形式化方法导引

第 4 章 逻辑问题求解

4.2 理论 - (2) SMT 求解

黄文超

http://staff.ustc.edu.cn/~huangwc/fm.html

回顾: 定义: SAT 问题

SAT is the *decision* problem: given a propositional formula, is it *satisfiable*?

回顾: 定义: SMT problem

Extension of SAT, to deal with *numbers* and *inequalities*.

回顾: SMT 在软件测试中的一个重要应用: 符号执行 (*Symbolic Execution*) 问题: How to explore different program paths and for each path to

- *generate* a set of concrete *input* values exercising that path
- *check* for the presence of *various kinds* of errors

本节讲解内容: The Simplex method (单纯形法).

- dealing with *linear inequalities*

回顾: 定义: SAT 问题

SAT is the *decision* problem: given a propositional formula, is it *satisfiable*?

回顾: 定义: SMT problem

Extension of SAT, to deal with *numbers* and *inequalities*.

回顾: SMT 在软件测试中的一个重要应用: 符号执行 (*Symbolic Execution*) 问题: How to explore different program paths and for each path to

- *generate* a set of concrete *input* values exercising that path
- *check* for the presence of *various kinds* of errors

本节讲解内容: The Simplex method (单纯形法).

- dealing with *linear inequalities*

回顾: 定义: SAT 问题

SAT is the *decision* problem: given a propositional formula, is it *satisfiable*?

回顾: 定义: SMT problem

Extension of SAT, to deal with *numbers* and *inequalities*.

回顾: SMT 在软件测试中的一个重要应用: 符号执行 (*Symbolic Execution*) 问题: How to explore different program paths and for each path to

- *generate* a set of concrete *input* values exercising that path
- *check* for the presence of *various kinds* of errors

本节讲解内容: The Simplex method (单纯形法).

- dealing with *linear inequalities*

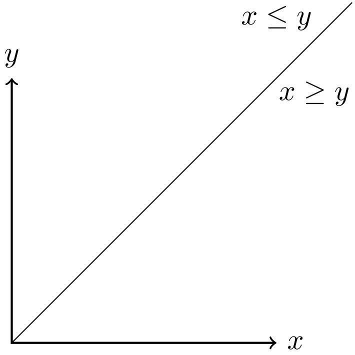
The Simplex method (单纯形法).

- dealing with *linear inequalities*

Do real numbers x, y exist such that

$$x \geq y$$
$$y \geq 2$$
$$2x + y \leq 7$$

So indeed the *blue* part describes the values x, y satisfying the requirements.



The Simplex method (单纯形法).

- dealing with *linear inequalities*

Do real numbers x, y exist such that

$$x \ge y$$

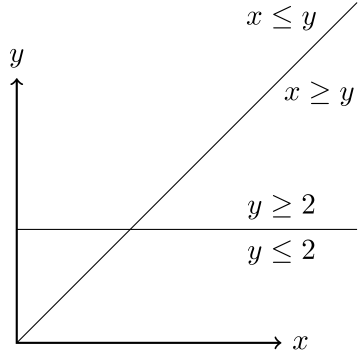
$$y \ge 2$$

$$2x + y \le 7$$

So indeed the *blue* part describes the values x, y satisfying the requirements.

2. 理论

2.2 Solve SMT | 问题分析



The Simplex method (单纯形法).

- dealing with *linear inequalities*

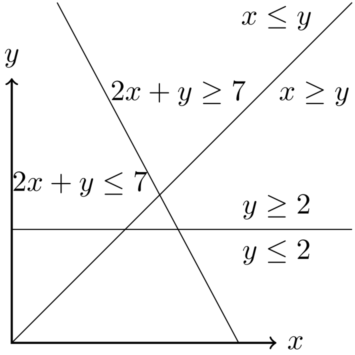
Do real numbers x, y exist such that

$$x \geq y$$

$$y \geq 2$$

$$2x + y \leq 7$$

So indeed the *blue* part describes the values x, y satisfying the requirements.



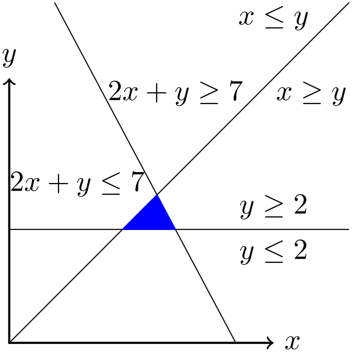
The Simplex method (单纯形法).

- dealing with *linear inequalities*

Do real numbers x, y exist such that

$$x \geq y$$
$$y \geq 2$$
$$2x + y \leq 7$$

So indeed the *blue* part describes the values x, y satisfying the requirements.



The Simplex method (单纯形法).

- dealing with *linear inequalities*

Do real numbers x, y exist such that

$$x \geq y$$
$$y \geq 2$$
$$2x + y \leq 7$$

So indeed the *blue* part describes the values x, y satisfying the requirements.

问题: How to solve for > 2 variables?

分析: *No such pictures*: we want to do this for *thousands* of inequalities /variables

For SMT the underlying approach is the *simplex method* for *linear optimization* = *linear programming* (线性规划)

- Not only determines existence of solution, but finds an *optimal*
  - one for which a given linear expression has the *highest possible* value

问题: How to solve for > 2 variables?

分析: *No such pictures*: we want to do this for *thousands* of inequalities /variables

For SMT the underlying approach is the *simplex method* for *linear optimization* = *linear programming* (线性规划)

- Not only determines existence of solution, but finds an *optimal*
  - one for which a given linear expression has the *highest possible* value

问题: How to solve for > 2 variables?

分析: *No such pictures*: we want to do this for *thousands* of inequalities /variables

For SMT the underlying approach is the *simplex method* for *linear optimization* = *linear programming* (线性规划)

- Not only determines existence of solution, but finds an *optimal*
  - one for which a given linear expression has the *highest possible* value

问题: How to solve for > 2 variables?

分析: *No such pictures*: we want to do this for *thousands* of inequalities /variables

For SMT the underlying approach is the *simplex method* for *linear optimization* = *linear programming* (线性规划)

- Not only determines existence of solution, but finds an *optimal*
  - one for which a given linear expression has the *highest possible* value

问题: How to solve for > 2 variables?

分析: *No such pictures*: we want to do this for *thousands* of inequalities /variables

For SMT the underlying approach is the *simplex method* for *linear optimization* = *linear programming* (线性规划)

- Not only determines existence of solution, but finds an *optimal*
  - one for which a given linear expression has the *highest possible* value

问题定义: The Simplex Method

Among all real values $x_1, \ldots, x_n \geq 0$ find the *maxmal* value of a linear *goal function*

$$v + c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

satisfying k linear constraints

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, 2, \ldots, k$. Here v, a_{ij}, c_i and b_i are given real values, satisfying $b_i \geq 0$ for $i = 1, 2, \ldots, k$

求解思路

Initialization: $x_i = 0$ for all i,
Do steps (*pivots*, 转轴):

- the value of the goal function *increases until* its optimal value

问题定义：The Simplex Method

Among all real values $x_1, \ldots, x_n \geq 0$ find the *maxmal* value of a linear *goal function*

$$v + c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

satisfying k linear constraints

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, 2, \ldots, k$. Here v, a_{ij}, c_i and b_i are given real values, satisfying $b_i \geq 0$ for $i = 1, 2, \ldots, k$

求解思路

Initialization: $x_i = 0$ for all i,

Do steps (*pivots*, 转轴):

