Résolution efficace des modèles logiques

IA303

Alexandre Chapoutot

ENSTA ParisTech

2018-2019

Course Outline

Main goals of the course:

- To be able to model decision problems using logical formulas combined with other theories.
- Know the solving algorithms for SAT/SMT problems

Remarks:

- consider unquantified logical formula
- consider semantic methods based on exhaustivity

General information

- Teacher: Alexandre Chapoutot
- Mail: `alexandre.chapoutot@ensta-paristech.fr`
- Office: R216 at U2IS, ENSTA ParisTech, 828 bd des maréchaux, Palaiseau
- Class: Tuesday morning (9h-12h)
- Course grade: presentation of scientific papers
- Webpage:

  `http://perso.ensta-paristech.fr/~chapoutot/sat-smt/`

Syllabus

Nov. 13, 2018	Propositional logic
Nov. 27, 2018	SAT solver algorithms
Dec. 04, 2018	First-order theories and SMT solver algorithms
Dec. 11, 2018	Theory of uninterpreted functions
Dec. 18, 2018	Theory of linear real expressions
Jan. 08, 2019	Theory of nonlinear real expressions
Jan. 15, 2019	Combination of theories
Jan. 22, 2019	Exam

Theory of linear real expressions

Context

We consider decision procedure for linear arithmetic theory such as the one able to deal with

Example

$$
\begin{aligned}
x + y &\geqslant 2 \quad \wedge \\
2x - y &\geqslant 0 \quad \wedge \\
-x + 2y &\geqslant 1 \quad .
\end{aligned}
$$

Syntax of linear real arithmetic

A formula F of Linear Arithmetic (LA) theory follows the syntax:

$$\begin{aligned} F &::= F \wedge F \mid (F) \mid A \\ A &::= S \quad \text{op} \quad S \\ \text{op} &::= \; < \mid \leqslant \mid = \\ S &::= T \mid S + T \\ T &::= \text{id} \mid \text{cst} \mid \text{cst} \times \text{id} \end{aligned}$$

Depending of the set of values for constants different theories are considered:

- in $\mathbb{Z}$ Linear Integer Arithmetic (LIA) is considered
- in $\mathbb{Q}$ or $\mathbb{R}$ Linear Real Arithmetic (LRA) is considered

Decision procedure for LRA has exponential complexity while for LIA it is NP-complete.

Solving linear systems

Algorithm coming from Linear Programming (LP) which considers

$$\begin{aligned} \max \mathbf{cx} \\ \text{s.t.} \\ \mathbf{Ax} \leqslant \mathbf{b} \\ \mathbf{x} \geqslant 0 \end{aligned}$$

A classical approach to solve LP is to use *simplex algorithm*.

In context of decision procedure we consider *general simplex algorithm* dealing with problems of the form

$$\begin{cases} \mathbf{Ax} = \mathbf{0} \\ \ell \leqslant \mathbf{x} \leqslant \mathbf{u} \end{cases}$$

A is a $m \times n$ matrix (m constraint with n variables)

Putting linear systems into general form

We can transform arbitrary linear constraints of the form

$$L \bowtie R$$

with $\bowtie \in \{=, >, \geqslant\}$ such that

1. Move to the left all elements of R to get constraint of the form $L' \bowtie b$ where b is a constant.
2. Introduce a new variable s_i in order to have constraints

$$\begin{aligned} L' - s_i = 0 \quad & \wedge \\ s_i \bowtie b \quad & . \end{aligned}$$

is $\bowtie \; \equiv \; =$ then $s_i = b$ is writes as $s_i \geqslant b \wedge s_i \leqslant b$

Example

Consider

$$\begin{aligned} x + y &\geqslant 2 \quad \land \\ 2x - y &\geqslant 0 \quad \land \\ -x + 2y &\geqslant 1 \quad . \end{aligned}$$

which is put into the general form

$$\begin{aligned} x + y - s_1 &= 0 \quad \land \\ 2x - y - s_2 &= 0 \quad \land \\ -x + 2y - s_3 &= 0 \quad \land \\ s_1 &\geqslant 2 \quad \land \\ s_2 &\geqslant 0 \quad \land \\ s_3 &\geqslant 1 \quad . \end{aligned}$$

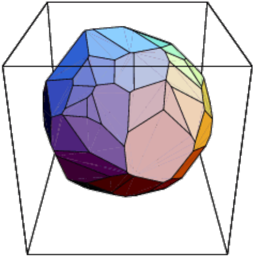
x and y are named **problem variables** while s_i are named **additional variables**.

Geometric view of the consistent sets

The set of values satisfying the linear constraints is named **admissible set**.

It is defined a convex polyhedron as a convex intersection of convex half-spaces.

