CSE4051: Program Verification

Theory Solvers

2025 Fall

Woosuk Lee

Review: First-Order Theories

- A first-order theory T is defined by the two components:
  - **Signature**: a set of nonlogical symbols. Given a signature Σ, a Σ-formula is one whose nonlogical symbols are from Σ. Signature restricts the syntax.
  - **Axioms**: A set of closed FOL formulas whose nonlogical symbols are from Σ. Axioms restrict the interpretations.

Theory Solver

- Decides satisfiability of a formula in a theory
- In this lecture, we only consider quantifier-free & conjunctive fragments.
  - There are various techniques for removing quantifiers such as quantifier instantiation or quantifier elimination.
  - Formulas containing disjunctions can be handled by **DPLL(T)** (will be covered later)

Review: Theory of Equality

- A theory with a fixed interpretation for =. For example, the formula must be valid according to the conventional interpretation of =:

$$\forall x, y, z\,.\,(((x = y) \land \neg(y = z)) \implies \neg(x = z))$$

- To fix this interpretation, it is sufficient to enforce the following axioms:
  - Reflexivity: $\forall x\,.\,x = x$
  - Symmetry: $\forall x, y\,.\,x = y \implies y = x$
  - Transitivity: $\forall x, y, z\,.\,x = y \land y = z \implies x = z$
  - …

Review: Theory of Equality (T_E)

- …
- Function congruence (for each positive integer n and n-ary function symbol f):

$$\forall \overline{x}, \overline{y}. \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \rightarrow f(\overline{x}) = f(\overline{y})$$

$\overline{x}$: list of variables $x_1, \ldots, x_n$

- Predicate congruence (for each positive integer n and n-ary predicate symbol p):

$$\forall \overline{x}, \overline{y}. \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \rightarrow (p(\overline{x}) \leftrightarrow p(\overline{y}))$$

$\leftrightarrow$: $\Rightarrow$ and $\Leftarrow$

- Meaning: no matter what functions and predicates are used, if the inputs are the same, the outcomes are also the same.

Eliminating Predicates in T_E

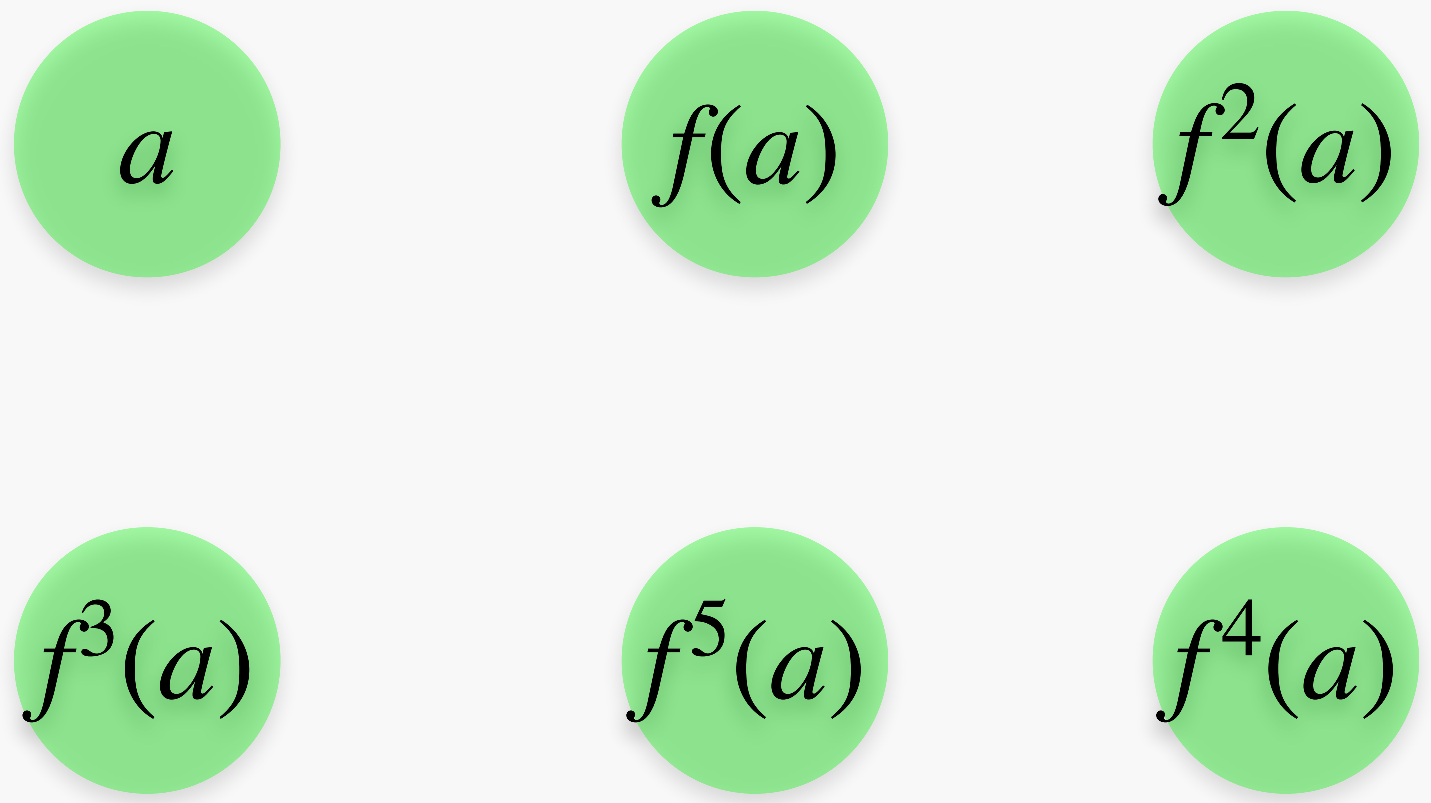
- Let's remove predicates. For each predicate p, rewrite $p(x_1, \ldots, x_n)$ as $f_p(x_1, \ldots, x_n) = t$ for a fresh function symbol f_p and variable t.
- Then, axioms are
  - Reflexivity: $\forall x . x = x$
  - Symmetry: $\forall x, y . x = y \implies y = x$
  - Transitivity: $\forall x, y, z . x = y \land y = z \implies x = z$
  - Function congruence (for each positive integer n and n-ary function symbol f):

$$\forall \overline{x}, \overline{y}. \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \rightarrow f(\overline{x}) = f(\overline{y})$$

