universität innsbruck

Constraint Solving | SS 2024 | LVA 703304

EXAM 1 | June 25, 2024

$\boxed{1}$ (a) *calculation + explanation*

$$\varphi := \exists x.\exists y.\ 2x + 3y < 2 \wedge -x + 4y \geq 1$$

eliminate $\geq$

$$\longleftrightarrow \exists x.\exists y.\ 2x + 3y < 2 \wedge (-x + 4y > 1 \vee -x + 4y = 1)$$

gather y on one side

$$\longleftrightarrow \exists x.\exists y.\ 3y < 2 - 2x \wedge (4y > 1 + x \vee 4y = 1 + x)$$

eliminate leading coefficient of y

$$\longleftrightarrow \exists x.\exists y.\ y < \frac{2-2x}{3} \wedge \left(y > \frac{1+x}{4} \vee y = \frac{1+x}{4}\right)$$

determine $S = \{\frac{2-2x}{3}, \frac{1+x}{4}\}$ and eliminate y, both infinite projections simplify to $\bot$

$$\longleftrightarrow \exists x. \bigvee_{s,t \in S} \frac{s+t}{2} < \frac{2-2x}{3} \wedge \left(\frac{s+t}{2} > \frac{1+x}{4} \vee \frac{s+t}{2} = \frac{1+x}{4}\right)$$

(b) *calculation + explanation*

The formula can be written as

$$2x + 3y < 2$$
$$-x + 4y \geq 1$$

and the elimination of strict inequalities yields

$$2x + 3y \leq 2 - \delta$$
$$-x + 4y \geq 1$$

So we get the following initial tableau and bounds and an initial assignment where everything becomes 0:

tableau	x	y	bounds	assignment	x	y	s	t
s	2	3	$s \leq 2 - \delta$		0	0	0	0
t	-1	4	$t \geq 1$					

There is a violation for t. Both x and y are suitable, but Bland's rule will select x. Pivoting of t and x results in:

tableau	t	y	bounds	assignment	x	y	s	t
s	-2	11	$s \leq 2 - \delta$		0	0	0	0
x	-1	4	$t \geq 1$					

Updating the assignment $t := 1$ results in:

tableau	t	y	bounds	assignment	x	y	s	t
s	-2	11	$s \leq 2 - \delta$		-1	0	-2	1
x	-1	4	$t \geq 1$					

Since no bound is violated, no further iterations of the main loop are required.

$\boxed{2}$ *definition, algorithm and example calculation*

(a) A mixed solved form is similar to a solved form, where condition 1. is replaced as follows:

1'. each entry in the list is of the shape $x = e_x$ where e_x is a linear expression and either $x \in \mathcal{V}_{\mathbb{Q}}$ or $x \in \mathcal{V}_{\mathbb{Z}}$ and e_x has only integer coefficients and uses only variables of $\mathcal{V}_{\mathbb{Z}}$.

(b) We take a two-staged approach, where first all variables in $\mathcal{V}_{\mathbb{Q}}$ are eliminated, and afterwards Griggio's algorithm is applied. So, here are the steps:

i. if all variables in the equations are from $\mathcal{V}_{\mathbb{Z}}$, just apply Griggio's algorithm

ii. otherwise, pick some equations e that contains a variable $x \in \mathcal{V}_{\mathbb{Q}}$

iii. reorder equation e to the form $x = e'$ with e' not containing x.

iv. store $x = e'$ as part of the final mixed solved form

v. remove e from the set of equations and substitute x by e' in the remaining equations

vi. normalize the equations

vii. goto i.

(c) The mixed algorithm works as follows:

- the first equation is rearranged to $x = \frac{5}{2} - \frac{3}{2}y$
- substituting in the second equation yields $3(\frac{5}{2} - \frac{3}{2}y) + 2y + 5z = 4$, so after normalization this results in $-5y + 10z = -7$
- now Griggio's algorithm detects unsatisfiability since the gcd of 5 and 10 does not divide 7.

