Satisfiability Checking - WS 2019/2020

Written Exam II

Tuesday, May 19, 2020

Sample solution

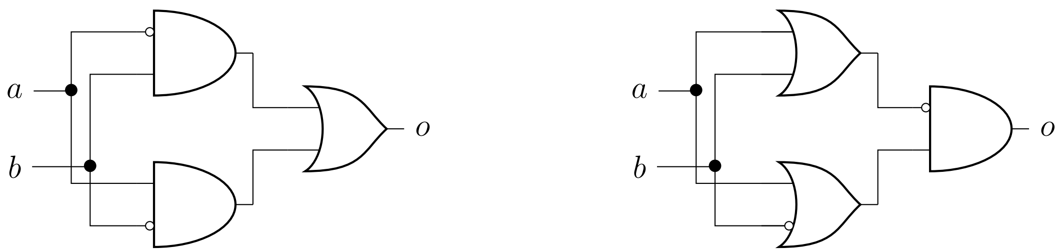
1.) SAT Checking

6+3+11 Points

i) Logical circuits are often depicted graphically using the symbols defined below.

(a) $o = a \wedge b$ (b) $o = a \vee b$ (c) $o = \neg a \wedge b$ (d) $o = a \vee \neg b$

An engineer designed the following two circuits and claims they are equivalent, that is for all possible inputs a and b, their outputs are the same. Formulate the two circuits as propositional logic formulas $\varphi_1(a, b)$ and $\varphi_2(a, b)$ and give a formula φ that allows a SAT solver to check whether they are in fact equivalent. Black dots represent junctions of wires and not negations.



(a) $\varphi_1(a, b) =$

(b) $\varphi_2(a, b) =$

(c) $\varphi =$

Solution:

(a) $\varphi_1(a, b) = (\neg a \wedge b) \vee (a \wedge \neg b)$

(b) $\varphi_2(a, b) = \neg(a \vee b) \wedge (a \vee \neg b)$

(c) $\varphi = \exists a, b.(\varphi_1(a, b) \wedge \neg\varphi_2(a, b)) \vee (\neg\varphi_1(a, b) \wedge \varphi_2(a, b))$

ii) Are the circuits in fact equivalent? If they are, argue why. If not, give inputs a and b such that the outputs differ.

Solution: They are not equivalent.
For a and $\neg b$ the left circuit outputs *true* while the right circuit outputs *false*. /
For $\neg a$ and b the left circuit outputs *true* while the right circuit outputs *false*. /
For $\neg a$ and $\neg b$ the left circuit outputs *false* while the right circuit outputs *true*.

iii) Now consider the following propositional formula in conjunctive normal form, which is satisfiability-equivalent to φ from i). Solve this formula using the CDCL algorithm as given in the lecture. You may select any decision heuristic. Explicitly state all decisions and propagations (including the clauses used for these propagations), as well as all conflict resolutions with the learnt clauses.

$$
\begin{array}{llll}
(a \vee \neg h_2) & \wedge(a \vee \neg b \vee \neg h_4) & \wedge(\neg a \vee \neg h_1) & \wedge(\neg a \vee \neg h_3) \\
\wedge(b \vee \neg h_1) & \wedge(\neg b \vee \neg h_2) & \wedge(\neg b \vee \neg h_3) & \wedge(h_1 \vee h_2 \vee \neg h_5) \\
\wedge(\neg h_1 \vee \neg h_2 \vee \neg h_7) & \wedge(h_3 \vee \neg h_8) & \wedge(\neg h_3 \vee \neg h_4 \vee \neg h_6) & \wedge(h_4 \vee \neg h_8) \\
\wedge(h_5 \vee h_7 \vee \neg h_9) & \wedge(h_5 \vee h_8 \vee \neg h_9) & \wedge(h_6 \vee h_7 \vee \neg h_9) & \wedge(h_6 \vee h_8 \vee \neg h_9) \\
\wedge(h_9) & & &
\end{array}
$$