Congruence Closure Algorithm for T_E

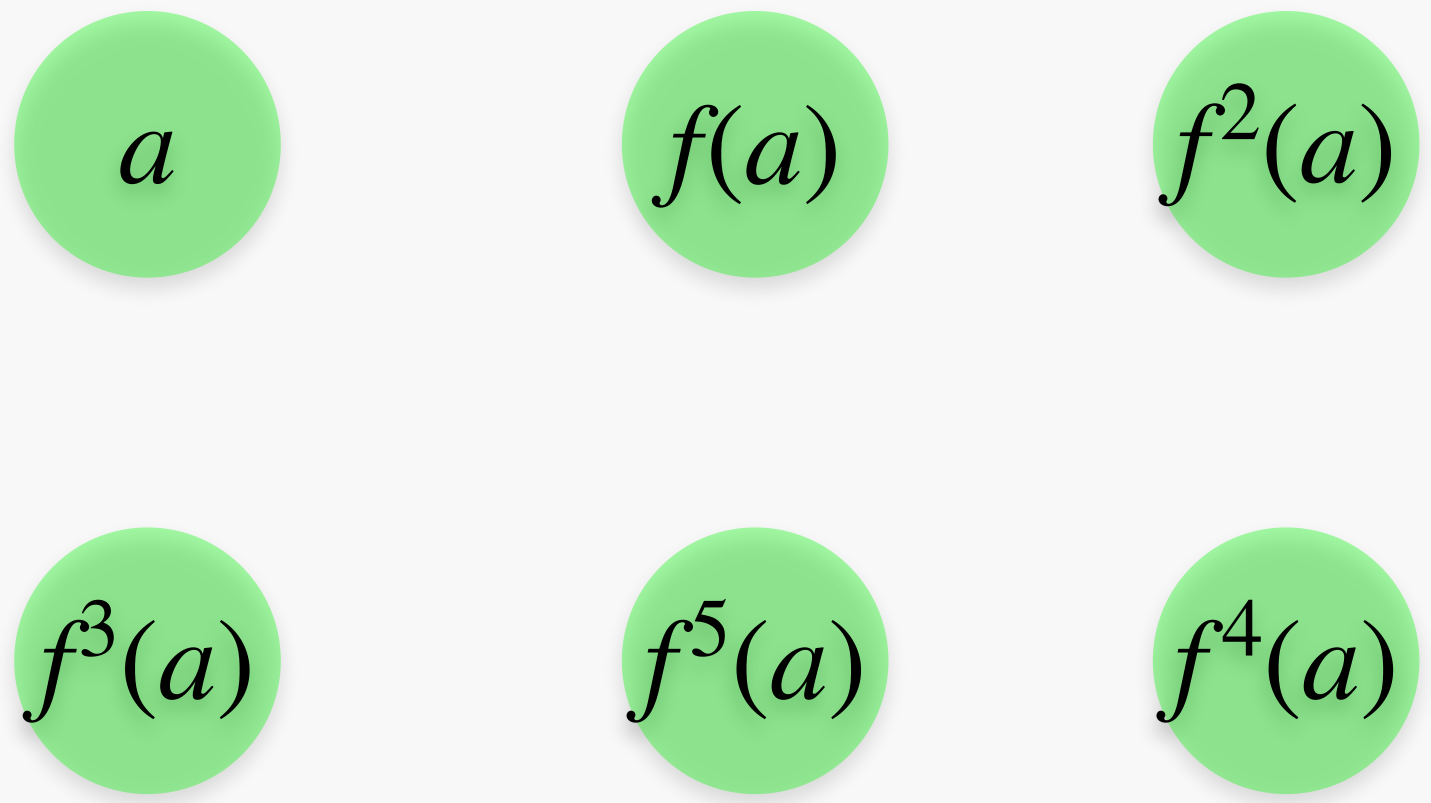
- Consider $F : f^3(a) = a \wedge f^5(a) = a \wedge f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))}_{n}\cdots)$