- the value of the goal function *increases until* its optimal value

问题定义: The Simplex Method

Among all real values $x_1, \ldots, x_n \geq 0$ find the *maxmal* value of a linear *goal function*

$$v + c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

satisfying k linear constraints

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, 2, \ldots, k$. Here v, a_{ij}, c_i and b_i are given real values, satisfying $b_i \geq 0$ for $i = 1, 2, \ldots, k$

求解思路

Initialization: $x_i = 0$ for all i,
Do steps (*pivots*, 转轴):

- the value of the goal function *increases until* its optimal value

Applies to more general format:

- in an inequality $\geq$, multiply both sides by -1:

$$x_1 - 2x_2 + 3x_3 \geq -5 \equiv -x_1 + 2x_2 - 3x_3 \leq 5$$

- an equality $A = B$ is replaced by two inequalities $A \leq B$ and $B \leq A$
- if one wants to minimize, multiply goal function by -1
- if a variable x runs over all reals (positive and negative), replace it by $x_1 - x_2$ for fresh variables x_1, x_2 satisfying $x_i \geq 0$ for $i = 1, 2$

Later we will see how to deal if there is no trivial start solution (in case b_i<0).

Applies to more general format:

- in an inequality $\geq$, multiply both sides by -1:

$$x_1 - 2x_2 + 3x_3 \geq -5 \equiv -x_1 + 2x_2 - 3x_3 \leq 5$$

- an equality $A = B$ is replaced by two inequalities $A \leq B$ and $B \leq A$
- if one wants to minimize, multiply goal function by -1
- if a variable x runs over all reals (positive and negative), replace it by $x_1 - x_2$ for fresh variables x_1, x_2 satisfying $x_i \geq 0$ for $i = 1, 2$

Later we will see how to deal if there is no trivial start solution (in case b_i<0).

Applies to more general format:

- in an inequality $\geq$, multiply both sides by -1:

$$x_1 - 2x_2 + 3x_3 \geq -5 \equiv -x_1 + 2x_2 - 3x_3 \leq 5$$

- an equality $A = B$ is replaced by two inequalities $A \leq B$ and $B \leq A$
- if one wants to minimize, multiply goal function by -1
- if a variable x runs over all reals (positive and negative), replace it by $x_1 - x_2$ for fresh variables x_1, x_2 satisfying $x_i \geq 0$ for $i = 1, 2$

Later we will see how to deal if there is no trivial start solution (in case b_i<0).

Applies to more general format:

- in an inequality $\geq$, multiply both sides by -1:

$$x_1 - 2x_2 + 3x_3 \geq -5 \equiv -x_1 + 2x_2 - 3x_3 \leq 5$$

- an equality $A = B$ is replaced by two inequalities $A \leq B$ and $B \leq A$
- if one wants to minimize, multiply goal function by -1
- if a variable x runs over all reals (positive and negative), replace it by $x_1 - x_2$ for fresh variables x_1, x_2 satisfying $x_i \geq 0$ for $i = 1, 2$

Later we will see how to deal if there is no trivial start solution (in case b_i<0).

Applies to more general format:

- in an inequality $\geq$, multiply both sides by -1:

$$x_1 - 2x_2 + 3x_3 \geq -5 \equiv -x_1 + 2x_2 - 3x_3 \leq 5$$

- an equality $A = B$ is replaced by two inequalities $A \leq B$ and $B \leq A$
- if one wants to minimize, multiply goal function by -1
- if a variable x runs over all reals (positive and negative), replace it by $x_1 - x_2$ for fresh variables x_1, x_2 satisfying $x_i \geq 0$ for $i = 1, 2$

Later we will see how to deal if there is no trivial start solution (in case b_i<0).

Applies to more general format:

- in an inequality $\geq$, multiply both sides by -1:

$$x_1 - 2x_2 + 3x_3 \geq -5 \equiv -x_1 + 2x_2 - 3x_3 \leq 5$$

- an equality $A = B$ is replaced by two inequalities $A \leq B$ and $B \leq A$
- if one wants to minimize, multiply goal function by -1
- if a variable x runs over all reals (positive and negative), replace it by $x_1 - x_2$ for fresh variables x_1, x_2 satisfying $x_i \geq 0$ for $i = 1, 2$

Later we will see how to deal if there is no trivial start solution (in case b_i<0).

定义: Slack form 松弛型

For every linear inequality

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

a fresh variable $y_i \geq 0$ is introduced

The linear inequality is *replaced* by the equality

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

Together with $y_i \geq 0$, this is equivalent to the original inequality

This format with equalities is called the *slack form*

定义: Slack form 松弛型

For every linear inequality

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

a fresh variable $y_i \geq 0$ is introduced

The linear inequality is *replaced* by the equality

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

Together with $y_i \geq 0$, this is equivalent to the original inequality

This format with equalities is called the *slack form*

定义: Slack form 松弛型

For every linear inequality

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

a fresh variable $y_i \geq 0$ is introduced

The linear inequality is *replaced* by the equality

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

Together with $y_i \geq 0$, this is equivalent to the original inequality

This format with equalities is called the *slack form*

定义: basic/non-basic variables, basic solution

Some terminology on a slack form with equations

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

for $i = 1, 2, \ldots, k$

- The solution $y_i = b_i$, for $i = 1, 2, \ldots, k$ and $x_j = 0$ for $j = 1, 2, \ldots, k$ is called the *basic solution*
- The variables y_i for $i = 1, 2, \ldots, k$ are called *basic variables* (基变量)
- The variables x_j for $j = 1, 2, \ldots, k$ are called *non-basic variables* (非基变量)

算法思路: The simplex algorithm consists of a *repetition* of pivots

- A *pivot chooses* a *basic* variable and a non-basic variable, *swaps* their roles, and makes a *new slack form* that is *equivalent* to the original one, but with a *higher value* for the goal function.

定义: basic/non-basic variables, basic solution

Some terminology on a slack form with equations

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

for $i = 1, 2, \ldots, k$

- The solution $y_i = b_i$, for $i = 1, 2, \ldots, k$ and $x_j = 0$ for $j = 1, 2, \ldots, k$ is called the *basic solution*
- The variables y_i for $i = 1, 2, \ldots, k$ are called *basic variables* (基变量)
- The variables x_j for $j = 1, 2, \ldots, k$ are called *non-basic variables* (非基变量)

算法思路: The simplex algorithm consists of a *repetition* of pivots

- A *pivot chooses* a *basic* variable and a non-basic variable, *swaps* their roles, and makes a *new slack form* that is *equivalent* to the original one, but with a *higher value* for the goal function.

定义: basic/non-basic variables, basic solution

Some terminology on a slack form with equations

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

for $i = 1, 2, \ldots, k$

- The solution $y_i = b_i$, for $i = 1, 2, \ldots, k$ and $x_j = 0$ for $j = 1, 2, \ldots, k$ is called the *basic solution*
- The variables y_i for $i = 1, 2, \ldots, k$ are called *basic variables* (基变量)
- The variables x_j for $j = 1, 2, \ldots, k$ are called *non-basic variables* (非基变量)

算法思路: The simplex algorithm consists of a *repetition* of pivots

- A *pivot chooses* a *basic* variable and a non-basic variable, *swaps* their roles, and makes a *new slack form* that is *equivalent* to the original one, but with a *higher value* for the goal function.