Solution:

$$
\begin{array}{rlr}
\text{Propagation:} & h_9 & (h_9) \\
\text{Decision:} & a & \\
\text{Propagation:} & \neg h_1 & (\neg a \vee \neg h_1) \\
\text{Propagation:} & \neg h_3 & (\neg a \vee \neg h_3) \\
\text{Propagation:} & \neg h_8 & (h_3 \vee \neg h_8) \\
\text{Propagation:} & h_5 & (h_5 \vee h_8 \vee \neg h_9) \\
\text{Propagation:} & h_6 & (h_6 \vee h_8 \vee \neg h_9) \\
\text{Propagation:} & h_2 & (h_1 \vee h_2 \vee \neg h_5) \\
\text{Propagation:} & \neg b & (\neg b \vee \neg h_2) \\
\text{Decision:} & h_4 & \\
\text{Decision:} & h_7 &
\end{array}
$$

After propagating h_9 different decisions can be made, depending on the decision heuristic. Alternative satisfying assignments:

$$
\begin{array}{rrrrrrrrrrr}
a & \neg b & \neg h_1 & h_2 & \neg h_3 & & h_5 & h_6 & & \neg h_8 & h_9 \\
\neg a & b & h_1 & \neg h_2 & \neg h_3 & \neg h_4 & h_5 & h_6 & & \neg h_8 & h_9 \\
\neg a & \neg b & \neg h_1 & \neg h_2 & h_3 & h_4 & \neg h_5 & \neg h_6 & h_7 & h_8 & h_9
\end{array}
$$

2.) SMT Solving **3+2+13 Points**

i) Name the three main requirements on a theory solver that are desirable for SMT solving.

Solution:

(a) Incrementality: In less lazy solving we extend the set of constraints. The solver should make use of the previous satisfiability check for the check of the extended set.

(b) (Preferably minimal) infeasible subsets: Compute a reason for unsatisfaction.

(c) Backtracking: The theory solver should be able to remove constraints in inverse chronological order.

ii) Describe eager SMT solving.

Solution: Eager SMT solving transforms logical formulas over some theories into satisfiability-equivalent propositional logic formulas and applies SAT solving.

iii) Consider the following equality logic formula:

$$\begin{array}{rcclclclc} \varphi^{EQ} & := & & (& x_3 = x_4 & \vee & x_2 = x_4 & & &) \\ & & \wedge & (& x_3 = x_4 & \vee & x_1 = x_2 & \vee & x_1 = x_4 &) \\ & & \wedge & (& \neg(x_1 = x_2) & \vee & x_1 = x_4 & & &) \\ & & \wedge & (& \neg(x_3 = x_4) & \vee & \neg(x_2 = x_4) & \vee & \neg(x_2 = x_3) &) \end{array}$$

The Boolean abstraction of this formula is

$$(a \vee b) \wedge (a \vee c \vee d) \wedge (\neg c \vee d) \wedge (\neg a \vee \neg b \vee \neg e)$$

Explain how a *less-lazy* SMT solver solves φ^{EQ} for satisfiability as presented in the lecture. Describe

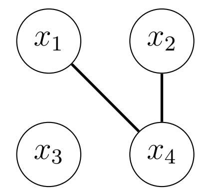
- the *SAT solver's assignments with decision levels* before each theory solver invocation,
- the *equivalence classes* determined by the theory solver in each invocation and whether a *conflict* occurs,
- for each theory conflict, how a *minimal infeasible subset* is generated and
- the subsequent *conflict resolution* in the SAT solver.

Assume that

- if the SAT solver makes a decision it chooses the lexicographically smallest unassigned variable and assigns it to *false*, and that
- all equalities (or disequalities) are given to the theory solver (i.e. no constraints are dropped because only one polarity is present).

Solution: Calculations:

