

# SAT and SMT Solving

**Sarah Winkler**

Computational Logic Group  
Department of Computer Science  
University of Innsbruck

lecture 8  
SS 2019

# Outline

- Summary of Last Week
- Cyclic Simplex Example
- Deciding Equality Logic
- Branch and Bound

## Definition (Theory of Linear Arithmetic over $C$ )

- ▶ for variables  $x_1, \dots, x_n$ , formulas built according to grammar

$$\varphi ::= \varphi \wedge \varphi \mid t = t \mid t < t \mid t \leq t$$

$$t ::= a_1x_1 + \dots + a_nx_n + b \quad \text{for } a_1, \dots, a_n, b \in \text{in carrier } C$$

- ▶ axioms are equality axioms plus calculation rules of arithmetic over  $C$
- ▶ solution assigns values in  $C$  to  $x_1, \dots, x_n$

## Definitions

- ▶ carrier  $\mathbb{R}$ : linear real arithmetic (LRA),  
DPLL( $T$ ) simplex algorithm is decision procedure
- ▶ carrier  $\mathbb{Z}$ : linear integer arithmetic (LIA)

## DPLL( $\mathcal{T}$ ) Simplex Algorithm (1)

- ▶ linear arithmetic constraint solving over real or rational variables
- ▶  $x_1, \dots, x_n$  are split into basic variables  $\vec{x}_B$  and nonbasic variables  $\vec{x}_N$

### Input

constraints plus upper and lower bounds for  $x_1, \dots, x_n$ :

$$A \vec{x}_N = \vec{x}_B \quad \text{with tableau } A \in \mathbb{R}^{|B| \times |N|} \quad (1)$$

$$-\infty \leq l_i \leq x_i \leq u_i \leq +\infty \quad (2)$$

### Output

satisfying assignment or “unsatisfiable”

### Invariant

(1) is satisfied and (2) holds for all nonbasic variables  $x_i$

## DPLL( $T$ ) Simplex Algorithm (2)

$$A\vec{x}_N = \vec{x}_B \quad (1)$$

$$-\infty \leq l_i \leq x_i \leq u_i \leq +\infty \quad (2)$$

### Method

- ▶ if (2) holds for all basic variables, return current assignment
- ▶ otherwise select basic variable  $x_i$  (so  $i \in B$ ) which violates (2)
- ▶ select **suitable** nonbasic variable  $x_j$  (so  $j \in N$ ) such that  $x_i$  and  $x_j$  can be swapped in a **pivoting** step, resulting in new tableau

$$A' x_{N'} = x_{B'}$$

with  $N' = N \cup \{i\} - \{j\}$  and  $B' = B \cup \{j\} - \{i\}$

- ▶ change value of  $x_j$  to  $l_j$  or  $u_j$  and update values of basic variables accordingly

## DPLL( $T$ ) Simplex Algorithm (3)

$$A\vec{x}_N = \vec{x}_B \quad (1)$$

$$-\infty \leq l_i \leq x_i \leq u_i \leq +\infty \quad (2)$$

### Pivoting

- ▶ swap basic  $x_i$  and non-basic  $x_j$

$$x_i = \sum_{k \in N} A_{ik} x_k \quad \Longrightarrow \quad x_j = \frac{1}{A_{ij}} \left( x_i - \sum_{k \in N - \{j\}} A_{ik} x_k \right) \quad (*)$$

- ▶ new tableau  $A'$  consists of (\*) and  $A_{B - \{i\}} \vec{x}_N = \vec{x}_{B - \{i\}}$  with (\*) substituted

### Update

- ▶ assignment of  $x_i$  is updated to previously violated bound  $l_i$  or  $u_i$ ,
- ▶ assignment of  $x_k$  is recomputed using (\*) and  $A'$  for all  $k \in B - \{i\} \cup \{j\}$