定义: basic/non-basic variables, basic solution

Some terminology on a slack form with equations

$$y_i = b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n$$

for $i = 1, 2, \ldots, k$

- The solution $y_i = b_i$, for $i = 1, 2, \ldots, k$ and $x_j = 0$ for $j = 1, 2, \ldots, k$ is called the *basic solution*
- The variables y_i for $i = 1, 2, \ldots, k$ are called *basic variables* (基变量)
- The variables x_j for $j = 1, 2, \ldots, k$ are called *non-basic variables* (非基变量)

算法思路: The simplex algorithm consists of a *repetition* of pivots

- A *pivot chooses* a *basic* variable and a non-basic variable, *swaps* their roles, and makes a *new slack form* that is *equivalent* to the original one, but with a *higher value* for the goal function.

回顾：问题定义：The Simplex Method

Among all real values $x_1, \ldots, x_n \geq 0$ find the *maxmal* value of a linear *goal function*

$$v + c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

satisfying k linear constraints

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, 2, \ldots, k$.

例：

Maximize $z = 3 + x_1 + x_3$ satisfying the constraints:

$$\begin{aligned} -x_1 + x_2 - x_3 &\leq 2 \\ x_1 + x_3 &\leq 3 \\ 2x_1 - x_2 &\leq 4 \end{aligned}$$

回顾：问题定义：The Simplex Method

Among all real values $x_1, \ldots, x_n \geq 0$ find the *maxmal* value of a linear *goal function*

$$v + c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

satisfying k linear constraints

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, 2, \ldots, k$.

例：

Maximize $z = 3 + x_1 + x_3$ satisfying the constraints:

$$\begin{aligned} -x_1 + x_2 - x_3 &\leq 2 \\ x_1 + x_3 &\leq 3 \\ 2x_1 - x_2 &\leq 4 \end{aligned}$$

例：

Maximize $z = 3 + x_1 + x_3$ satisfying the constraints:

$$\begin{aligned} -x_1 + x_2 - x_3 &\leq 2 \\ x_1 + x_3 &\leq 3 \\ 2x_1 - x_2 &\leq 4 \end{aligned}$$

步骤 0: Slack form 松弛型

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

例：

Maximize $z = 3 + x_1 + x_3$ satisfying the constraints:

$$\begin{aligned} -x_1 + x_2 - x_3 &\leq 2 \\ x_1 + x_3 &\leq 3 \\ 2x_1 - x_2 &\leq 4 \end{aligned}$$

步骤 0: Slack form 松弛型

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

步骤 0: Slack form 松弛型

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

步骤 1: basic solution

$$x_1 = x_2 = x_3 = 0$$

$$y_1 = 2,\ y_2 = 3,\ y_3 = 4$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

Observation: if we increase x_1, and x_2 and x_3 remain 0, then the goal function z will increase

步骤 0: Slack form 松弛型

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

步骤 1: basic solution

$$x_1 = x_2 = x_3 = 0$$

$$y_1 = 2,\ y_2 = 3,\ y_3 = 4$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

Observation: if we increase x_1, and x_2 and x_3 remain 0, then the goal function z will increase

步骤 0: Slack form 松弛型

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

步骤 1: basic solution

$$x_1 = x_2 = x_3 = 0$$

$$y_1 = 2,\ y_2 = 3,\ y_3 = 4$$

Goal: Maximize $z = 3 + x_1 + x_3$, while keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

Observation: if we increase x_1, and x_2 and x_3 remain 0, then the goal function z will increase

分析: We want to *increase* x_1 as much as possible, keeping $x_2 = x_3 = 0$, while in the equations

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

keeping $\forall i. x_i \geq 0$ and $\forall i. y_i \geq 0$

- $y_1 = 2 + x_1 \geq 0$: OK if x_1 increases
- $y_2 = 3 - x_1 \geq 0$: *only* OK if $x_1 \leq 3$
- $y_3 = 4 - 2x_1 \geq 0$: *only* OK if $x_1 \leq 2$

So $y_i \geq 0$ only holds for all i if $x_1 \leq 2$

The highest allowed value for x_1 is 2, and then y_3 will get the value 0

分析: We want to *increase* x_1 as much as possible, keeping $x_2 = x_3 = 0$, while in the equations

$$
\begin{aligned}
y_1 &= 2 + x_1 - x_2 + x_3 \\
y_2 &= 3 - x_1 - x_3 \\
y_3 &= 4 - 2x_1 + x_2
\end{aligned}
$$

keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

- $y_1 = 2 + x_1 \geq 0$: OK if x_1 increases
- $y_2 = 3 - x_1 \geq 0$: *only* OK if $x_1 \leq 3$
- $y_3 = 4 - 2x_1 \geq 0$: *only* OK if $x_1 \leq 2$

So $y_i \geq 0$ only holds for all i if $x_1 \leq 2$

The highest allowed value for x_1 is 2, and then y_3 will get the value 0

分析: We want to *increase* x_1 as much as possible, keeping $x_2 = x_3 = 0$, while in the equations

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

keeping $\forall i. x_i \geq 0$ and $\forall i. y_i \geq 0$

- $y_1 = 2 + x_1 \geq 0$: OK if x_1 increases
- $y_2 = 3 - x_1 \geq 0$: *only* OK if $x_1 \leq 3$
- $y_3 = 4 - 2x_1 \geq 0$: *only* OK if $x_1 \leq 2$

So $y_i \geq 0$ only holds for all i if $x_1 \leq 2$

The highest allowed value for x_1 is 2, and then y_3 will get the value 0

分析: We want to *increase* x_1 as much as possible, keeping $x_2 = x_3 = 0$, while in the equations

$$
\begin{aligned}
y_1 &= 2 + x_1 - x_2 + x_3 \\
y_2 &= 3 - x_1 - x_3 \\
y_3 &= 4 - 2x_1 + x_2
\end{aligned}
$$

keeping $\forall i. x_i \geq 0$ and $\forall i. y_i \geq 0$

- $y_1 = 2 + x_1 \geq 0$: OK if x_1 increases
- $y_2 = 3 - x_1 \geq 0$: *only* OK if $x_1 \leq 3$
- $y_3 = 4 - 2x_1 \geq 0$: *only* OK if $x_1 \leq 2$

So $y_i \geq 0$ only holds for all i if $x_1 \leq 2$

The highest allowed value for x_1 is 2, and then y_3 will get the value 0

分析: We want to *increase* x_1 as much as possible, keeping $x_2 = x_3 = 0$, while in the equations

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

keeping $\forall i. x_i \geq 0$ and $\forall i. y_i \geq 0$

- $y_1 = 2 + x_1 \geq 0$: OK if x_1 increases
- $y_2 = 3 - x_1 \geq 0$: *only* OK if $x_1 \leq 3$
- $y_3 = 4 - 2x_1 \geq 0$: *only* OK if $x_1 \leq 2$

So $y_i \geq 0$ only holds for all i if $x_1 \leq 2$

The highest allowed value for x_1 is 2, and then y_3 will get the value 0

分析: We want to *increase* x_1 as much as possible, keeping $x_2 = x_3 = 0$, while in the equations

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

keeping $\forall i.x_i \geq 0$ and $\forall i.y_i \geq 0$

- $y_1 = 2 + x_1 \geq 0$: OK if x_1 increases
- $y_2 = 3 - x_1 \geq 0$: *only* OK if $x_1 \leq 3$
- $y_3 = 4 - 2x_1 \geq 0$: *only* OK if $x_1 \leq 2$