SAT solver	**Theory solver**

$DL0:$	Received (in)equalities: $\emptyset$ There is no conflict.
$DL0:$ $DL1: \quad a:0, b:1$	Received (in)equalities: $x_3 \neq x_4, \quad x_2 = x_4$ Equivalence classes: $\langle x_1 \rangle = \{x_1\}$ $\langle x_2 \rangle = \{x_2, x_4\}$ $\langle x_3 \rangle = \{x_3\}$ There is no conflict.
$DL0:$ $DL1: \quad a:0, b:1$ $DL2: \quad c:0, d:1$	Received (in)equalities: $x_3 \neq x_4, \quad x_2 = x_4, \quad x_1 \neq x_2, \quad x_1 = x_4$ Equivalence classes: $\langle x_1 \rangle = \{x_1, x_2, x_4\}$ $\langle x_3 \rangle = \{x_3\}$ We have a conflict for $x_1 \neq x_2$, as both are in the same equivalence class. We consider the equality graph (without polarity) of the *equations* and search for a shortest path from x_1 to x_2.  Hence, we find the infeasible subset $\{x_1 = x_4, x_2 = x_4, x_1 \neq x_2\}$.
The infeasible subset leads to the clause $(\neg b \vee c \vee \neg d)$. As it is not asserting we apply resolution: $$\frac{(\neg b \vee c \vee \neg d) \quad (a \vee c \vee d)}{(a \vee \neg b \vee c)}$$ We backtrack to decision level 1, add the clause $(a \vee \neg b \vee c)$ to the SAT solver's set of clauses, assign c to true and achieve after BCP has been applied: $DL0:$ $DL1: \quad a:0, b:1, c:1, d:1$	Received (in)equalities: $x_3 \neq x_4, \quad x_2 = x_4, \quad x_1 = x_2, \quad x_1 = x_4$ Equivalence classes: $\langle x_1 \rangle = \{x_1, x_2, x_4\}$ $\langle x_3 \rangle = \{x_3\}$ There is no conflict.
$DL0:$ $DL1: \quad a:0, b:1, c:1, d:1$ $DL2: \quad e:0$	Received (in)equalities: $x_3 \neq x_4, \quad x_2 = x_4, \quad x_1 = x_2, \quad x_1 = x_4,$ $x_2 \neq x_3$ Equivalence classes: $\langle x_1 \rangle = \{x_1, x_2, x_4\}$ $\langle x_3 \rangle = \{x_3\}$ There is no conflict. The SMT solver returns SAT.

3.) Fourier-Motzkin Variable Elimination **9+5 Points**

i) *Eliminate* x by applying the Fourier-Motzkin variable elimination method to the following set of linear real-arithmetic constraints:

$$\begin{array}{rrcrcrcl} c_1: & -x & + & 2y & - & 2z & \leq & 0 \\ c_2: & x & - & y & + & z & \leq & -2 \\ c_3: & x & - & y & - & z & \geq & 2 \\ c_4: & x & - & 2y & & & \leq & 1 \end{array}$$

Answer without further applying Fourier-Motzkin: Is this constraint set satisfiable? Why?

Solution: We first identify the lower and upper bounds:

$$\begin{array}{rrclcl} c_1: & 2y-2z & \leq & x & & \\ c_2: & & & x & \leq & y-z-2 \\ c_3: & y+z+2 & \leq & x & & \\ c_4: & & & x & \leq & 2y+1 \end{array}$$

We now combine all lower-upper-bound pairs, i.e. (c_1, c_2), (c_1, c_4), (c_3, c_2) and (c_3, c_4).

$$\begin{array}{lrcl} (c_1,c_2): & 2y-2z & \leq & y-z-2 \\ \Rightarrow & y & \leq & z-2 \\ \hline (c_1,c_4): & 2y-2z & \leq & 2y+1 \\ \Rightarrow & -2z & \leq & 1 \\ \Rightarrow & -1/2 & \leq & z \\ \hline (c_3,c_2): & y+z+2 & \leq & y-z-2 \\ \Rightarrow & 2z & \leq & -4 \\ \Rightarrow & z & \leq & -2 \\ \hline (c_3,c_4): & y+z+2 & \leq & 2y+1 \\ \Rightarrow & z+1 & \leq & y \end{array}$$

The resulting constraint set contains lower and upper bounds on both y and z that contradict each other. I.e. $-1/2 \leq z$ and $z \leq -2$ or $y \leq z-2$ and $z+1 \leq y$ respectively. Therefore, subsequent elimination steps will produce a contradiction and the constraint set is unsatisfiable.