## DPLL( $T$ ) Simplex Algorithm (4)

$$A\vec{x}_N = \vec{x}_B \quad (1)$$

$$-\infty \leq l_i \leq x_i \leq u_i \leq +\infty \quad (2)$$

### Suitability

- ▶ basic variable  $x_i$  violates lower and/or upper bound
- ▶ pick nonbasic variable  $x_j$  such that
  - ▶ if  $x_i < l_i$ :  $A_{ij} > 0$  and  $x_j < u_j$  or  $A_{ij} < 0$  and  $x_j > l_j$
  - ▶ if  $x_i > u_i$ :  $A_{ij} > 0$  and  $x_j > l_j$  or  $A_{ij} < 0$  and  $x_j < u_j$
- ▶ problem is unsatisfiable if no suitable pivot exists

### Bland's Rule

- ▶ pick lexicographically smallest  $(i, j)$  that is suitable pivot
- ▶ guarantees termination

# Outline

- Summary of Last Week
- Cyclic Simplex Example
- Deciding Equality Logic
- Branch and Bound



## Example (due to B. Felgenhauer)

$$-1 \leq x_1 \leq 0$$

$$-4 \leq x_2 \leq 0$$

$$-5 \leq x_3 \leq -4$$

$$-7 \leq x_4 \leq 1$$

$$\begin{array}{c}
 x_3 \\
 x_4
 \end{array}
 \begin{pmatrix} x_1 & x_2 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline 0 \quad 0 \quad 0 \quad 0 \end{array}
 \quad \leftarrow \quad
 \begin{array}{c}
 x_3 \\
 x_2
 \end{array}
 \begin{pmatrix} x_1 & x_4 \\ -3 & 2 \\ -2 & 1 \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline 0 \quad 1 \quad 2 \quad 1 \end{array}$$

$$\downarrow$$

$$\begin{array}{c}
 x_3 \\
 x_4
 \end{array}
 \begin{pmatrix} x_3 & x_2 \\ 1 & -2 \\ 2 & -3 \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline -4 \quad 0 \quad -4 \quad -8 \end{array}
 \quad \begin{array}{c}
 x_3 \\
 x_2
 \end{array}
 \begin{pmatrix} x_3 & x_4 \\ -\frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline \frac{7}{3} \quad -\frac{11}{3} \quad -5 \quad 1 \end{array}$$

$$\downarrow$$

$$\begin{array}{c}
 x_3 \\
 x_2
 \end{array}
 \begin{pmatrix} x_3 & x_4 \\ -\frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline -\frac{10}{3} \quad -\frac{1}{3} \quad -4 \quad -7 \end{array}
 \quad \begin{array}{c}
 x_3 \\
 x_4
 \end{array}
 \begin{pmatrix} x_3 & x_2 \\ 1 & -2 \\ 2 & -3 \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline 3 \quad -4 \quad -5 \quad 2 \end{array}$$

$$\downarrow$$

$$\begin{array}{c}
 x_3 \\
 x_2
 \end{array}
 \begin{pmatrix} x_1 & x_4 \\ -3 & 2 \\ -2 & 1 \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline -1 \quad -5 \quad -11 \quad -7 \end{array}
 \quad \rightarrow \quad
 \begin{array}{c}
 x_3 \\
 x_4
 \end{array}
 \begin{pmatrix} x_1 & x_2 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}
 \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline -1 \quad -4 \quad -9 \quad -6 \end{array}$$

## Example (due to B. Felgenhauer)

$$-1 \leq x_1 \leq 0$$

$$-4 \leq x_2 \leq 0$$

$$-5 \leq x_3 \leq -4$$

$$-7 \leq x_4 \leq 1$$

$$\begin{array}{l} x_3 \\ x_4 \end{array} \begin{pmatrix} x_1 & x_2 \\ 1 & 2 \\ 2 & 1 \end{pmatrix} \quad \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline 0 \quad 0 \quad 0 \quad 0 \end{array}$$



$$\begin{array}{l} x_1 \\ x_4 \end{array} \begin{pmatrix} x_3 & x_2 \\ 1 & -2 \\ 2 & -3 \end{pmatrix} \quad \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline -4 \quad 0 \quad -4 \quad -8 \end{array}$$