So $y_i \geq 0$ only holds for all i if $x_1 \leq 2$

The highest allowed value for x_1 is 2, and then y_3 will get the value 0

Highest allowed value for x_1 is 2, then y_3 will be 0

步骤 2: *pivot*: swap x_1 and y_3

Recall that

x_1 is *non-basic*: right from '$=$',$=$0 in basic solution

y_3 is *basic*: left from '$=$', possibly $\geq$ 0 in basic solution

By the *pivot*

- the *non-basic* variable x_1 will become *basic*
- the *basic* variable y_3 will become *non-basic*

问题: How to implement *pivot*? (见下页)

Highest allowed value for x_1 is 2, then y_3 will be 0

步骤 2: *pivot*: swap x_1 and y_3

Recall that

x_1 is *non-basic*: right from '$=$',$=$0 in basic solution

y_3 is *basic*: left from '$=$', possibly $\geq$ 0 in basic solution

By the *pivot*

- the *non-basic* variable x_1 will become *basic*
- the *basic* variable y_3 will become *non-basic*

问题: How to implement *pivot*? (见下页)

Highest allowed value for x_1 is 2, then y_3 will be 0

步骤 2: *pivot*: swap x_1 and y_3

Recall that

x_1 is *non-basic*: right from '=',=0 in basic solution

y_3 is *basic*: left from '=', possibly ≥ 0 in basic solution

By the *pivot*

- the *non-basic* variable x_1 will become *basic*
- the *basic* variable y_3 will become *non-basic*

问题: How to implement *pivot*? (见下页)

Highest allowed value for x_1 is 2, then y_3 will be 0

步骤 2: *pivot*: swap x_1 and y_3

Recall that

x_1 is *non-basic*: right from '$=$',$=0$ in basic solution

y_3 is *basic*: left from '$=$', possibly ≥ 0 in basic solution

By the *pivot*

- the *non-basic* variable x_1 will become *basic*
- the *basic* variable y_3 will become *non-basic*

问题: How to implement *pivot*? (见下页)

Highest allowed value for x_1 is 2, then y_3 will be 0

步骤 2: *pivot*: swap x_1 and y_3

Recall that

x_1 is *non-basic*: right from '=',=0 in basic solution

y_3 is *basic*: left from '=', possibly $\geq$ 0 in basic solution

By the *pivot*

- the *non-basic* variable x_1 will become *basic*
- the *basic* variable y_3 will become *non-basic*

问题: How to implement *pivot*? (见下页)

Highest allowed value for x_1 is 2, then y_3 will be 0

步骤 2: *pivot*: swap x_1 and y_3

Recall that

x_1 is *non-basic*: right from '$=$',$=$0 in basic solution

y_3 is *basic*: left from '$=$', possibly $\geq$ 0 in basic solution

By the *pivot*

- the *non-basic* variable x_1 will become *basic*
- the *basic* variable y_3 will become *non-basic*

问题: How to implement *pivot*? (见下页)

pivot: swap x_1 and y_3 in

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

In the equation $y_3 = 4 - 2x_1 + x_2$ move x_1 to the left and y_3 to the right:

$$x_1 = 2 + \frac{1}{2}x_2 - \frac{1}{2}y_3$$

Next replace every x_1 by $2 + \frac{1}{2}x_2 - \frac{1}{2}y_3$ in the equations for z, y_1, y_2, yielding the following *slack form*:

maximize $z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$

$$\begin{aligned} x_1 &= 2 + \frac{1}{2}x_2 - \frac{1}{2}y_3 \\ y_1 &= 4 - \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3 \\ y_2 &= 1 - \frac{1}{2}x_2 - x_3 + \frac{1}{2}y_3 \end{aligned}$$

pivot: swap x_1 and y_3 in

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

In the equation $y_3 = 4 - 2x_1 + x_2$ move x_1 to the left and y_3 to the right:

$$x_1 = 2 + \frac{1}{2}x_2 - \frac{1}{2}y_3$$

Next replace every x_1 by $2 + \frac{1}{2}x_2 - \frac{1}{2}y_3$ in the equations for z, y_1, y_2, yielding the following *slack form*:
maximize $z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$

$$\begin{aligned} x_1 &= 2 + \frac{1}{2}x_2 - \frac{1}{2}y_3 \\ y_1 &= 4 - \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3 \\ y_2 &= 1 - \frac{1}{2}x_2 - x_3 + \frac{1}{2}y_3 \end{aligned}$$

pivot: swap x_1 and y_3 in

$$\begin{aligned} y_1 &= 2 + x_1 - x_2 + x_3 \\ y_2 &= 3 - x_1 - x_3 \\ y_3 &= 4 - 2x_1 + x_2 \end{aligned}$$

In the equation $y_3 = 4 - 2x_1 + x_2$ move x_1 to the left and y_3 to the right:

$$x_1 = 2 + \frac{1}{2}x_2 - \frac{1}{2}y_3$$

Next replace every x_1 by $2 + \frac{1}{2}x_2 - \frac{1}{2}y_3$ in the equations for z, y_1, y_2, yielding the following *slack form*:

maximize $z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$

$$\begin{aligned} x_1 &= 2 + \frac{1}{2}x_2 - \frac{1}{2}y_3 \\ y_1 &= 4 - \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3 \\ y_2 &= 1 - \frac{1}{2}x_2 - x_3 + \frac{1}{2}y_3 \end{aligned}$$

This is the *end* of the *first pivot*

Observe

- all equations are replaced by *equivalent* equations, so the optimization problem is *equivalent* to the original one
- Now the *basic variables* are x_1, y_1, y_2 and the *non-basic variables* are x_2, x_3, y_3
- By construction again, we have a *slack form* with a *basic solution* in which *all non-basic variables are 0*, and *all basic variables are* ≥ 0
- In this new basic solution the goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

has *value 5*, *improving* the original *value 3*

This is the *end* of the *first pivot*
Observe

- all equations are replaced by *equivalent* equations, so the optimization problem is *equivalent* to the original one
- Now the *basic variables* are x_1, y_1, y_2 and the *non-basic variables* are x_2, x_3, y_3
- By construction again, we have a *slack form* with a *basic solution* in which *all non-basic variables are 0*, and *all basic variables are* ≥ 0
- In this new basic solution the goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

has *value 5*, *improving* the original *value 3*

This is the *end* of the *first pivot*
Observe

- all equations are replaced by *equivalent* equations, so the optimization problem is *equivalent* to the original one
- Now the *basic variables* are x_1, y_1, y_2 and the *non-basic variables* are x_2, x_3, y_3
- By construction again, we have a *slack form* with a *basic solution* in which *all non-basic variables are 0*, and *all basic variables are* ≥ 0
- In this new basic solution the goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

has *value 5*, *improving* the original *value 3*

This is the *end* of the *first pivot*
Observe

- all equations are replaced by *equivalent* equations, so the optimization problem is *equivalent* to the original one
- Now the *basic variables* are x_1, y_1, y_2 and the *non-basic variables* are x_2, x_3, y_3
- By construction again, we have a *slack form* with a *basic solution* in which *all non-basic variables are 0*, and *all basic variables are* ≥ 0
- In this new basic solution the goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

has *value 5*, *improving* the original *value 3*

This is the *end* of the *first pivot*

Observe

- all equations are replaced by *equivalent* equations, so the optimization problem is *equivalent* to the original one
- Now the *basic variables* are x_1, y_1, y_2 and the *non-basic variables* are x_2, x_3, y_3
- By construction again, we have a *slack form* with a *basic solution* in which *all non-basic variables are 0*, and *all basic variables are* ≥ 0
- In this new basic solution the goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

has *value 5*, *improving* the original *value 3*

In this goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

the non-basic variable x_2 has a *positive* factor $\frac{1}{2}$: increasing x_2 is the goal of the *next pivot*, while the other non-basic variables remain 0

$$x_1 = 2 + \tfrac{1}{2}x_2 - \tfrac{1}{2}y_3$$
$$y_1 = 4 - \tfrac{1}{2}x_2 + x_3 - \tfrac{1}{2}y_3$$
$$y_2 = 1 - \tfrac{1}{2}x_2 - x_3 + \tfrac{1}{2}y_3$$