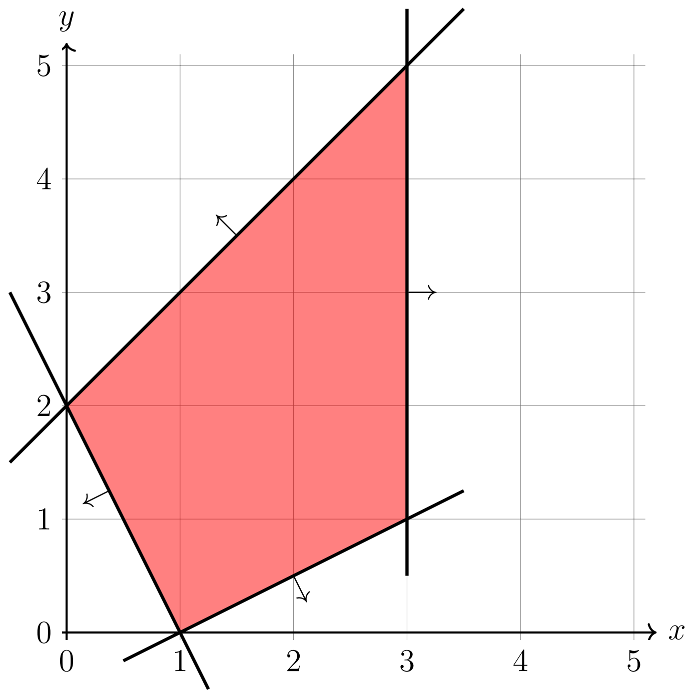
ii) Sketch the *solution set* of the following set of linear real-arithmetic constraints graphically:

$$\begin{array}{rrcrcl} c_1: & -x & + & y & \leq & 2 \\ c_2: & x & & & \leq & 3 \\ c_3: & x & - & 2y & \leq & 1 \\ c_4: & 2x & + & y & \geq & 2 \end{array}$$

Solution: The following equalities describe the surfaces of the solution sets (halfspaces) of the constraints:

$$\begin{array}{rcrcl} y & = & x & + & 2 \\ x & = & & & 3 \\ y & = & 1/2 \cdot x & - & 1/2 \\ y & = & -2x & + & 2 \end{array}$$

The common solution set of all constraints is the intersection of the halfspaces.



4.) Simplex **12+3 Points**

i) Apply one *pivoting step* and update the current assignment, where the variable order is $x_1 < x_2 < s_1 < s_2 < s_3 < s_4$ and where the current tableau, the bounds and the current assignment are given by:

	s_4	s_3
s_2	1	3
x_1	-2	1
s_1	-1	2
x_2	2	1

$$\begin{array}{lll} s_1 & \leq & -2 \\ s_2 & \geq & 8 \\ s_3 & \geq & 2 \\ s_4 & \geq & 1 \end{array}$$

$$\begin{array}{ll} \alpha(s_1) & = 3 \\ \alpha(s_2) & = 7 \\ \alpha(s_3) & = 2 \\ \alpha(s_4) & = 1 \\ \alpha(x_1) & = 0 \\ \alpha(x_2) & = 4 \end{array}$$

Solution: Two basic variables violate their respective bounds: s_1 and s_2. According to the variable order, we try to pivot s_1 and only s_4 is suitable for pivoting. We decrease the value of s_1 by 5.

We pivot the third row and first column with

$$s_1 = -s_4 + 2 \cdot s_3 \Leftrightarrow s_4 = -s_1 + 2 \cdot s_3$$

and obtain

	s_1	s_3
s_2	-1	5
x_1	2	-3
s_4	-1	2
x_2	-2	5

$$\begin{array}{lll} s_1 & \leq & -2 \\ s_2 & \geq & 8 \\ s_3 & \geq & 2 \\ s_4 & \geq & 1 \end{array}$$

$$\begin{array}{ll} \alpha(s_1) & = -2 \\ \alpha(s_2) & = 12 \\ \alpha(s_3) & = 2 \\ \alpha(s_4) & = 6 \\ \alpha(x_1) & = -10 \\ \alpha(x_2) & = 14 \end{array}$$

ii) The simplex method terminates on the resulting tableau. Is the tableau satisfied or conflicting? Why? Give the satisfying assignment for the original variables or the infeasible subset.