- Place each atom of F into its own group



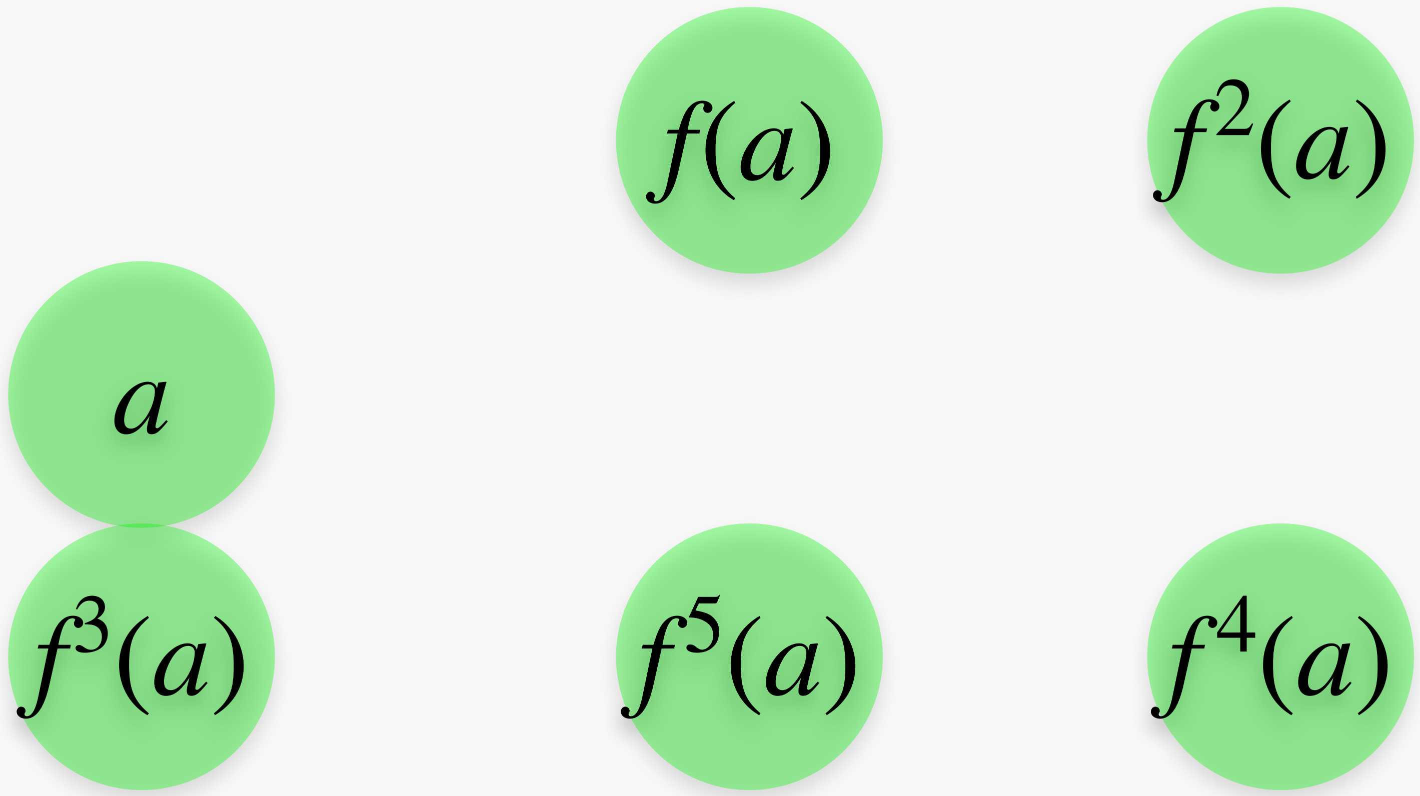
Congruence Closure Algorithm for T_E

- Consider $F : f^3(a) = a \land f^5(a) = a \land f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))\cdots)}_{n}$
- For each positive literal $t_1 = t_2$ in F



Congruence Closure Algorithm for T_E

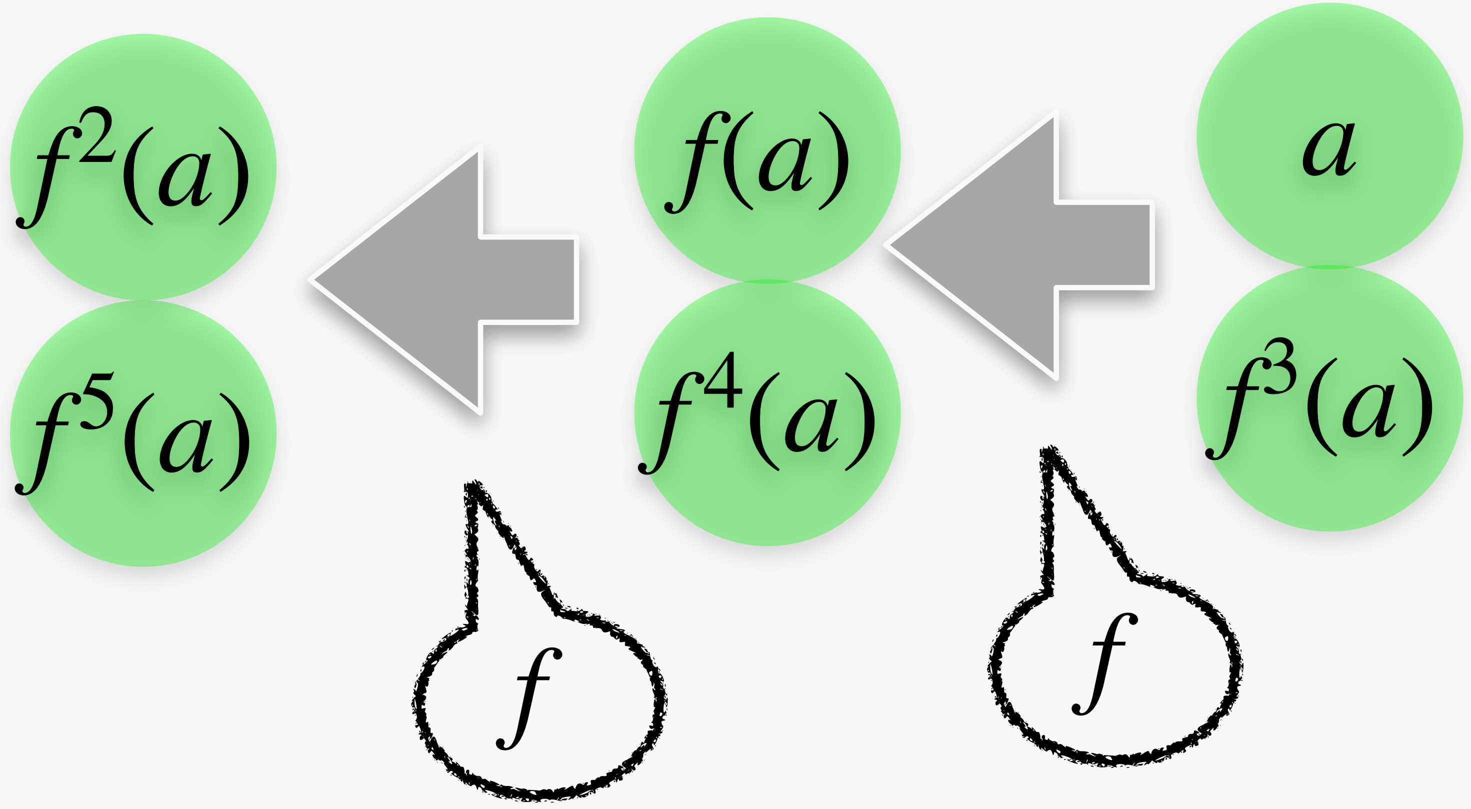
- Consider $F : f^3(a) = a \wedge f^5(a) = a \wedge f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))}_{n}\cdots)$
- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2