$x_1 = 2 + \frac{1}{2}x_2 \geq 0$. OK if x_2 increases
$y_1 = 4 - \frac{1}{2}x_2 \geq 0$. only OK if $x_2 \leq 8$
$y_2 = 1 - \frac{1}{2}x_2 \geq 0$, only OK if $x_2 \leq 2$
So the maximal allowed value for x_2 is 2, in which case y_2 will get the value 0 $\Longrightarrow$ *pivot swapping* x_2 *and* y_2

In this goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

the non-basic variable x_2 has a *positive* factor $\frac{1}{2}$: increasing x_2 is the goal of the *next pivot*, while the other non-basic variables remain 0

$$\begin{aligned} x_1 &= 2 + \tfrac{1}{2}x_2 - \tfrac{1}{2}y_3 \\ y_1 &= 4 - \tfrac{1}{2}x_2 + x_3 - \tfrac{1}{2}y_3 \\ y_2 &= 1 - \tfrac{1}{2}x_2 - x_3 + \tfrac{1}{2}y_3 \end{aligned}$$

$x_1 = 2 + \frac{1}{2}x_2 \geq 0$. OK if x_2 increases
$y_1 = 4 - \frac{1}{2}x_2 \geq 0$. only OK if $x_2 \leq 8$
$y_2 = 1 - \frac{1}{2}x_2 \geq 0$, only OK if $x_2 \leq 2$
So the maximal allowed value for x_2 is 2, in which case y_2 will get the value 0 $\Longrightarrow$ *pivot swapping x_2 and y_2*

In this goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

the non-basic variable x_2 has a *positive* factor $\frac{1}{2}$: increasing x_2 is the goal of the *next pivot*, while the other non-basic variables remain 0

$$\begin{aligned} x_1 &= 2 + \tfrac{1}{2}x_2 - \tfrac{1}{2}y_3 \\ y_1 &= 4 - \tfrac{1}{2}x_2 + x_3 - \tfrac{1}{2}y_3 \\ y_2 &= 1 - \tfrac{1}{2}x_2 - x_3 + \tfrac{1}{2}y_3 \end{aligned}$$

$x_1 = 2 + \frac{1}{2}x_2 \geq 0$. OK if x_2 increases
$y_1 = 4 - \frac{1}{2}x_2 \geq 0$. only OK if $x_2 \leq 8$
$y_2 = 1 - \frac{1}{2}x_2 \geq 0$, only OK if $x_2 \leq 2$
So the maximal allowed value for x_2 is 2, in which case y_2 will get the value 0 $\Longrightarrow$ *pivot swapping x_2 and y_2*

In this goal function

$$z = 5 + \frac{1}{2}x_2 + x_3 - \frac{1}{2}y_3$$

the non-basic variable x_2 has a *positive* factor $\frac{1}{2}$: increasing x_2 is the goal of the *next pivot*, while the other non-basic variables remain 0

$$\begin{aligned} x_1 &= 2 + \tfrac{1}{2}x_2 - \tfrac{1}{2}y_3 \\ y_1 &= 4 - \tfrac{1}{2}x_2 + x_3 - \tfrac{1}{2}y_3 \\ y_2 &= 1 - \tfrac{1}{2}x_2 - x_3 + \tfrac{1}{2}y_3 \end{aligned}$$

$x_1 = 2 + \frac{1}{2}x_2 \geq 0$. OK if x_2 increases
$y_1 = 4 - \frac{1}{2}x_2 \geq 0$. only OK if $x_2 \leq 8$
$y_2 = 1 - \frac{1}{2}x_2 \geq 0$, only OK if $x_2 \leq 2$
So the maximal allowed value for x_2 is 2, in which case y_2 will get the value 0 $\Longrightarrow$ *pivot swapping* x_2 *and* y_2

步骤 3: *pivot swapping* x_2 *and* y_2

$$y_2 = 1 - \frac{1}{2}x_2 - x_3 + \frac{1}{2}y_3$$

yields $x_2 = 2 - 2x_3 - 2y_2 + y_3$, so in the goal function and in the other equations we replace x_2 by $2 - 2x_3 - 2y_2 + y_3$, yielding

Maximize $z = 6 - y_2$

satisfying

$$x_1 = 3 - x_3 - y_2$$
$$x_2 = 2 - 2x_3 - 2y_2 + y_3$$
$$y_1 = 3 + 2x_3 + y_2 - y_3$$

步骤 3: *pivot swapping* x_2 *and* y_2

$$y_2 = 1 - \frac{1}{2}x_2 - x_3 + \frac{1}{2}y_3$$

yields $x_2 = 2 - 2x_3 - 2y_2 + y_3$, so in the goal function and in the other equations we replace x_2 by $2 - 2x_3 - 2y_2 + y_3$, yielding
Maximize $z = 6 - y_2$
satisfying

$$x_1 = 3 - x_3 - y_2$$
$$x_2 = 2 - 2x_3 - 2y_2 + y_3$$
$$y_1 = 3 + 2x_3 + y_2 - y_3$$

Observation: since in

$$z = 6 - y_2$$

y_2 should be ≥ 0, the value z will always be ≤ 6

On the other hand, in the basic solution with

$$x_3 = y_2 = y_3 = 0$$

and

$$x_1 = 3, x_2 = 2, y_1 = 3$$

we obtain $z = 6$, so this basic solution yields the *maximal value* for z

Observation: since in

$$z = 6 - y_2$$

y_2 should be ≥ 0, the value z will always be ≤ 6

On the other hand, in the basic solution with

$$x_3 = y_2 = y_3 = 0$$

and

$$x_1 = 3, x_2 = 2, y_1 = 3$$

we obtain $z = 6$, so this basic solution yields the *maximal value* for z

算法: Simplex Method (单纯形法)

Solve a linear optimization problem in slack form maximize $z = v + \sum_{j=1}^{n} c_j x_j$ under a set of constraints of the shape

$$y_i = b_i + \sum_{j=1}^{n} a_{ij} x_j$$

with $b_i \geq 0$,for $i = 1, \ldots, k$

As long as there exists j such that $c_j > 0$ do a *pivot*, that is

- find the highest value for x_j for which $b_i + a_{ij} x_j \geq 0$ for all i, and $b_i + a_{ij} x_j = 0$ for one particular i
- swap x_j and y_i and bring the result in slack form

At the end $c_j \leq 0$ for all j, from which can be concluded that the basic solution of this last slack form yields the maximal value for z

算法: Simplex Method (单纯形法)

Solve a linear optimization problem in slack form maximize $z = v + \sum_{j=1}^{n} c_j x_j$ under a set of constraints of the shape

$$y_i = b_i + \sum_{j=1}^{n} a_{ij} x_j$$

with $b_i \geq 0$,for $i = 1, \ldots, k$
As long there exists j such that $c_j > 0$ do a *pivot*, that is