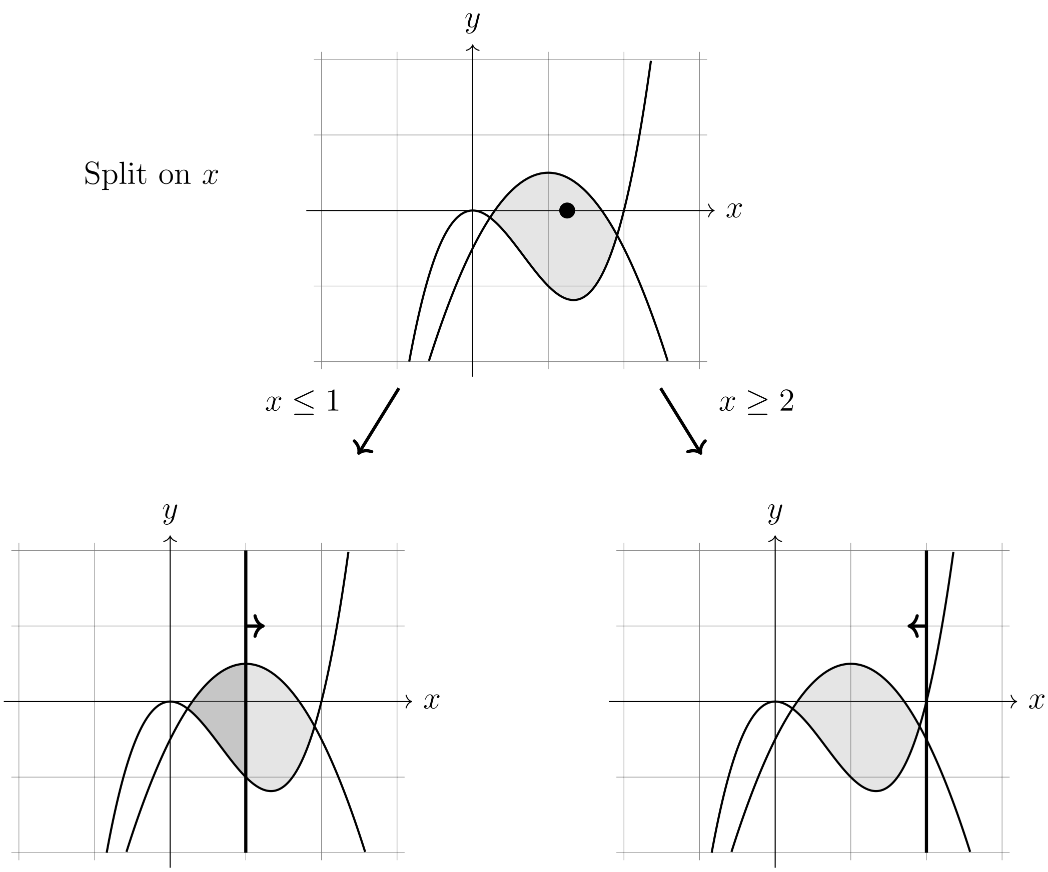
Solution: The tableau is satisfied. All auxiliary variables $s_1, \dots s_4$ are within their bounds. The assignment is simply taken from α and we have $\alpha(x_1) = -10$ and $\alpha(x_2) = 14$.

5.) Integer Arithmetic **7+3 Points**

i) Branch&Bound employs a decision procedure for real arithmetic to solve integer arithmetic problems. Though we only showed linear examples in the lecture it can also be applied to non-linear examples, given an algorithm for non-linear real arithmetic. Consider the following example and *execute one branching* of the Branch&Bound method.

- Give the *variable used for branching.*
- Give the *new bounds* for each sub-problem.
- Sketch the *branch* for each sub-problem graphically.
- Sketch the *real solution space* for each sub-problem.

Solution:



ii) In some applications, problems contain both integer-valued and real-valued variables. Can we apply Branch&Bound (as presented in the lecture) to such problems? How?

Solution: This is possible. When checking whether the current real assignment is integral, we only need to check the integer-valued variables and ignore the assignments for real-valued variables.

6.) Interval Constraint Propagation **6+8 Points**

i) Apply *basic interval arithmetic* as presented in the lecture to compute the following:

Solution:

a) $[1;2] + [-3;-3] + [-1;3] = [-2;-1] + [-1;3] = [-3;2]$

b) $[2;4] \cdot [-2;2] = [-8;8]$

c) $[1;\infty) \cdot [-2;-1] = (-\infty;-1]$

d) $[1;2]/[-2;1] = (-\infty;-\frac{1}{2}] \cup [1;\infty)$

ii) Assume an initial box $\mathcal{B}_0 = [2;5] \times [1;6]$ for the variables x and y and a set of constraints as follows:

$$C_1 : -x + 2y - 4 = 0$$
$$C_2 : -x - 2y + 10 \leq 0$$

Please contract each constraint C_i once individually for each variable. Do not reuse previous contractions, i.e., always use $\mathcal{B}_0$ as given above for your initial interval values for the variables. Please give all intermediate computations.

