y$ (already in NNF)
- Step 2: Normalize literals:

$$\exists x. y < 2x + 1 \wedge 2x < y + 1$$

- Step 3: Collect x on one side:

$$\exists x. y - 1 < 2x \wedge 2x < y + 1$$

- Step 4a: x's coefficients already same everywhere
- Step 4b: Replace $2x$ with x'; add divisibility constraint: $2|x'$

$$\exists x'. y - 1 < x' \wedge x' < y + 1 \wedge 2|x'$$

Example 2,cont

$$\exists x'.y - 1 < x' \land x' < y + 1 \land 2|x'$$

- Step 5a: Compute left infinite projection: $\perp$
- Step 5b: Assume there is a least n satisfying formula
- Which literal could be responsible?

Example 2,cont

$$\exists x'. y - 1 < x' \land x' < y + 1 \land 2 | x'$$

- Step 5a: Compute left infinite projection: $\perp$
- Step 5b: Assume there is a least n satisfying formula
- Which literal could be responsible? $y - 1 < x'$

Example 2,cont

$$\exists x'. y - 1 < x' \wedge x' < y + 1 \wedge 2 | x'$$

- Step 5a: Compute left infinite projection: $\perp$
- Step 5b: Assume there is a least n satisfying formula
- Which literal could be responsible? $y - 1 < x'$
- In what range must this least n be?

Example 2,cont

$$\exists x'. y - 1 < x' \wedge x' < y + 1 \wedge 2 | x'$$

- Step 5a: Compute left infinite projection: $\perp$
- Step 5b: Assume there is a least n satisfying formula
- Which literal could be responsible? $y - 1 < x'$
- In what range must this least n be? $[y - 1, y - 1 + 2]$

Example 2,cont

$$\exists x'. y-1 < x' \wedge x' < y+1 \wedge 2|x'$$

- Step 5a: Compute left infinite projection: $\perp$
- Step 5b: Assume there is a least n satisfying formula
- Which literal could be responsible? $y-1 < x'$
- In what range must this least n be? $[y-1, y-1+2]$
- Thus, x' must be one of $y-1, y, y+1$

Example 2,cont

$$\exists x'.y - 1 < x' \wedge x' < y + 1 \wedge 2|x'$$

- x' must be one of $y-1, y, y+1$
- Plug in $y-1$ for x', we get:

Example 2,cont

$$\exists x'.y - 1 < x' \wedge x' < y + 1 \wedge 2 | x'$$

- x' must be one of $y-1, y, y+1$
- Plug in $y-1$ for x', we get: $\perp$

Example 2,cont

$$\exists x'. y - 1 < x' \wedge x' < y + 1 \wedge 2 | x'$$

- x' must be one of $y - 1, y, y + 1$
- Plug in $y - 1$ for x', we get: $\perp$
- Plug in y for x', we get:

Example 2,cont

$$\exists x'.y - 1 < x' \wedge x' < y + 1 \wedge 2|x'$$

- x' must be one of $y - 1, y, y + 1$
- Plug in $y - 1$ for x', we get: $\perp$
- Plug in y for x', we get: $2|y$

Example 2,cont

$$\exists x'.y - 1 < x' \wedge x' < y + 1 \wedge 2|x'$$

- x' must be one of $y - 1, y, y + 1$
- Plug in $y - 1$ for x', we get: $\bot$
- Plug in y for x', we get: $2|y$
- Plug in $y + 1$ for x', we get:

Example 2,cont

$$\exists x'.y-1 < x' \wedge x' < y+1 \wedge 2|x'$$

- x' must be one of $y-1, y, y+1$
- Plug in $y-1$ for x', we get: $\perp$
- Plug in y for x', we get: $2|y$
- Plug in $y+1$ for x', we get: $\perp$

Example 2,cont

$$\exists x'. y - 1 < x' \wedge x' < y + 1 \wedge 2 | x'$$

- x' must be one of $y - 1, y, y + 1$
- Plug in $y - 1$ for x', we get: $\perp$
- Plug in y for x', we get: $2 | y$
- Plug in $y + 1$ for x', we get: $\perp$
- Thus, formula equivalent to:

Example 2,cont

$$\exists x'.y - 1 < x' \land x' < y + 1 \land 2|x'$$

- x' must be one of $y - 1, y, y + 1$
- Plug in $y - 1$ for x', we get: $\perp$
- Plug in y for x', we get: $2|y$
- Plug in $y + 1$ for x', we get: $\perp$
- Thus, formula equivalent to: $2|y$

An Alternative Construction

- To produce equivalent formula, we performed a case analysis:
  1. Either there are infinitely many very small numbers satisfying it
  2. Or there exists a least number satisfying it
- But we could have also performed the case analysis this way:
  1. Either there are infinitely many very large numbers satisfying it
  2. Or there exists a greatest number satisfying it

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\perp$

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\perp$
  2. Replace $b < x'$ with

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\bot$
  2. Replace $b < x'$ with $\top$

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\perp$
  2. Replace $b < x'$ with $\top$
- For the second case (i.e., greatest number), which literals must be responsible?

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\perp$
  2. Replace $b < x'$ with $\top$
- For the second case (i.e., greatest number), which literals must be responsible? $x' < a$

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\bot$
  2. Replace $b < x'$ with $\top$
- For the second case (i.e., greatest number), which literals must be responsible? $x' < a$
- If literal $x' < a$ is responsible for greatest satisfying number, in which range must this greatest number lie?

Alternative Case Analysis

- Let's see what happens using this alternative case analysis
- For the first case, we construct $F_{+\infty}$ instead of $F_{-\infty}$
  1. Replace $x' < a$ with $\bot$
  2. Replace $b < x'$ with $\top$
- For the second case (i.e., greatest number), which literals must be responsible? $x' < a$
- If literal $x' < a$ is responsible for greatest satisfying number, in which range must this greatest number lie? $[a - \delta, a]$

An Optimization

- Using this alternative construction, we obtain the equivalence:

$$\exists x. F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F_{+-\infty}[j] \vee \bigvee_{i=1}^{k} \bigvee_{j=1}^{\delta} F_4[a_i - j]$$

- This immediately gives a way to optimize Cooper's method
- Observe: If there are n terms of the form $b < x'$, we get n disjuncts using left infinite projection
- Observe: If there are k terms of the form $x' < a$, we get k disjuncts using right infinite projection

An Optimization

- Using this alternative construction, we obtain the equivalence:

$$\exists x. F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F + -\infty[j] \vee \bigvee_{i=1}^{k} \bigvee_{j=1}^{\delta} F_4[a_i - j]$$

- This immediately gives a way to optimize Cooper's method
- Observe: If there are n terms of the form $b < x'$, we get n disjuncts using left infinite projection
- Observe: If there are k terms of the form $x' < a$, we get k disjuncts using right infinite projection
- Thus, if there are more terms of the form $b < x'$, advantageous to use

An Optimization

- Using this alternative construction, we obtain the equivalence:

$$\exists x. F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F_{+-\infty}[j] \vee \bigvee_{i=1}^{k} \bigvee_{j=1}^{\delta} F_4[a_i - j]$$

- This immediately gives a way to optimize Cooper's method
- Observe: If there are n terms of the form $b < x'$, we get n disjuncts using left infinite projection
- Observe: If there are k terms of the form $x' < a$, we get k disjuncts using right infinite projection
- Thus, if there are more terms of the form $b < x'$, advantageous to use $F_{+\infty}$

An Optimization

- Using this alternative construction, we obtain the equivalence:

$$\exists x.F[x] \Leftrightarrow \bigvee_{j=1}^{\delta} F+-\infty[j] \vee \bigvee_{i=1}^{k} \bigvee_{j=1}^{\delta} F_4[a_i - j]$$

- This immediately gives a way to optimize Cooper's method
- Observe: If there are n terms of the form $b < x'$, we get n disjuncts using left infinite projection
- Observe: If there are k terms of the form $x' < a$, we get k disjuncts using right infinite projection
- Thus, if there are more terms of the form $b < x'$, advantageous to use $F_{+\infty}$
- If there are more $x' < a$ terms, better to use $F_{-\infty}$.