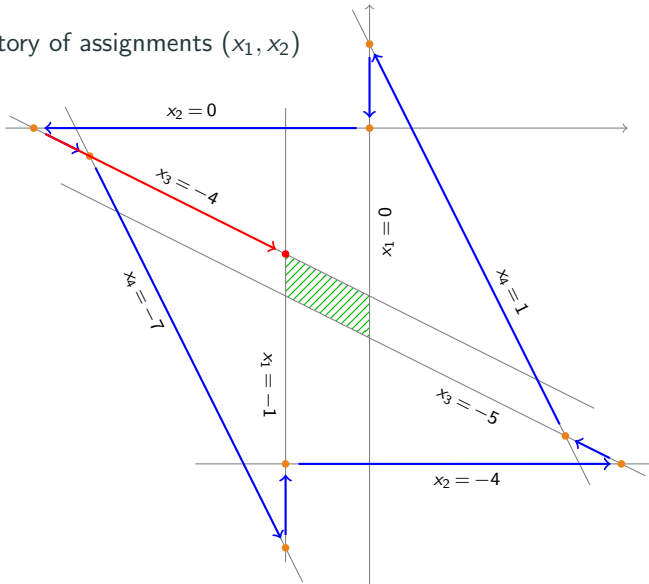
violation of Bland's rule



$$\begin{array}{l} x_2 \\ x_4 \end{array} \begin{pmatrix} x_3 & x_1 \\ \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{3}{2} \end{pmatrix} \quad \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \hline -1 \quad -\frac{3}{2} \quad -4 \quad -\frac{7}{2} \end{array}$$

satisfying assignment

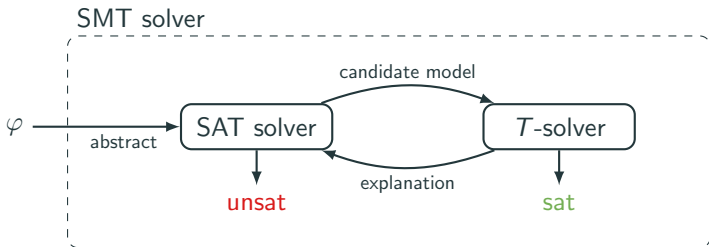
trajectory of assignments  $(x_1, x_2)$



# Outline

- Summary of Last Week
- Cyclic Simplex Example
- Deciding Equality Logic
- Branch and Bound

# How to Be Lazy



## Theory $T$

- ▶ equality logic
- ▶ equality + uninterpreted functions (EUF)
- ▶ linear real arithmetic (LRA)
- ▶ linear integer arithmetic (LIA)
- ▶ bitvectors (BV)
- ▶ arrays (A)

## $T$ -solving method

- equality graphs
- congruence closure ✓
- DPLL( $T$ ) Simplex ✓
- DPLL( $T$ ) Simplex + cuts

## Input to Satisfiability Problem for Equality Logic

conjunction  $\varphi$  of equality logic literals over set of variables  $V$

### Definitions

- ▶  $\varphi_ =$  is set of **positive literals** (equality literals) in  $\varphi$
- ▶  $\varphi_{\neq}$  is set of **negative literals** (inequality literals) in  $\varphi$
- ▶ **equality graph** is undirected graph  $G_=(\varphi) = (V, \varphi_ =, \varphi_{\neq})$

### Definitions

equality graph  $G_=(\varphi) = (V, \varphi_ =, \varphi_{\neq})$

- ▶ **contradictory cycle** is cycle with exactly one  $\varphi_{\neq}$  edge
- ▶ contradictory cycle is **simple** if it contains no node twice

### Lemma

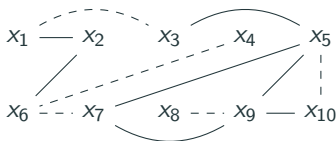
$\varphi$  is satisfiable iff  $G_=(\varphi)$  contains no simple contradictory cycles

## Example

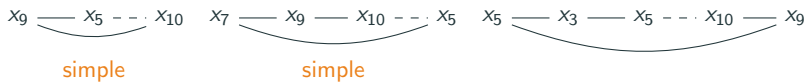
conjunction of equality literals

$$\varphi = x_1 = x_2 \wedge x_1 \neq x_3 \wedge x_3 = x_5 \wedge x_4 \neq x_6 \wedge x_6 \neq x_7 \wedge x_5 = x_9 \wedge x_5 = x_7 \wedge x_8 \neq x_9 \wedge x_9 = x_{10} \wedge x_7 = x_9 \wedge x_5 \neq x_{10}$$