Solution:

$$
\begin{aligned}
C_{1,x}: -x =& -2y + 4 \\
x =& 2y - 4 && \text{interval arithmetic} \\
x \in& [2;5] \cap (2[1;6] - 4) \\
x \in& [2;5] \cap ([2;12] - 4) \\
x \in& [2;5] \cap [-2;8] \Rightarrow x \in [2;5] \\
C_{1,y}: 2y =& x + 4 \\
y =& \frac{1}{2}x + 2 && \text{interval arithmetic} \\
y \in& [1;6] \cap ([2;5]/2 + 2) \\
y \in& [1;6] \cap ([1;2.5] + 2) \\
y \in& [1;6] \cap [3;4.5] \Rightarrow y \in [3;4.5] \\
C_{2,x}: -x \leq& 2y - 10 \\
x \geq& -2y + 10 && \text{interval arithmetic} \\
x \geq& -[2;12] + 10 \\
x \geq& [-12;-2] + 10 \\
x \geq& [-2;8] && \text{lower bound} \\
x \in& [2;5] \cap [-2;\infty) \Rightarrow x \in [2;5] \\
C_{2,y}: -2y \leq& x - 10 \\
y \geq& \frac{-x + 10}{2} && \text{interval arithmetic} \\
y \geq& \frac{[5;8]}{2} \\
y \geq& [2.5;4] && \text{lower bound} \\
y \in& [1;6] \cap [2.5;\infty) \Rightarrow y \in [2.5;6]
\end{aligned}
$$

7.) Subtropical Satisfiability **6+6 Points**

i) Consider the polynomial $f(x, y) = x^2 - y - \frac{3}{4}$. We have already evaluated $f(x, y)$ at the two points $f(1, 1) = -\frac{3}{4}$ and $f(2, 2) = \frac{5}{4}$.

Use this information to compute a real root of $f(x, y)$ as shown in the lecture:

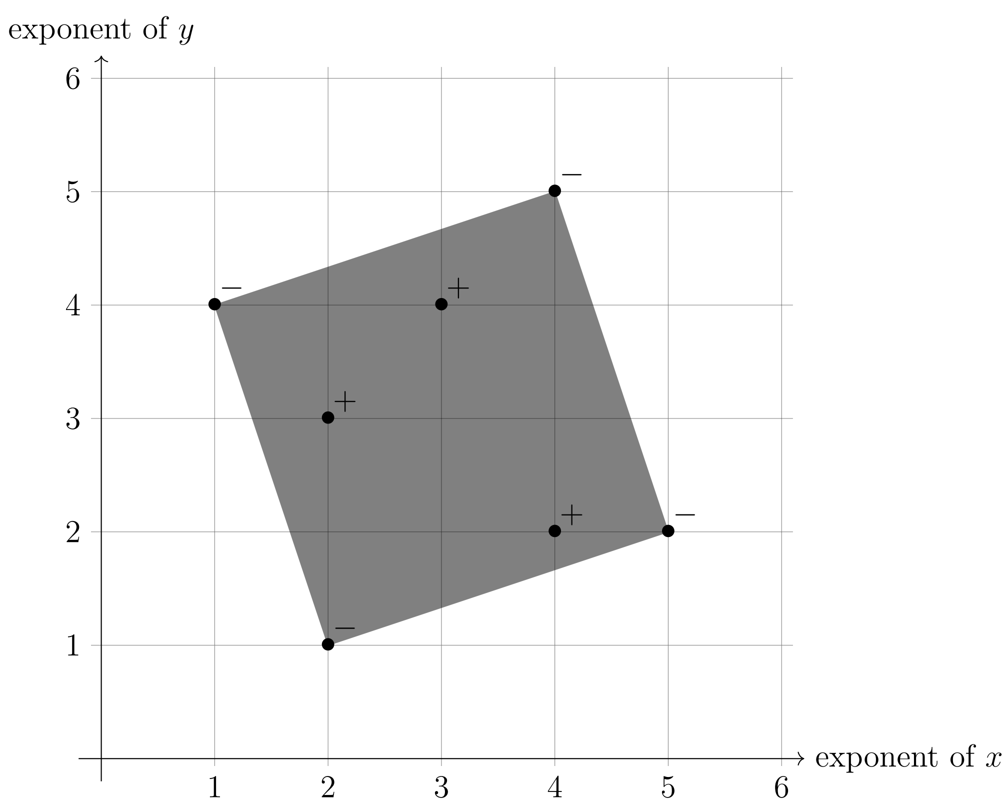
- Construct a univariate $f^*(t)$ for $f(x, y)$.
- Identify a real root of $f^*(t)$ with $t \in [0, 1]$.
- Use this real root to compute a real root of $f(x, y)$.