Example[1]

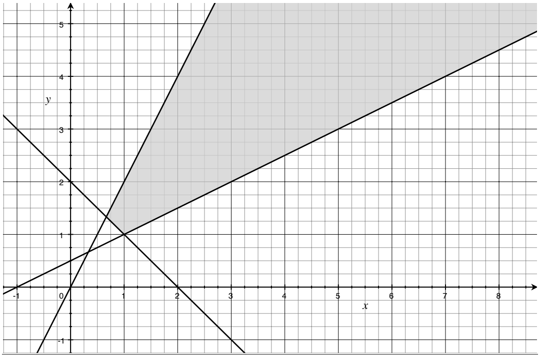


[1] http://mathworld.wolfram.com/ConvexPolyhedron.html

Example of admissible set

For

$$\begin{aligned} x + y &\geqslant 2 \quad \wedge \\ 2x - y &\geqslant 0 \quad \wedge \\ -x + 2y &\geqslant 1 \quad . \end{aligned}$$



General simplex algorithm

Recall a system in a general form is such that

$$\mathbf{A}\mathbf{x} = \mathbf{0}$$
$$\forall i \in \{1, \ldots, n\}, \quad \ell_i \leqslant s_i \leqslant u_i$$

Note that, due to additional variables, matrix A is of dimension $m \times (n + m)$

$$\begin{aligned} x + y - s_1 = 0 &\quad \wedge \\ 2x - y - s_2 = 0 &\quad \wedge \\ -x + 2y - s_3 = 0 &\quad \wedge \\ s_1 \geqslant 2 &\quad \wedge \\ s_2 \geqslant 0 &\quad \wedge \\ s_3 \geqslant 1 &\quad . \end{aligned}$$

$$\mathbf{A} = \begin{matrix} x & y & s_1 & s_2 & s_3 \end{matrix} \\ \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 2 & -1 & 0 & -1 & 0 \\ -1 & 2 & 0 & 0 & -1 \end{pmatrix}$$

The tableau

- The diagonal sub-matrix comes from the general form of the problem

$$\mathbf{A} = \begin{array}{c} \\ \end{array}\begin{pmatrix} \overset{x}{1} & \overset{y}{1} & \overset{s_1}{-1} & \overset{s_2}{0} & \overset{s_3}{0} \\ 2 & -1 & 0 & -1 & 0 \\ -1 & 2 & 0 & 0 & -1 \end{pmatrix}$$

- Instead we can only write, what is named the **tableau**

$$\begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \begin{pmatrix} \overset{x}{1} & \overset{y}{1} \\ 2 & -1 \\ -1 & 2 \end{pmatrix}$$

  - x and y are the **non-basic variables**
  - s_1, s_2, s_3 are the **basic variables**

The general simplex algorithm will transform this tableau but keeping the dimension $m \times n$.

The tableau – 2

- Denote by
  - $\mathcal{B}$ the set of basic variables
  - $\mathcal{N}$ the set of non-basic variables
- The tableau is representation of the linear system such that

$$\bigwedge_{x_i \in \mathcal{B}} \left(x_i = \sum_{x_j \in \mathcal{N}} a_{ij} x_j \right)$$

Remark: basic variables are also named **dependent variables**

The general simplex algorithm

- Simplex algorithm maintains
  - the tableau
  - an assignment α to all variables
  - the bounds of s_i
- Initially, one has
  - $\mathcal{B}$ = additional variables
  - $\mathcal{N}$ = problem variables
  - $\alpha(x_i) = 0$ for all $i \in \{1, \ldots, n+m\}$

The general simplex algorithm – invariant properties

Two invariant are maintained during the execution

- $\mathbf{Ax} = \mathbf{0}$
- All non-basic variables satisfy their bounds

At the beginning, the initial assignment satisfies $\mathbf{Ax} = \mathbf{0}$

- If the bounds of all basic variables are satisfied by α then return "Satisfiable"
- Otherwise, the algorithm updates α with an operation named **pivoting**.

Pivoting

- Find a basic variable x_i that violates its bounds
  - Suppose that $\alpha(x_i) > u_i$
- Find a non-basic variable x_j such that
  - $a_{ij} > 0$ and $\alpha(x_j) < u_j$, (increase the lower bound of x_i) or
  - $a_{ij} < 0$ and $\alpha(x_j) > \ell_j$ (decrease the upper bound of x_i)
- Such variable x_j is named **suitable**
- If there is no suitable variable then return "Unsatisfiable"

Definitions

Given two variables x_i and x_j, the coefficient a_{ij} is called the **pivot element**. The column of x_j is called the **pivot column** while the row of x_i is the **pivot row**.