Congruence Closure Algorithm for T_E

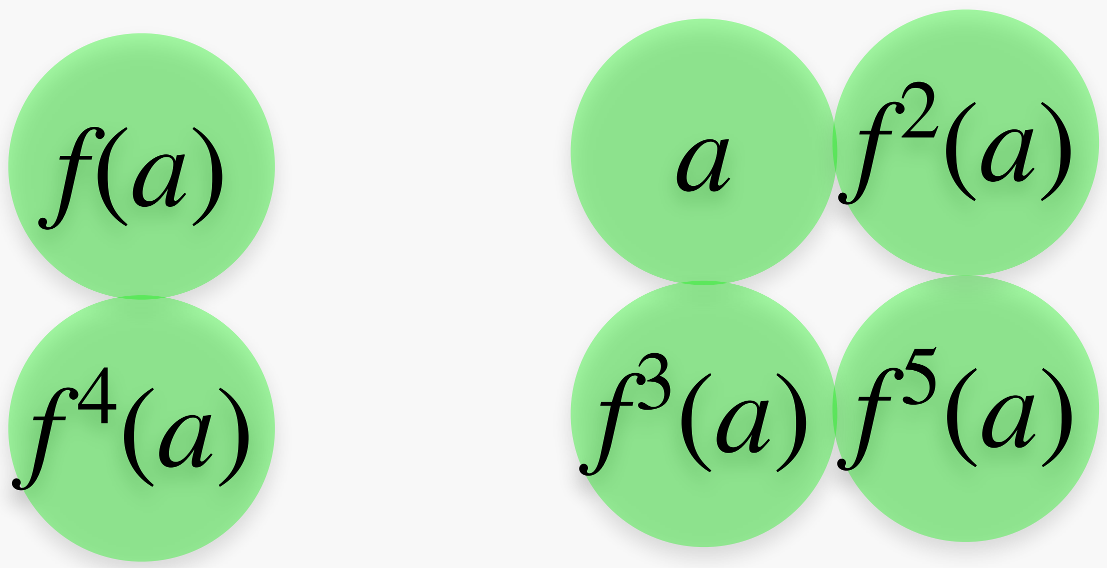
- Consider $F : f^3(a) = a \wedge f^5(a) = a \wedge f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))\cdots)}_{n}$

- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities



Congruence Closure Algorithm for T_E

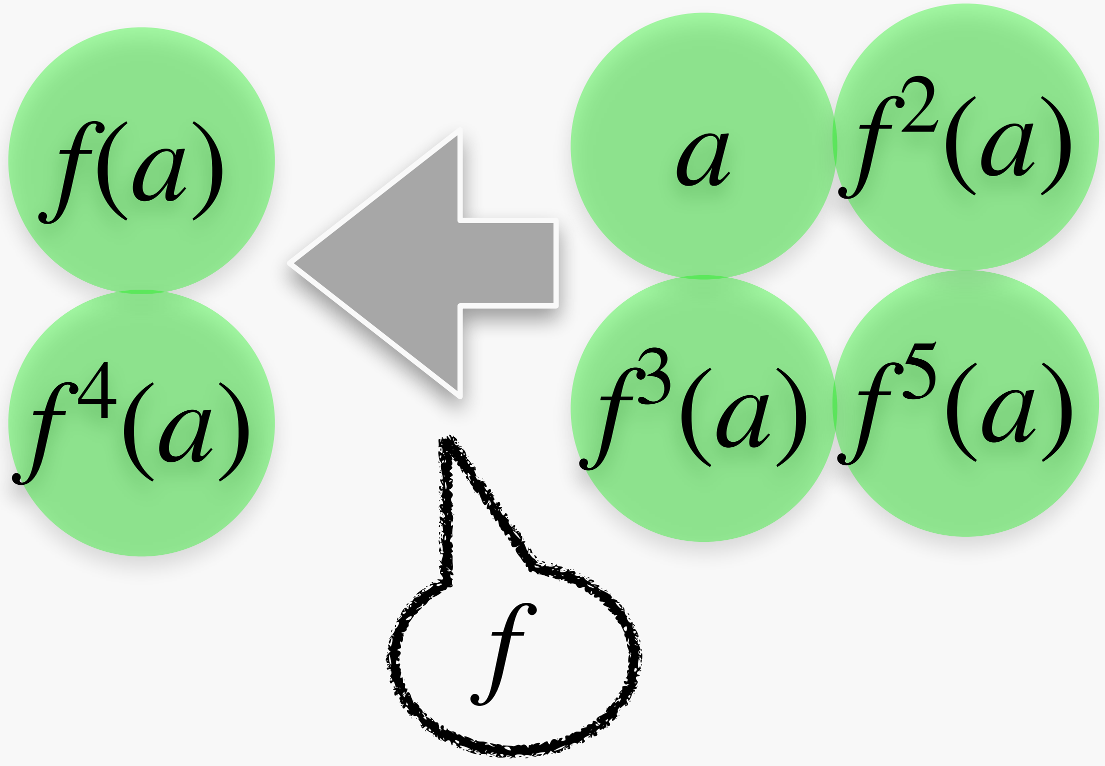
- Consider $F : f^3(a) = a \land f^5(a) = a \land f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))}_{n}\cdots)$
- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities



Congruence Closure Algorithm for T_E

- Consider $F : f^3(a) = a \wedge f^5(a) = a \wedge f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))\cdots)}_{n}$

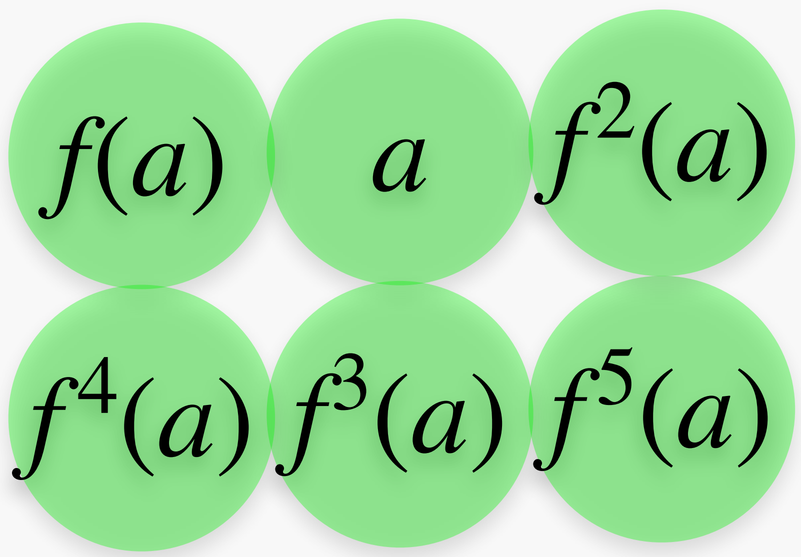
- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities



Congruence Closure Algorithm for T_E

- Consider $F : f^3(a) = a \land f^5(a) = a \land f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))}_{n}\cdots)$

- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities

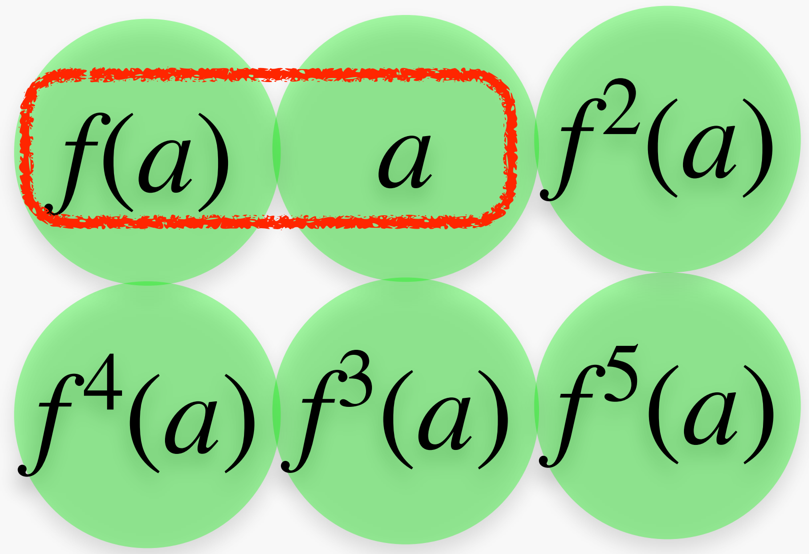


Congruence Closure Algorithm for T_E

- Consider $F : f^3(a) = a \wedge f^5(a) = a \wedge f(a) \neq a$ where $f^n(a) = \underbrace{f(f(\cdots(f(a))\cdots)}_{n}$

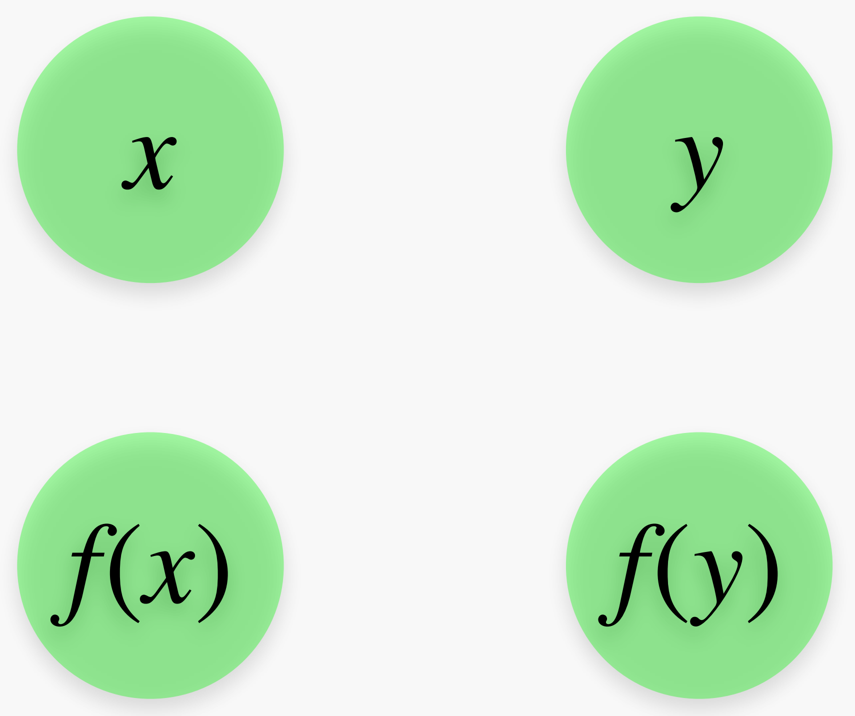
- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities
  - If F has a negative literal $t_1 \neq t_2$ with both terms in the same group, output UNSAT. Otherwise, output SAT

UNSAT



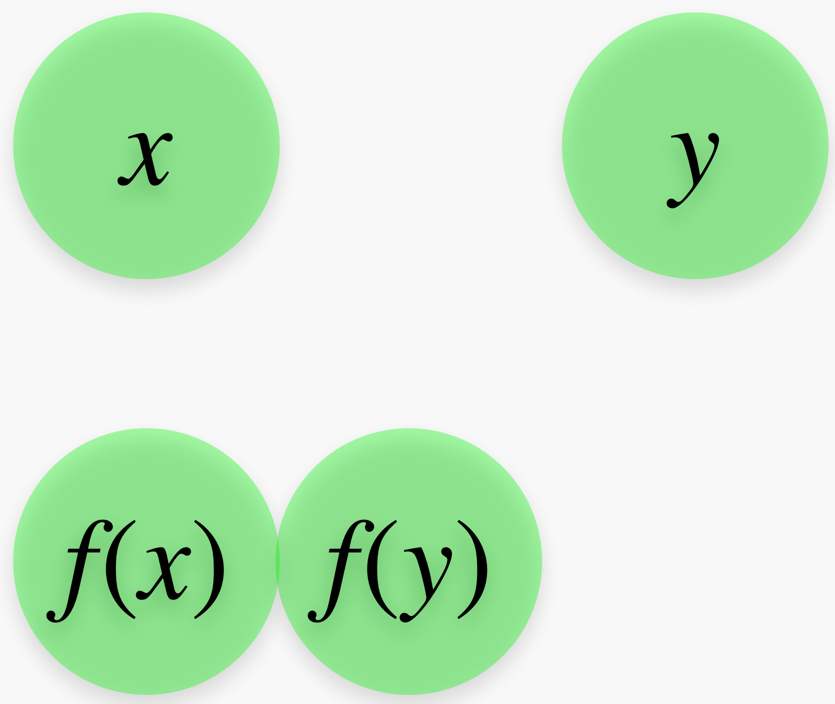
Congruence Closure Algorithm for T_E

- Consider $F : f(x) = f(y) \wedge x \neq y$
- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities
  - If F has a negative literal $t_1 \neq t_2$ with both terms in the same group, output UNSAT. Otherwise, output SAT