Solution:

- From $f(1, 1) < 0$ and $f(2, 2) > 0$ we identify the parametrization $x \mapsto 1 + t$ and $y \mapsto 1 + t$ (such that we have $(1, 1)$ for $t = 0$ and $(2, 2)$ for $t = 1$).
- With this we get $f^*(t) = (1 + t)^2 - (1 + t) - \frac{3}{4} = t^2 + t - \frac{3}{4}$ whose roots are -1.5 and 0.5.
- For $t \in [0, 1]$ we thus have $t = 0.5$ as real root and thus obtain the real root $f(1.5, 1.5) = 0$.

ii) Visualize $newton(g)$ for the polynomial $g(x, y) = 2x^2y^3 - 5yx^2 + 7x^3y^4 + x^4y^2 - 2x^4y^5 - 4y^4x - x^5y^2$.
Is the subtropical method as presented in the lecture suitable to determine the satisfiability of the constraint $g = 0$? Explain why not or describe how to apply the method to this example.

Solution:

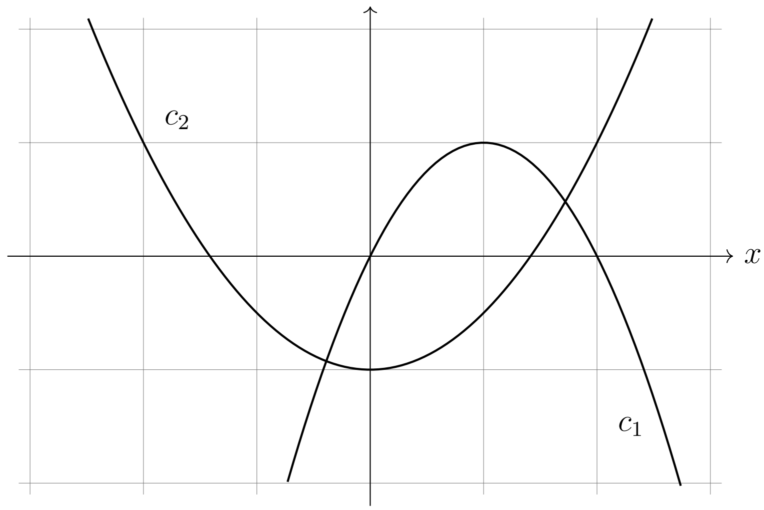


We read the exponents of g and put them into the graphic. $newton(g) = conv(frame(g))$. The subtropical method as presented in the lecture is not suitable to solve the given constraint, because there exists no $p \in \mathit{frame}^+(g)$ with $p \in V(newton(g))$. No hyperplane can separate some $p \in \mathit{frame}^+(g)$ from $\mathit{frame}(g) \setminus \{p\}$.