Example

- Consider the formula:

$$\exists x.(x < 13 \vee 15 < x) \wedge x < y$$

- Which projection is better?

Example

- Consider the formula:

$$\exists x.(x < 13 \vee 15 < x) \wedge x < y$$

- Which projection is better? left infinite

Example

- Consider the formula:

$$\exists x.(x < 13 \lor 15 < x) \land x < y$$

- Which projection is better? left infinite
- There are two terms of the form $x < a$ forming upper bound on x: construction using $F_{+\infty}$ has 2 disjuncts
- There is one term of the form $b < x$ forming lower bound: construction using $F_{-\infty}$ has one disjunct
- Thus, left infinite projection yields smaller formula

Theory of Rationals

- Now we discuss theory of rationals $T_{\mathbb{Q}}$ with signature:

$$\Sigma_{\mathbb{Q}} : \{..., -2, -1, 0, 1, 2, ..., +, -, =, <\}$$

- Quantifier elimination for $T_{\mathbb{Q}}$ is simpler than $\widehat{T_{\mathbb{Z}}}$
- The algorithm is called Ferrante and Rackoff's method
- The idea is very similar to Cooper's method

Ferrante and Rackoff's method

- Ferrante and Rackoff's method has four main steps:
  1. Put $F[x]$ into NNF
  2. Normalize literals:
     - $\neg(s < t) \Leftrightarrow t < s \vee t = s$
     - $\neg(s = t) \Leftrightarrow t < s \vee t > s$
  3. Isolate terms containing x on one side: $hx < t, s < hx$ and replace literals $cx \odot t$ with $x \odot t/c$, for $\odot \in \{>, =, <\}$.
  4. Replace $F[x]$ with a disjunction of $F[j]$'s for finitiely many j

Formula after Step 3

- After step 3, formula is of the form $\exists x. F_3[x]$
- Furthermore $\exists x. F_3[x]$ is equivalent to $\exists x. F[x]$
- In addition, each literal in $\exists x. F_3[x]$ is one of the following:
  - A $x < a$
  - B $b < x$
  - C $x = c$
- Here, a, b, c do not contain x

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by $\top$
  2. Replace literals $b < x$ by

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by $\top$
  2. Replace literals $b < x$ by $\perp$
  3. Replace literals $c = x$ by

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by $\top$
  2. Replace literals $b < x$ by $\bot$
  3. Replace literals $c = x$ by $\bot$
- To compute right infinite projection $F_{+\infty}$:
  1. Replace literals $x < a$ by

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by $\top$
  2. Replace literals $b < x$ by $\bot$
  3. Replace literals $c = x$ by $\bot$
- To compute right infinite projection $F_{+\infty}$:
  1. Replace literals $x < a$ by $\bot$
  2. Replace literals $b < x$ by

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by $\top$
  2. Replace literals $b < x$ by $\bot$
  3. Replace literals $c = x$ by $\bot$
- To compute right infinite projection $F_{+\infty}$:
  1. Replace literals $x < a$ by $\bot$
  2. Replace literals $b < x$ by $\top$
  3. Replace literals $c = x$ by

Step 4a: Left and Right Infinite Projection

- To compute left infinite projection $F_{-\infty}$:
  1. Replace literals $x < a$ by $\top$
  2. Replace literals $b < x$ by $\bot$
  3. Replace literals $c = x$ by $\bot$
- To compute right infinite projection $F_{+\infty}$:
  1. Replace literals $x < a$ by $\bot$
  2. Replace literals $b < x$ by $\top$
  3. Replace literals $c = x$ by $\bot$

Step 4b: Remove Quantifiers

- Let S be the set of a, b, c terms for the A, B, C atoms in F_3
- The final result:

$$\exists x.F[x] \Leftrightarrow F_{+\infty} \vee F_{+\infty} \vee \bigvee_{s,t \in S} F_3[\frac{s+t}{2}]$$

- Intuition: for any $T_{\mathbb{Q}}$-interpretation, $|S| - 1$ pairs $s, t \in S$ are adjacent; $\frac{s+t}{2}$ is indistinguishable with any other point in the interval (s, t).