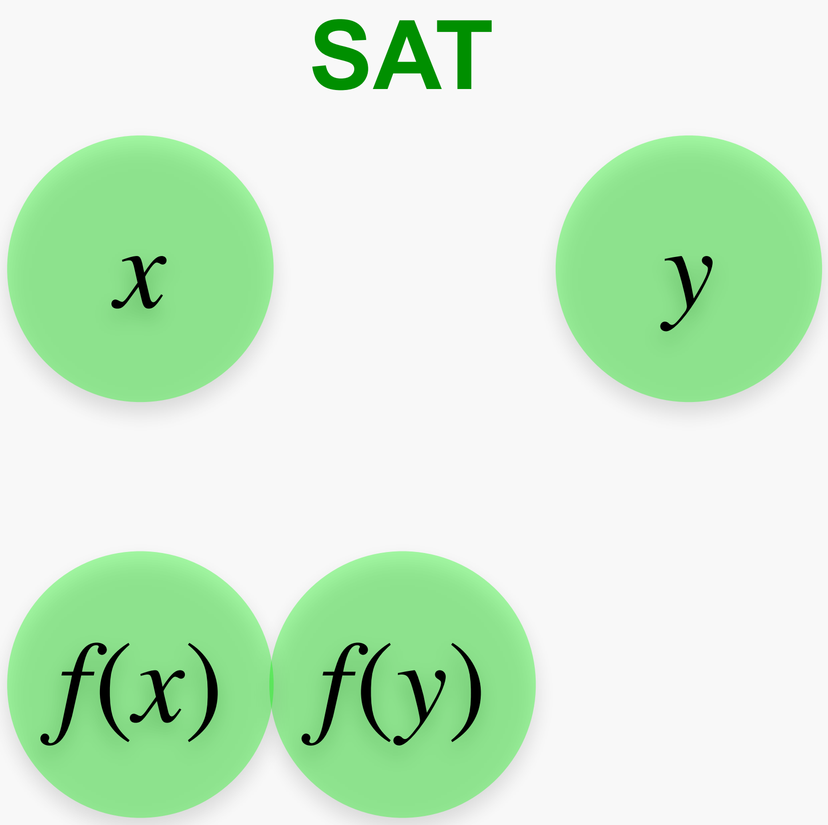
Congruence Closure Algorithm for T_E

- Consider $F : f(x) = f(y) \land x \neq y$
- For each positive literal $t_1 = t_2$ in F
  - Merge the groups for t_1 and t_2
  - Propagate the resulting equalities
  - If F has a negative literal $t_1 \neq t_2$ with both terms in the same group, output UNSAT. Otherwise, output SAT



Congruence Closure Algorithm for T_E

- Consider $F : f(x) = f(y) \wedge x \neq y$
- For each positive literal $t_1 = t_2$ in F
    - Merge the groups for t_1 and t_2
    - Propagate the resulting equalities
    - If F has a negative literal $t_1 \neq t_2$ with both terms in the same group, output UNSAT. Otherwise, output SAT



Congruence Closure Algorithm for T_E in General

- A binary relation (predicate) R is an equivalence relation if it's reflexive, symmetric, and transitive.
- An equivalence relation R is a congruence relation if for every n-ary function f,

$$\forall \bar{x}, \bar{y} \,.\, (\bigwedge_{i=1}^{n} x_i \; R \; y_i) \implies f(\bar{x}) \; R \; f(\bar{y})$$

- The equivalence closure of R over set S is the complete set of all equivalences. Suppose $S = \{a, b, c\}$ and $a \; R \; b, b \; R \; c$, then the equivalence closure is $\{aRb, bRa, aRa, bRb, bRc, cRb, aRc, cRa, cRc\}$

Quiz

- If $S = \{a, b, c, d\}$ and $a = b, b = c, d = d$, then what is the equivalence closure of = over S?

  {a=b, b=a, a=a, b=b, b=c, c=b, c=c, a=c, c=a, d=d}

Congruence Closure Algorithm for T_E in General

- The congruence closure of R over set S is the complete set of all congruence relations.
- The sub term set S_F of formula F is the set that contains all the sub terms of F.
  - The sub term set of $F : f(a, b) = a \wedge f(f(a, b), b) \neq a$ is $S_F = \{a, b, f(a, b), f(f(a, b), b)\}$.
  - The congruence closure of = over S_F is $\{f(a, b) = a, b = b . f(f(a, b), b) = f(a, b), \ldots\}$

Congruence Closure Algorithm for T_E in General

- Algorithm
  1. Given a formula

     $$F : s_1 = t_1 \wedge \ldots \wedge s_m = t_m \wedge s_{m+1} \neq t_{m+1} \wedge \ldots \wedge s_n \neq t_n$$

     construct the congruence closure of = of

     $$\{s_1 = t_1, \ldots, s_m = t_m\}$$

     over S_F.
  2. If $s_i = t_i$ according to the closure for any $i \in \{m + 1, \ldots, n\}$, return UNSAT.
  3. Otherwise, return SAT.