Pivoting x_i and x_j – transformation of the tableau

The pivoting is performed in two steps assuming that $a_{ij} \neq 0$:

- Solve equation i for x_j, from

$$x_i = a_{ij}x_j + \sum_{k \neq j} a_{ik}x_k$$

to

$$x_j = \frac{x_i}{a_{ij}} - \sum_{k \neq j} \frac{a_{ik}}{a_{ij}} x_k$$

- Swap x_i and x_j and update the i-th row accordingly, from

a_{i1}	$\cdots$	a_{ij}	$\cdots$	a_{in}

to

$\frac{-a_{i1}}{a_{ij}}$	$\cdots$	$\frac{1}{a_{ij}}$	$\cdots$	$\frac{-a_{in}}{a_{ij}}$

Pivoting x_i and x_j – transformation of the tableau

- Update all other rows $k \neq i$ by replacing x_j by its equivalent obtained from row i, *i.e.*,

$$x_j = \frac{x_i}{a_{ij}} - \sum_{k \neq j} \frac{a_{ik}}{a_{ij}} x_k$$

Remark: note that pivot operation is the same operation as in the **Gaussian variable elimination** procedure.

Pivoting x_i and x_j – update of α

- Increase (or decrease) $\alpha(x_j)$ by $\theta = \frac{u_i - \alpha(x_i)}{a_{ij}}$
- $\alpha(x_i)$ is set to the upper (or lower) bound of x_i
- update all basic (dependent) variables accordingly

In consequence, x_j is basic variable so it may violate its bounds so repeat the general simplex algorithm.

Example

Recall, the tableau with bounds

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 2 \end{pmatrix} \end{array} \qquad \begin{array}{ll} s_1 \geqslant 2 & \wedge \\ s_2 \geqslant 0 & \wedge \\ s_3 \geqslant 1 & . \end{array} \qquad \begin{array}{r} \alpha(x) = 0 \\ \alpha(y) = 0 \\ \alpha(s_1) = 0 \\ \alpha(s_2) = 0 \\ \alpha(s_3) = 0 \end{array}$$

Initially, α assigns 0 to all variables so bounds for s_1 and s_3 are violated.

Example

Recall, the tableau with bounds

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 2 \end{pmatrix} \end{array} \qquad \begin{array}{ll} s_1 \geqslant 2 & \wedge \\ s_2 \geqslant 0 & \wedge \\ s_3 \geqslant 1 & . \end{array} \qquad \begin{array}{r} \alpha(x) = 0 \\ \alpha(y) = 0 \\ \alpha(s_1) = 0 \\ \alpha(s_2) = 0 \\ \alpha(s_3) = 0 \end{array}$$

- We will solve s_1
  - x is then a suitable (non-basic) variable for pivoting (has no upper bound)
- So pivoting s_1 with x

Example

Recall, the tableau with bounds

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 2 \end{pmatrix} \end{array}$$

$$\begin{aligned} s_1 &\geqslant 2 \quad \wedge \\ s_2 &\geqslant 0 \quad \wedge \\ s_3 &\geqslant 1 \quad . \end{aligned}$$

$$\begin{aligned} \alpha(x) &= 0 \\ \alpha(y) &= 0 \\ \alpha(s_1) &= 0 \\ \alpha(s_2) &= 0 \\ \alpha(s_3) &= 0 \end{aligned}$$

- Solve 1-st row for x: $s_1 = x + y \Leftrightarrow x = s_1 - y$
- Replace x with s_1 in other rows

$$\begin{aligned} s_2 &= 2(s_1 - y) - y \Leftrightarrow s_2 = 2s_1 - 3y \\ s_3 &= -(s_1 - y) + 2y \Leftrightarrow s_3 = -s_1 + 3y \end{aligned}$$

Example

New tableau with bounds

$$\begin{array}{c} \\ x \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & y \end{array} \\ \begin{pmatrix} 1 & -1 \\ 2 & -3 \\ -1 & 3 \end{pmatrix} \end{array}$$

$$\begin{array}{ll} s_1 \geqslant 2 & \wedge \\ s_2 \geqslant 0 & \wedge \\ s_3 \geqslant 1 & . \end{array}$$

$$\begin{aligned} \alpha(x) &= 2 \\ \alpha(y) &= 0 \\ \alpha(s_1) &= 2 \\ \alpha(s_2) &= 4 \\ \alpha(s_3) &= -2 \end{aligned}$$

Update of the assignment

- We have to increase x by $\theta = \frac{2-0}{1} = 2$ so $\alpha(x) = 0 + 2 = 2$
- Now s_1 is equal to its lower bound so $\alpha(s_1) = 2$
- update all other basic variables accordingly