- find the highest value for x_j for which $b_i + a_{ij} x_j \geq 0$ for all i, and $b_i + a_{ij} x_j = 0$ for one particular i
- swap x_j and y_i and bring the result in slack form

At the end $c_j \leq 0$ for all j, from which can be concluded that the basic solution of this last slack form yields the maximal value for z

算法: Simplex Method (单纯形法)

Solve a linear optimization problem in slack form maximize $z = v + \sum_{j=1}^{n} c_j x_j$ under a set of constraints of the shape

$$y_i = b_i + \sum_{j=1}^{n} a_{ij} x_j$$

with $b_i \geq 0$,for $i = 1, \ldots, k$

As long there exists j such that $c_j > 0$ do a *pivot*, that is

- find the highest value for x_j for which $b_i + a_{ij} x_j \geq 0$ for all i, and $b_i + a_{ij} x_j = 0$ for one particular i
- swap x_j and y_i and bring the result in slack form

At the end $c_j \leq 0$ for all j, from which can be concluded that the basic solution of this last slack form yields the maximal value for z

算法: Simplex Method (单纯形法)

Solve a linear optimization problem in slack form maximize $z = v + \sum_{j=1}^{n} c_j x_j$ under a set of constraints of the shape

$$y_i = b_i + \sum_{j=1}^{n} a_{ij} x_j$$

with $b_i \geq 0$,for $i = 1, \ldots, k$

As long there exists j such that $c_j > 0$ do a *pivot*, that is

- find the highest value for x_j for which $b_i + a_{ij} x_j \geq 0$ for all i, and $b_i + a_{ij} x_j = 0$ for one particular i
- swap x_j and y_i and bring the result in slack form

At the end $c_j \leq 0$ for all j, from which can be concluded that the basic solution of this last slack form yields the maximal value for z

算法: Simplex Method (单纯形法)

Solve a linear optimization problem in slack form maximize $z = v + \sum_{j=1}^{n} c_j x_j$ under a set of constraints of the shape

$$y_i = b_i + \sum_{j=1}^{n} a_{ij} x_j$$

with $b_i \geq 0$,for $i = 1, \ldots, k$
As long there exists j such that $c_j > 0$ do a *pivot*, that is

- find the highest value for x_j for which $b_i + a_{ij} x_j \geq 0$ for all i, and $b_i + a_{ij} x_j = 0$ for one particular i
- swap x_j and y_i and bring the result in slack form

At the end $c_j \leq 0$ for all j, from which can be concluded that the basic solution of this last slack form yields the maximal value for z

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
  - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution*
However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
    - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution*
However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
  - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution* However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
  - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution* However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
  - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution* However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
  - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution*
However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
    - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution*
However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

General remarks

- Optimization problems may be *unbounded*; in this mechanism this will be encountered if no equation yields an upper bound on x_j
- A pivot only requires complexity O(kn)
- If $c_j > 0$ for more than one value of j, then the procedure is non-deterministic
- Always choose smallest j with $c_j > 0$: then repetition of pivots will terminate
- Worst case: number of pivots may be *exponential*.
  - In practice the simplex method is very efficient

新问题：Until now: *only* find *optimal value* when starting by *basic solution* However, for SMT the situation is *opposite*: *no interest* in *optimal solution*, only in *existence* of *a solution*

定义: feasible

A set of constraints is called *feasible* if it admits a solution.

问题: how to apply the *simplex method* presented so far

- to determine *feasibility* of *any* given set of inequalities

问题定义: feasibility

For a set of linear inequalities

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, \ldots, k$, and $x_j \geq 0$ for $j = 1, \ldots, n$.
Note: (not all $b_i \geq 0$)

定义: feasible

A set of constraints is called *feasible* if it admits a solution.

问题: how to apply the *simplex method* presented so far

- to determine *feasibility* of *any* given set of inequalities

问题定义: feasibility

For a set of linear inequalities

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, \ldots, k$, and $x_j \geq 0$ for $j = 1, \ldots, n$.
Note: (not all $b_i \geq 0$)

定义: feasible

A set of constraints is called *feasible* if it admits a solution.

问题: how to apply the *simplex method* presented so far

- to determine *feasibility* of *any* given set of inequalities

问题定义: feasibility

For a set of linear inequalities

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, \ldots, k$, and $x_j \geq 0$ for $j = 1, \ldots, n$.

Note: (not all $b_i \geq 0$)

定义：feasible

A set of constraints is called *feasible* if it admits a solution.

问题：how to apply the *simplex method* presented so far

- to determine *feasibility* of *any* given set of inequalities

问题定义：feasibility

For a set of linear inequalities

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, \ldots, k$, and $x_j \geq 0$ for $j = 1, \ldots, n$.
Note: (not all $b_i \geq 0$)

问题定义：feasibility

For a set of linear inequalities

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, \ldots, k$, and $x_j \geq 0$ for $j = 1, \ldots, n$.
Note: (not all $b_i \geq 0$)

方法思路：introduce a *fresh variable* $z \geq 0$
Extend the set of inequalities to

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - z \leq b_i$$

for $i = 1, \ldots, k$

问题定义：feasibility

For a set of linear inequalities

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

for $i = 1, \ldots, k$, and $x_j \geq 0$ for $j = 1, \ldots, n$.
Note: (not all $b_i \geq 0$)

方法思路：introduce a *fresh variable* $z \geq 0$
Extend the set of inequalities to

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - z \leq b_i$$

for $i = 1, \ldots, k$

The extended problem

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - z \leq b_i$$

is *always* feasible, even if $b_i \leq 0$: choose z very *large*

定理

Original problem feasible $\Leftrightarrow$ maximal value of $-z$ is 0 in extended problem

证明:

($\Leftarrow$) If extended problem has solution with $-z = 0$, then it is solution of the original problem, so feasible

($\Rightarrow$) If original problem feasible, then extended one has solution with $-z = 0$

- Since $z \geq 0$, this is the maximal value for $-z$

The extended problem

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - z \leq b_i$$

is *always* feasible, even if $b_i \leq 0$: choose z very *large*

定理

Original problem feasible $\Leftrightarrow$ maximal value of $-z$ is 0 in extended problem

证明:

($\Leftarrow$) If extended problem has solution with $-z = 0$, then it is solution of the original problem, so feasible

($\Rightarrow$) If original problem feasible, then extended one has solution with $-z = 0$

- Since $z \geq 0$, this is the maximal value for $-z$

The extended problem

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - z \leq b_i$$

is *always* feasible, even if $b_i \leq 0$: choose z very *large*

定理

Original problem feasible $\Leftrightarrow$ maximal value of $-z$ is 0 in extended problem

证明:
($\Leftarrow$) If extended problem has solution with $-z = 0$, then it is solution of the original problem, so feasible
($\Rightarrow$) If original problem feasible, then extended one has solution with $-z = 0$

- Since $z \geq 0$, this is the maximal value for $-z$

定理

Original problem feasible ⇔ maximal value of $-z$ is 0 in extended problem

So now we want to *maximize* $-z$ in the extended problem by the *simplex method*

例: 问题: Check feasibility

Find values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

定理

Original problem feasible $\Leftrightarrow$ maximal value of $-z$ is 0 in extended problem

So now we want to *maximize* $-z$ in the extended problem by the *simplex method*