Review: Theory of Rationals

- The theory of rationals $T_{\mathbb{R}}$ has signature $\Sigma_{\mathbb{Q}}$

$$\Sigma_{\mathbb{Q}} : \ \{0,\ 1,\ +,\ -,\ =,\ \geq\}$$

- Axioms $A_{\mathbb{Q}}$

1. $\forall x, y.\ x \geq y \ \wedge\ y \geq x \ \rightarrow\ x = y$ (antisymmetry)
2. $\forall x, y, z.\ x \geq y \ \wedge\ y \geq z \ \rightarrow\ x \geq z$ (transitivity)
3. $\forall x, y.\ x \geq y \ \vee\ y \geq x$ (totality)
4. $\forall x, y, z.\ (x + y) + z = x + (y + z)$ (+ associativity)
5. $\forall x.\ x + 0 = x$ (+ identity)
6. $\forall x.\ x + (-x) = 0$ (+ inverse)
7. $\forall x, y.\ x + y = y + x$ (+ commutativity)
8. $\forall x, y, z.\ x \geq y \ \rightarrow\ x + z \geq y + z$ (+ ordered)

$\ldots$

Linear Programming

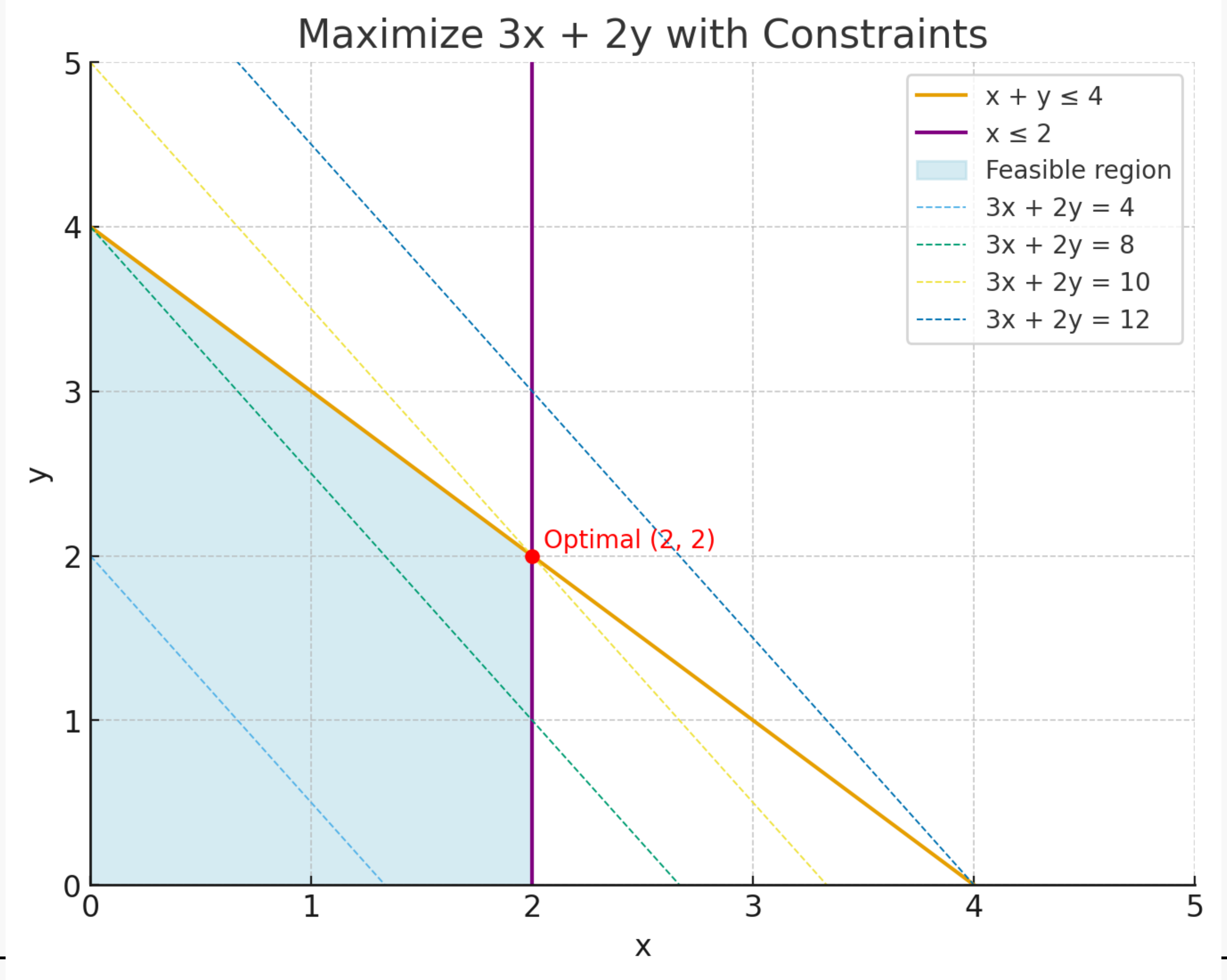
- Linear Programming: we want to find a solution for $x_1, \ldots, x_n$ maximizing objective function $c_1x_1 + \ldots + c_nx_n$ subject to linear inequality constraints

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le c_1 \land$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le c_2 \land$$
$$\ldots$$
$$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n \le c_n$$

- Very important problem: production management, finance, transportation, scheduling, …

Linear Programming

- Maximize $3x + 2y$ subject to $x + y \leq 4 \wedge x \leq 2 \wedge x, y \geq 0$



Deciding $T_{\mathbb{Q}}$ as Linear Programming

- Suppose we have a quantifier-free conjunctive $T_{\mathbb{Q}}$ formula F

$$F:\neg(x \geq 4) \wedge -x \geq -2 \wedge x \geq 0$$

- Rewrite each atomic formula into one only with "$\leq$" and ">0"
  - $\neg(x \geq 4)$ ➞ $x < 4$ ➞ $x + y \leq 4 \wedge y > 0$
  - $-x \geq -2$ ➞ $x \leq 2$
  - $x \geq 0$ ➞ $-x \leq 0$
- And obtain $x + y \leq 4 \wedge y > 0 \wedge x \leq 2 \wedge -x \leq 0$