Example

New tableau with bounds

$$\begin{array}{c} \\ x \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & y \end{array} \\ \begin{pmatrix} 1 & -1 \\ 2 & -3 \\ -1 & 3 \end{pmatrix} \end{array}$$

$$\begin{array}{lll} s_1 \geqslant 2 & \wedge \\ s_2 \geqslant 0 & \wedge \\ s_3 \geqslant 1 & . \end{array}$$

$$\begin{aligned} \alpha(x) &= 2 \\ \alpha(y) &= 0 \\ \alpha(s_1) &= 2 \\ \alpha(s_2) &= 4 \\ \alpha(s_3) &= -2 \end{aligned}$$

Now s_3 still violates its lower bound

- y is a suitable variable for pivoting.
- y should be increase by $\theta = \frac{1-(-2)}{3} = 1$

Example

Final tableau with bounds

$$
\begin{array}{c} \\ x \\ s_2 \\ y \end{array}
\begin{array}{c}
\begin{array}{cc} s_1 & s_3 \end{array} \\
\begin{pmatrix} 2/3 & -1/3 \\ 1 & -1 \\ 1/3 & 1/3 \end{pmatrix}
\end{array}
\qquad
\begin{array}{l}
s_1 \geqslant 2 \quad \wedge \\
s_2 \geqslant 0 \quad \wedge \\
s_3 \geqslant 1 \quad .
\end{array}
\qquad
\begin{array}{r}
\alpha(x) = 1 \\
\alpha(y) = 1 \\
\alpha(s_1) = 2 \\
\alpha(s_2) = 1 \\
\alpha(s_3) = 1
\end{array}
$$

All the constraints are satisfied!
Solution is $x \mapsto 1$ and $y \mapsto 1$

Pseudo code of the general simplex algorithm[2]

Algorithm 5.2.1: GENERAL-SIMPLEX

Input: A linear system of constraints S
Output: "Satisfiable" if the system is satisfiable, "Unsatisfiable" otherwise

1. Transform the system into the general form

$$A\mathbf{x} = 0 \quad \text{and} \quad \bigwedge_{i=1}^{m} l_i \leq s_i \leq u_i \ .$$

2. Set $\mathcal{B}$ to be the set of additional variables $s_1, \ldots, s_m$.
3. Construct the tableau for A.
4. Determine a fixed order on the variables.
5. If there is no basic variable that violates its bounds, return "Satisfiable". Otherwise, let x_i be the first basic variable in the order that violates its bounds.
6. Search for the first suitable nonbasic variable x_j in the order for pivoting with x_i. If there is no such variable, return "Unsatisfiable".
7. Perform the pivot operation on x_i and x_j.
8. Go to step 5.

[2] Decision procedures. Springer

Linear integer arithmetic (LIA) – decision procedure

We consider the same kind of system of linear equations

$$\begin{cases} \mathbf{Ax} = \mathbf{0} \\ \ell \leqslant \mathbf{x} \leqslant \mathbf{u} \end{cases}$$

except that variables take integer values.

Definition – relaxed problem

Given an integer linear system $\mathcal{S}$, its relaxation is $\mathcal{S}$ without the integrality requirements, *i.e.*, the variables are not required to be integers.

Pseudo code of the branch and bound algorithm[3]

Algorithm 5.3.1: FEASIBILITY-BRANCH-AND-BOUND

Input: An integer linear system S
Output: "Satisfiable" if S is satisfiable, "Unsatisfiable" otherwise

```
procedure SEARCH-INTEGRAL-SOLUTION(S)
    res = LP_feasible (relaxed(S));
    if res = "Unsatisfiable" then return ;                ▷ prune branch
    else
        if res is integral then                   ▷ integer solution found
                abort("Satisfiable");
        else
            Select a variable v that is assigned a nonintegral value r;
            SEARCH-INTEGRAL-SOLUTION (S ∪ (v ≤ ⌊r⌋));
            SEARCH-INTEGRAL-SOLUTION (S ∪ (v ≥ ⌈r⌉));
                                   ▷ no integer solution in this branch

procedure FEASIBILITY-BRANCH-AND-BOUND(S)
    SEARCH-INTEGRAL-SOLUTION(S);
    return ("Unsatisfiable");
```

[3] Decision procedures. Springer

Conclusion

Linear arithmetic comes in two flavours

- Continuous (reals) for which decision procedures exist
  - Gaussian Elimination for linear equations
  - **General simplex algorithm** for linear inequalities
  - Fourier-Motzkin for linear inequalities
- Discrete (integers) for which decision procedures exist
  - **Branch and Bound** for integer linear inequalities.
  - The Omega-Test method for integer linear inequalities.

Theory of nonlinear real expressions