例: 问题: Check feasibility

Find values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

定理

Original problem feasible ⇔ maximal value of $-z$ is 0 in extended problem

So now we want to *maximize* $-z$ in the extended problem by the *simplex method*

例: 问题: Check feasibility

Find values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

定理

Original problem feasible $\Leftrightarrow$ maximal value of $-z$ is 0 in extended problem

So now we want to *maximize* $-z$ in the extended problem by the *simplex method*

例: 问题: Check feasibility

Find values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

例: 问题: Check feasibility

Fine values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

步骤 2: Slack form:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

问题分析: Since b_i may be negative, we may have *no* basic solution with all variables ≥ 0

解决技巧: As the first pivot swap, the non-basic variable z with the basic variable y_i for i for which b_i is the *most negative*

例: 问题: Check feasibility

Fine values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

步骤 2: Slack form:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

问题分析: Since b_i may be negative, we may have *no* basic solution with all variables ≥ 0

解决技巧: As the first pivot swap, the non-basic variable z with the basic variable y_i for i for which b_i is the *most negative*

例: 问题: Check feasibility

Fine values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

步骤 2: Slack form:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

问题分析: Since b_i may be negative, we may have *no* basic solution with all variables ≥ 0

解决技巧: As the first pivot swap, the non-basic variable z with the basic variable y_i for i for which b_i is the *most negative*

例: 问题: Check feasibility

Fine values $x, y \geq 0$ satisfying

$$-x - 3y \leq -12$$
$$x + y \leq 10$$
$$-x + y \leq -7$$

步骤 1:

Introduce z for maximizing $-z$:

$$-x - 3y - z \leq -12$$
$$x + y - z \leq 10$$
$$-x + y - z \leq -7$$

步骤 2: Slack form:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

问题分析: Since b_i may be negative, we may have *no* basic solution with all variables ≥ 0
解决技巧: As the first pivot swap, the non-basic variable z with the basic variable y_i for i for which b_i is the *most negative*

步骤 2:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

Most negative b_i is $b_1 = -12$, so do pivot on z and y_1

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

步骤 2:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

Indeed $x = y = z = 0$ *does not yield a basic solution*, since then $y_1 = -12$ and $y_3 = -7$ *do not satisfy* $y_i \geq 0$

Most negative b_i is $b_1 = -12$, so do pivot on z and y_1

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

步骤 2:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

Indeed $x = y = z = 0$ *does not yield a basic solution*, since then $y_1 = -12$ and $y_3 = -7$ *do not satisfy* $y_i \geq 0$

Most negative b_i is $b_1 = -12$, so do pivot on z and y_1

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

步骤 2:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

Indeed $x = y = z = 0$ *does not yield a basic solution*, since then $y_1 = -12$ and $y_3 = -7$ *do not satisfy* $y_i \geq 0$

Most negative b_i is $b_1 = -12$, so do pivot on z and y_1

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

步骤 2:

Maximize $-z$ satisfying

$$y_1 = -12 + x + 3y + z$$
$$y_2 = 10 - x - y + z$$
$$y_3 = -7 + x - y + z$$

Indeed now we have a *basic solution*:
$x = y = y_1 = 0,$
$z = 12, y_2 = 22, y_3 = 5,$
all ≥ 0

Most negative b_i is $b_1 = -12$, so do pivot on z and y_1

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

From now on: proceed by simplex algorithm as before

Maximize $-12 + x + 3y - y_1$, so swap x with z, y_2 or y_3

步骤 4: pivot: swap x with y_2

Maximize $-1 + y - \frac{1}{2}y_1 - \frac{1}{2}y_2$ satisfying

$$x = 11 - 2y + \frac{1}{2}y_1 - \frac{1}{2}y_2$$
$$z = 1 - y + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 5 - 4y + y_1$$

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

From now on: proceed by simplex algorithm as before

Maximize $-12 + x + 3y - y_1$, so swap x with z, y_2 or y_3

步骤 4: pivot: swap x with y_2

Maximize $-1 + y - \frac{1}{2}y_1 - \frac{1}{2}y_2$ satisfying

$$x = 11 - 2y + \frac{1}{2}y_1 - \frac{1}{2}y_2$$
$$z = 1 - y + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 5 - 4y + y_1$$

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$

$$y_2 = 22 - 2x - 4y + y_1$$

$$y_3 = 5 - 4y + y_1$$

From now on: proceed by simplex algorithm as before

Maximize $-12 + x + 3y - y_1$, so swap x with z, y_2 or y_3

步骤 4: pivot: swap x with y_2

Maximize $-1 + y - \frac{1}{2}y_1 - \frac{1}{2}y_2$ satisfying

$$x = 11 - 2y + \frac{1}{2}y_1 - \frac{1}{2}y_2$$

$$z = 1 - y + \frac{1}{2}y_1 + \frac{1}{2}y_2$$

$$y_3 = 5 - 4y + y_1$$

步骤 3: pivot z and y, replace every $z = 12 - x - 3y + y_1$

Maximize $-12 + x + 3y - y_1$ satisfying

$$z = 12 - x - 3y + y_1$$
$$y_2 = 22 - 2x - 4y + y_1$$
$$y_3 = 5 - 4y + y_1$$

From now on: proceed by simplex algorithm as before
Maximize $-12 + x + 3y - y_1$, so swap x with z, y_2 or y_3

步骤 4: pivot: swap x with y_2

Maximize $-1 + y - \frac{1}{2}y_1 - \frac{1}{2}y_2$ satisfying

$$x = 11 - 2y + \frac{1}{2}y_1 - \frac{1}{2}y_2$$
$$z = 1 - y + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 5 - 4y + y_1$$

步骤 4: pivot: swap x with y_2

Maximize $-1 + y - \frac{1}{2}y_1 - \frac{1}{2}y_2$ satisfying

$$x = 11 - 2y + \frac{1}{2}y_1 - \frac{1}{2}y_2$$
$$z = 1 - y + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 5 - 4y + y_1$$

步骤 5: pivot: swap y with z

Maximize $-z$ satisfying

$$x = 9 \cdots z \cdots y_1 \cdots y_2$$
$$y = 1 - z + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 1 \cdots z \cdots y_1 \cdots y_2$$

步骤 4: pivot: swap x with y_2

Maximize $-1 + y - \frac{1}{2}y_1 - \frac{1}{2}y_2$ satisfying

$$x = 11 - 2y + \frac{1}{2}y_1 - \frac{1}{2}y_2$$
$$z = 1 - y + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 5 - 4y + y_1$$

步骤 5: pivot: swap y with z

Maximize $-z$ satisfying

$$x = 9 \cdots z \cdots y_1 \cdots y_2$$
$$y = 1 - z + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 1 \cdots z \cdots y_1 \cdots y_2$$

步骤 5: pivot: swap y with z

Maximize $-z$ satisfying

$$x = 9 \cdots z \cdots y_1 \cdots y_2$$
$$y = 1 - z + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 1 \cdots z \cdots y_1 \cdots y_2$$

Surprise: we are back at our original maximization function $-z$
Since it has no positive factors anymore, the resulting basic solution

$$z = y_1 = y_2 = 0, x = 9, y = 1, y_3 = 1$$

yields the optimal value $-z = 0$
Since $-z = 0$, the original set of inequalities

$$-x - 3y \leq 12 \wedge x + y \leq 10 \wedge -x + y \leq -7$$

is satisfiable: we found the solution $x = 9, y = 1$

步骤 5: pivot: swap y with z

Maximize $-z$ satisfying

$$x = 9 \cdots z \cdots y_1 \cdots y_2$$
$$y = 1 - z + \frac{1}{2}y_1 + \frac{1}{2}y_2$$
$$y_3 = 1 \cdots z \cdots y_1 \cdots y_2$$