Deciding $T_{\mathbb{Q}}$ as Linear Programming

- Solve a linear programming problem

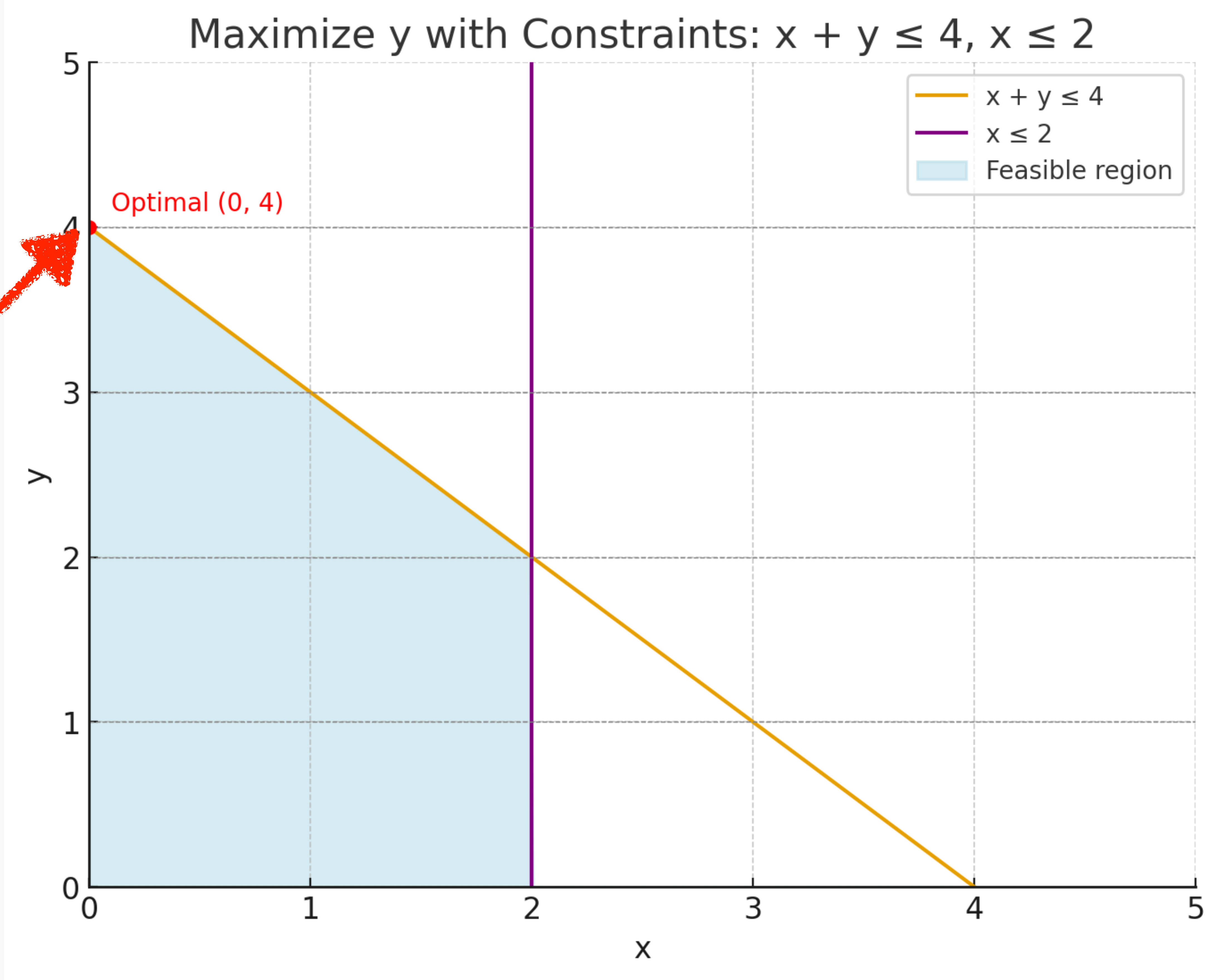
  maximize y

  subject to

  $$x + y \leq 4 \land x \leq 2 \land -x \leq 0$$

  y is maximized to 4 when $x = 0$

- The optimal solution is 4 which is > 0, therefore F is satisfiable.



Deciding $T_{\mathbb{Q}}$ as Linear Programming in General

- Consider a generic $T_{\mathbb{Q}}$ formula

$$
\begin{aligned}
F: \quad & \bigwedge_{i=1}^{m} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i \\
\wedge \quad & \bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \cdots + \alpha_{in}x_n < \beta_i
\end{aligned}
$$

Equalities can be written as two inequalities (e.g., $x = 0 \rightarrow x \geq 0 \wedge x \leq 0$).

Deciding $T_{\mathbb{Q}}$ as Linear Programming in General

- F is equivalent to

$$
\begin{aligned}
F' : \quad & \bigwedge_{i=1}^{m} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i \\
\wedge \quad & \bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \cdots + \alpha_{in}x_n + x_{n+1} \leq \beta_i \\
\wedge \quad & x_{n+1} > 0
\end{aligned}
$$

where x_{n+1} a fresh new variable.

Deciding $T_{\mathbb{Q}}$ as Linear Programming in General

- Deciding satisfiability of F is to solve the following linear programming problem

$$\textbf{max} \quad x_{n+1}$$
$$\textbf{subject to}$$

F':

$$\bigwedge_{i=1}^{m} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i$$
$$\bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \cdots + \alpha_{in}x_n + x_{n+1} \leq \beta_i$$

- If the optimum is positive (i.e., max of $x_{n+1} > 0$) , F is satisfiable.

Deciding $T_{\mathbb{Q}}$ as Linear Programming in General

- Suppose we have a quantifier-free conjunctive $T_{\mathbb{Q}}$ formula of the form:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \bowtie c_1 \wedge$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \bowtie c_2 \wedge$$
$$\ldots$$

  where $a_{i1}, \ldots, a_{in}, c_i$ are constants and $\bowtie \in \{=, \geq\}$.

- First, convert $T_{\mathbb{Q}}$ formula to NNF.
- In this form, every atomic formula is of the form:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \bowtie' c$$

  where $\bowtie' \in \{=, \neq, \geq, <\}$ (why?)

Deciding $T_{\mathbb{Q}}$ as Linear Programming in General

- Second, rewrite it as the one only with $\leq$ and > 0
  - $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geq c_i \rightarrow -a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{in}x_n + c_i \leq 0$
  - $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n < c_i \rightarrow a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n + y \leq c_i \wedge y > 0$
  - $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = c_i \rightarrow$

    $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq c_i \wedge -a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{in}x_n + c_i \leq 0$
  - $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \neq c_i \rightarrow$

    (transformation of $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n < c_i$) $\vee$

    (transformation of $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n > c_i$)

Summary

- Congruence closure algorithm for theory of equality
- Linear programming for theory of rationals