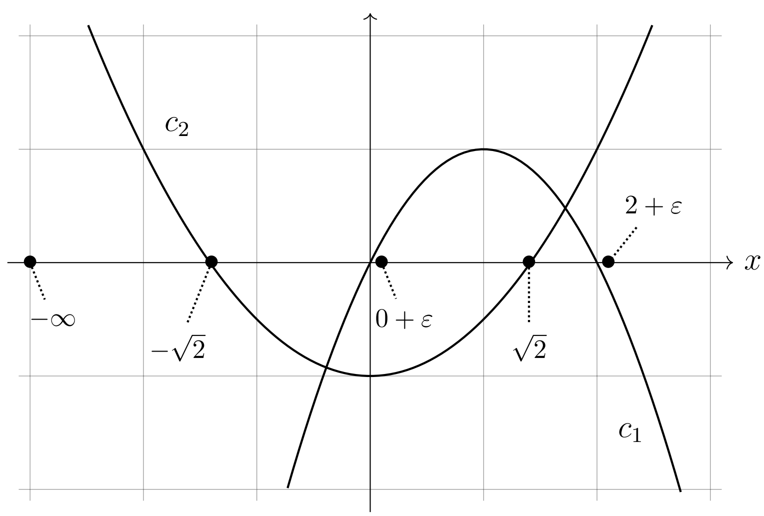
8.) Virtual Substitution **5 Points**

Consider the following constraints and their graphical depiction below. Give the test candidates for x resulting from these constraints *graphically* and label them appropriately.

$$c_1 : -x^2 + 2x > 0 \qquad c_2 : \frac{1}{2}x^2 - 1 \leq 0$$



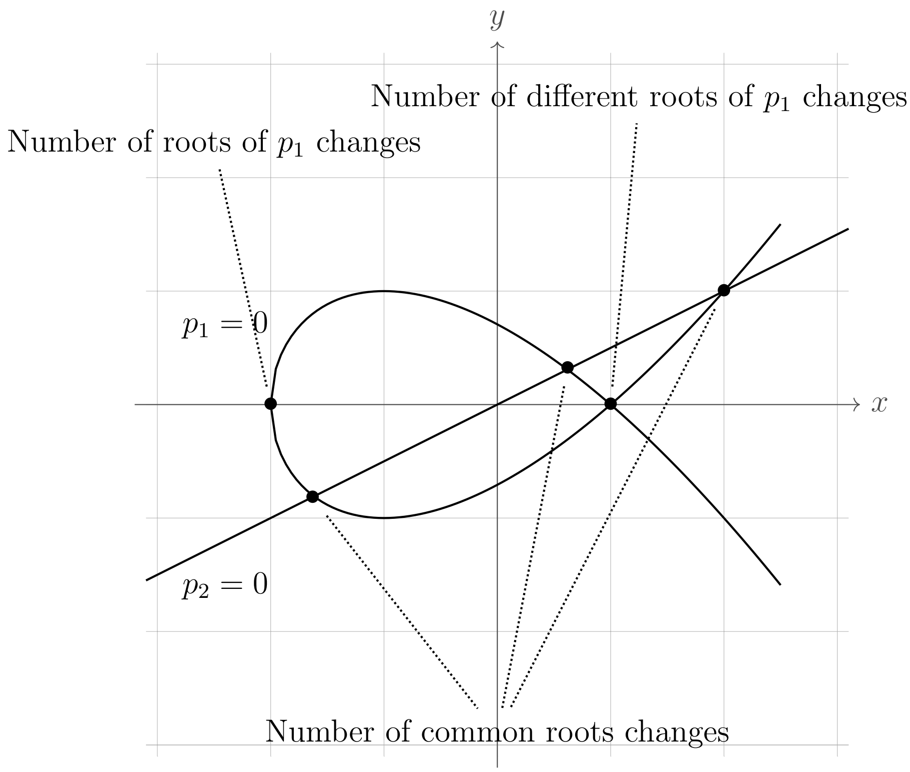
Solution:



9.) Cylindrical Algebraic Decomposition **8+4 Points**

i) Consider two polynomials p_1 and p_2 whose varieties are depicted in the picture below. Assume that we project onto the x axis. Identify every point that yields a cylinder boundary and state why it does so in the context of the definition of *delineability* from the lecture.

Solution:



ii) One requirement of SMT compliancy (as defined in the lecture) is the generation of *infeasible subsets*. Consider a one-dimensional CAD. What is the basic idea to compute infeasible subsets once the CAD method has determined infeasibility? How could we attempt to get *small* infeasible subsets?

Solution: As CAD has no way of producing an infeasible subset by itself, we compute it based on the sample points. We simply proceed as follows: for every sample point we collect one constraint that conflicts with this sample point. The set of all these constraints is an infeasible subset.

We may have *multiple conflicting constraints* for certain sample points. We can try to find a small subset of the constraints that "covers" all sample points, essentially yielding a *set cover problem*.