Surprise: we are back at our original maximization function $-z$
Since it has no positive factors anymore, the resulting basic solution

$$z = y_1 = y_2 = 0, x = 9, y = 1, y_3 = 1$$

yields the optimal value $-z = 0$
Since $-z = 0$, the original set of inequalities

$$-x - 3y \leq 12 \wedge x + y \leq 10 \wedge -x + y \leq -7$$

is satisfiable: we found the solution $x = 9, y = 1$

Concluding:

- We saw how the check on *feasibility* of a set of linear inequalities can be executed by *adding a fresh variable z* and maximize $-z$ by the original *simplex* approach

引申: *Linear programming* （线性规划问题）

- Given a set of linear inequalities on real valued variables and a linear goal function, determine whether this is feasible
- If so, find the maximal value of the goal function satisfying the inequalities

复杂度: *Worst case* the number of pivots is exponential, but *in practice* exponential blow-up is *rare*.

之后的问题: How to apply *simplex method* to SMT

Concluding:

- We saw how the check on *feasibility* of a set of linear inequalities can be executed by *adding a fresh variable* z and maximize $-z$ by the original *simplex* approach

引申: *Linear programming* （线性规划问题）

- Given a set of linear inequalities on real valued variables and a linear goal function, determine whether this is feasible
- If so, find the maximal value of the goal function satisfying the inequalities

复杂度: *Worst case* the number of pivots is exponential, but *in practice* exponential blow-up is *rare*.

之后的问题: How to apply *simplex method* to SMT

Concluding:

- We saw how the check on *feasibility* of a set of linear inequalities can be executed by *adding a fresh variable* z and maximize $-z$ by the original *simplex* approach

引申: *Linear programming* (线性规划问题)

- Given a set of linear inequalities on real valued variables and a linear goal function, determine whether this is feasible
- If so, find the maximal value of the goal function satisfying the inequalities

复杂度: *Worst case* the number of pivots is exponential, but *in practice* exponential blow-up is *rare*.

之后的问题: How to apply *simplex method* to SMT

Concluding:

- We saw how the check on *feasibility* of a set of linear inequalities can be executed by *adding a fresh variable* z and maximize $-z$ by the original *simplex* approach

引申: *Linear programming* （线性规划问题）

- Given a set of linear inequalities on real valued variables and a linear goal function, determine whether this is feasible
- If so, find the maximal value of the goal function satisfying the inequalities

复杂度: *Worst case* the number of pivots is exponential, but *in practice* exponential blow-up is *rare*.

之后的问题: How to apply *simplex method* to SMT

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT
定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT
定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT
定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT

定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT
定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT
定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT

定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

We saw how the *simplex method*

- applies for *optimizing a goal function* starting from a basic solution,
- also check *feasibility*: whether a set of linear inequalities has a solution
- applies for linear programming

Now: From *SAT* to SMT
定义与比较:

- a *CNF*: a conjunction of *clauses*
- a *clause*: a disjunction of *literals*
- a *literal*: an atom or a negation of an *atom*
- an atom:
  - For *SAT*: just a boolean *variable*
  - for *SMT*: an *expression* in the *corresponding theory*, in our case: *a linear inequality*

思路: extend CDCL to CNFs on linear inequalities.

Consider the CNF of the following three clauses

$$(1)\ x \geq y+1 \vee z \geq y+1$$
$$(2)\ y \geq z$$
$$(3)\ z \geq x+1$$

解:

$(y \geq z)$ **Unipropagate** on (2)

$(y \geq z)\ (x \geq y+1)$ **Unipropagate** on (1)

since $y \geq z \wedge z \geq y+1$ is unfeasible by Simplex

Fail on (3)

since $y \geq z \wedge x \geq y+1 \wedge z \geq x+1$ is unfeasible

This proves that the given CNF is unsatisfiable

Consider the CNF of the following three clauses

$$(1)\ x \geq y+1 \vee z \geq y+1$$
$$(2)\ y \geq z$$
$$(3)\ z \geq x+1$$

解:

$(y \geq z)$ **Unipropagate** on (2)

$(y \geq z)\ (x \geq y+1)$ **Unipropagate** on (1)

since $y \geq z \wedge z \geq y+1$ is unfeasible by Simplex

Fail on (3)

since $y \geq z \wedge x \geq y+1 \wedge z \geq x+1$ is unfeasible

This proves that the given CNF is unsatisfiable

Also for more complicated examples with **Decide** and **Backtrack**

This is how current SMT solvers (Z3, CVC4, Yices, ...) deal with linear inequalities

Theories:

- *Simplex* is called very often for combination of M and literals from clauses
- Other theories: combine CDCL with efficient method for checking whether a conjunction of literals is contradictory
- *Integer Linear programming (ILP), 整数线性规划* is harder (even NP-complete), but *effectively* supported by current SMT solvers

Also for more complicated examples with **Decide** and **Backtrack**

This is how current SMT solvers (Z3, CVC4, Yices, ...) deal with linear inequalities

Theories:

- *Simplex* is called very often for combination of M and literals from clauses
- Other theories: combine CDCL with efficient method for checking whether a conjunction of literals is contradictory
- *Integer Linear programming (ILP), 整数线性规划* is harder (even NP-complete), but *effectively* supported by current SMT solvers

Also for more complicated examples with **Decide** and **Backtrack**

This is how current SMT solvers (Z3, CVC4, Yices, ...) deal with linear inequalities

Theories:

- *Simplex* is called very often for combination of M and literals from clauses
- Other theories: combine CDCL with efficient method for checking whether a conjunction of literals is contradictory
- *Integer Linear programming (ILP), 整数线性规划* is harder (even NP-complete), but *effectively* supported by current SMT solvers

Also for more complicated examples with **Decide** and **Backtrack**

This is how current SMT solvers (Z3, CVC4, Yices, ...) deal with linear inequalities

Theories:

- *Simplex* is called very often for combination of M and literals from clauses
- Other theories: combine CDCL with efficient method for checking whether a conjunction of literals is contradictory
- *Integer Linear programming (ILP), 整数线性规划* is harder (even NP-complete), but *effectively* supported by current SMT solvers

Also for more complicated examples with **Decide** and **Backtrack**

This is how current SMT solvers (Z3, CVC4, Yices, ...) deal with linear inequalities

Theories:

- *Simplex* is called very often for combination of M and literals from clauses
- Other theories: combine CDCL with efficient method for checking whether a conjunction of literals is contradictory
- *Integer Linear programming (ILP), 整数线性规划* is harder (even NP-complete), but *effectively* supported by current SMT solvers

作业

使用 Simplex Method, 求解如下问题:

1. Maximize $z = 1 + 2x_1 + 3x_2 + 6x_3$ satisfying the constraints:

$$
\begin{aligned}
x_1 + 2x_2 + 3x_3 &\leq 10 \\
x_1 - x_3 &\leq 3 \\
-x_2 + 2x_3 &\leq 5
\end{aligned}
$$

where $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

2. Fine values $x, y \geq 0$ satisfying

$$
\begin{aligned}
x - y &\leq -3 \\
2x + y &\leq 7 \\
-x - 2y &\leq -9
\end{aligned}
$$