- ▶  $\varphi =$ :  $x_1 = x_2, x_3 = x_5, x_5 = x_7, x_9 = x_{10}, x_7 = x_9, x_5 = x_9$
- ▶  $\varphi \neq$ :  $x_1 \neq x_3, x_4 \neq x_6, x_6 \neq x_7, x_8 \neq x_9, x_5 \neq x_{10}$
- ▶ equality graph



- ▶ contradictory cycles



- ▶ unsatisfiable

## Input to Satisfiability Problem for Equality Logic

conjunction  $\varphi$  of equality logic literals over set of variables  $V$

### Definitions

- ▶  $\varphi_ =$  is set of positive literals (equality literals) in  $\varphi$
- ▶  $\varphi_{\neq}$  is set of negative literals (inequality literals) in  $\varphi$
- ▶ equality graph is undirected graph  $G_=(\varphi) = (V, \varphi_ =, \varphi_{\neq})$

### Definitions

equality graph  $G_=(\varphi) = (V, \varphi_ =, \varphi_{\neq})$

- ▶ **contradictory cycle** is cycle with exactly one  $\varphi_{\neq}$  edge
- ▶ contradictory cycle is **simple** if it contains no node twice

### Lemma

$\varphi$  is satisfiable iff  $G_=(\varphi)$  contains no simple contradictory cycles

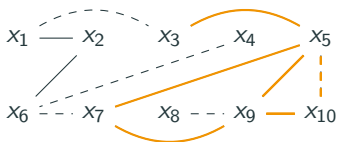


## Example

conjunction of equality literals

$$\varphi = x_1 = x_2 \wedge x_1 \neq x_3 \wedge x_3 = x_5 \wedge x_4 \neq x_6 \wedge x_6 \neq x_7 \wedge x_5 = x_9 \wedge x_5 = x_7 \wedge x_8 \neq x_9 \wedge x_9 = x_{10} \wedge x_7 = x_9 \wedge x_5 \neq x_{10}$$

- ▶  $\varphi =$ :  $x_1 = x_2, x_3 = x_5, x_5 = x_7, x_9 = x_{10}, x_7 = x_9, x_5 = x_9$
- ▶  $\varphi \neq$ :  $x_1 \neq x_3, x_4 \neq x_6, x_6 \neq x_7, x_8 \neq x_9, x_5 \neq x_{10}$
- ▶ equality graph



- ▶ contradictory cycles

$$x_9 \text{ --- } x_5 \text{ - - - } x_{10}$$

simple

$$x_7 \text{ --- } x_9 \text{ --- } x_{10} \text{ - - - } x_5$$

simple

$$x_5 \text{ --- } x_3 \text{ --- } x_5 \text{ - - - } x_{10} \text{ --- } x_9$$

not simple

- ▶ unsatisfiable

# Outline

- Summary of Last Week
- Cyclic Simplex Example
- Deciding Equality Logic
- Branch and Bound

## Example

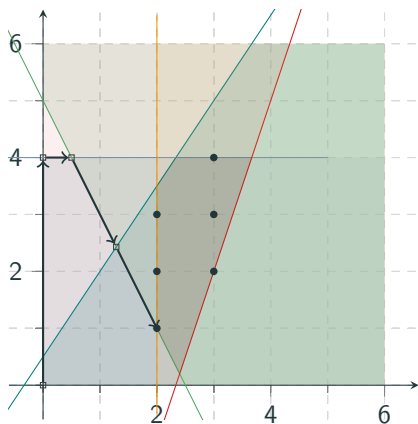
$$3x - 2y \geq -1$$

$$y \leq 4$$

$$2x + y \geq 5$$

$$3x - y \leq 7$$

- ▶ looking for solution in  $\mathbb{Z}^2$
- ▶ infinite  $\mathbb{R}^2$  solution space, six solutions in  $\mathbb{Z}^2$
- ▶ Simplex returns  $(\frac{9}{7}, \frac{17}{7})$



## Idea (Branch and Bound)

- ▶ add constraints that **exclude solution in  $\mathbb{R}^2$**  but **do not change solutions in  $\mathbb{Z}^2$**
- ▶ in current solution  $1 < x < 2$ , so use Simplex on two augmented problems:
  - ▶  $C \wedge x \leq 1$                     **unsatisfiable**
  - ▶  $C \wedge x \geq 2$                     **satisfiable, Simplex can return  $(2, 1)$**

---

**Algorithm** BranchAndBound( $\varphi$ )

---

**Input:** LIA constraint  $\varphi$

**Output:** unsatisfiable, or satisfying assignment

let  $res$  be result of deciding  $\varphi$  over  $\mathbb{R}$

▷ e.g. by Simplex

**if**  $res$  is unsatisfiable **then**

return unsatisfiable

**else if**  $res$  is solution over  $\mathbb{Z}$  **then**

return  $res$

**else**

let  $x$  be variable assigned non-integer value  $q$  in  $res$

$res = \text{BranchAndBound}(\varphi \wedge x \leq \lfloor q \rfloor)$

return  $res \neq \text{unsatisfiable} ? res : \text{BranchAndBound}(\varphi \wedge x \geq \lceil q \rceil)$

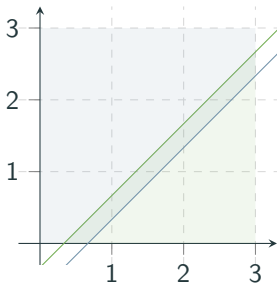
---

## Definition

$\mathbb{R}^2$ -solution space of linear arithmetic problem  $Ax \leq b$  is **bounded**

if for all  $x_i$  there exist  $l_i, u_i \in \mathbb{R}$  such that all  $\mathbb{R}^2$ -solutions  $v$  satisfy  $l_i \leq v(x_i) \leq u_i$

## Example



- ▶  $3x - 3y \geq 1 \wedge 3x - 3y \leq 2$
- ▶ unbounded problem
- ▶ no solution in  $\mathbb{Z}^2$
- ▶ BranchAndBound keeps adding  $x \geq n, y \geq m$

## Remarks

- ▶ BranchAndBound might **not terminate** if solution space is **unbounded**
- ▶ methods exist to derive **solution bounds** from tableau, but bounds are often too high for efficient practical procedures
- ▶ use **cutting planes** to restrict solution space more efficiently

# LIA Application: Finding Work Schedules

## Example (Scheduling Problem)

Is there a six-week cyclic work schedule for 22 employees who work 8-hour shifts, 24/7, 5 working days per week such that the following holds: In morning and afternoon shifts 6 employees are present, in night shifts 3. Joe does only morning shifts, Sally does not work on Sundays. Nobody works more than 6 days in a row.

## Shift Schedule Requirements

- ▶ number of employees  $n$
- ▶ set of shifts  $A$  (activities to be distributed)
- ▶ length of schedule (e.g. one week) and cyclicity
- ▶ requirement matrix  $R$ :  $R_{ij}$  is # employees required in shift  $i$  of day  $j$
- ▶ prohibited shift sequences, maximal length of work blocks, ...

## LIA Encoding

- ▶ integer variable corresponding to employee for each activity
- ▶ cardinality constraints for requirement matrix
- ▶ ...

# Bibliography



Bruno Dutertre and Leonardo de Moura.

## **A Fast Linear-Arithmetic Solver for DPLL(T).**

Proc. of International Conference on Computer Aided Verification, pp. 81–94, 2006.



Bruno Dutertre and Leonardo de Moura

## **Integrating Simplex with DPLL(T)**

Technical Report SRI-CSL-06-01, SRI International, 2006



Daniel Kroening and Ofer Strichman

## **The Simplex Algorithm**

Section 5.2 of Decision Procedures — An Algorithmic Point of View  
Springer, 2008



Bertram Felgenhauer and Aart Middeldorp

## **Constructing Cycles in the Simplex Method for DPLL(T)**

Proc. 14th International Colloquium on Theoretical Aspects of Computing,  
LNCS 10580, pp. 213–228, 2017



Christoph Erking and Nysret Musliu

## **Personnel Scheduling as Satisfiability Modulo Theories**

Proc. 26th International Joint Conference on Artificial Intelligence,  
pp. 614–621, 2017