Using Griggio's algorithm directly calculates as follows:

- select $x = -y + 2 + u$ for some fresh variable u and substitute

$$\begin{aligned} 2(-y + 2 + u) + 3y &= 5 \\ 3(-y + 2 + u) + 2y + 5z &= 4 \end{aligned}$$

  which normalizes to

$$\begin{aligned} 2u + y &= 1 \\ -y + 3u + 5z &= -2 \end{aligned}$$

- select $y = 1 - 2u$ and substitute

$$-(1 - 2u) + 3u + 5z = -2$$

  which normalizes to

$$5u + 5z = -1$$

- detect unsatisfiability, since 5 does not divide 1.

$\boxed{3}$ *convexity answer + brief explanation, calculation*

(a) Difference logic over $\mathbb{Q}$ is convex. Reason: it is a sub-logic of LRA, which is already convex.

(b) Since we are in the convex case, we choose the deterministic version of Nelson–Oppen.

First, we have to purify the formula into the EUF formula φ and the DL formula ψ:

$$\underbrace{f(f(u)) = f(a) \land y = f(v) \land y \neq a}_{\varphi} \land \underbrace{x - z = 7 \land u = x - 5 \land v = z + 2}_{\psi}$$

Next we determine the shared variables: $\mathit{Vars}(\varphi) \cap \mathit{Vars}(\psi) = \{u, y, v\} \cap \{x, z, u, v\} = \{u, v\}$.

Checking DL-satisfiability of ψ results in a satisfying assignment, e.g., $u = v = 2, x = 7, z = 0$, but $\psi \land u \neq v$ is unsatisfiable, so $u = v$ is added to the implied equation E.

Checking EUF-satisfiability of $\varphi \land E$ reports satisfiability: the congruence closure algorithm on $f(f(u)) = f(a) \land y = f(v) \land u = v$ with target equation $y = a$ results in equivalence classes

- $f(f(u)), f(a)$
- $y, f(v), f(u)$
- u, v
- a

where y and a are in different classes.

Since no contradiction is detected and no further equations are deduced, the Nelson–Oppen algorithm stops and reports satisfiability.

Remark: if one would have applied the non-deterministic version of Nelson–Oppen, then there would have been not much difference in comparison to the deterministic version, since there are only two equivalence classes: either $u = v$ or $u \neq v$.

$\boxed{4}$ *description of encoding, example application*

We just translate the formula into bit-vector arithmetic using unsigned comparison operations. However, we have to take care that no overflows happen, and that the search range is restricted. To this end, we

- first calculate upper bounds for the maximally required number of bits for each subexpression, assuming that all variables use at most b bits and determine m as the maximum of all these numbers;
- afterwards we translate the formula into BV-arithmetic with m bits;
- and finally we add constraints $x <_u 2^b$ for all variables to ensure that we restrict the allowed solutions to $0, \ldots, 2^b - 1$.

On the example formula with $b = 3$ we see that the maximal value that we can get for the subexpressions are $7 \cdot 7^2 = 343, 5 + 7 = 12, 7(7 + 7) = 98, 7^2 + 8 = 57, \ldots$ with maximum 343. To represent 343 we need 9 bits, so $m = 9$. Hence, the formula is converted to BV-arithmetic with 9 bits and we add the constraints $x <_u 8 \wedge y <_u 8 \wedge z <_u 8$. In total we get:

$$xy^2 \geq_u 5 + x \wedge y(x + z) \geq_u x^2 + 8 \vee 3x \geq_u 2y + z \wedge x <_u 8 \wedge y <_u 8 \wedge z <_u 8$$

$\boxed{5}$

Question	Yes	No
The decision procedure for difference logic is based on Dijkstra's shortest-path-algorithm.	☐	☑
In order to detect equalities for constraints $A\vec{x} \leq \vec{b}$, Bromberger and Weidenbach's method invokes the simplex algorithm on $A\vec{x} > \vec{b}$.	☐	☑
The small-model property of LIA is essential for termination of the branch-and-bound algorithm.	☑	☐
SAT is a decision problem in PSPACE.	☑	☐
$\forall x, y.\ x \neq y \wedge x \leq u \wedge v \leq y \longrightarrow a[x] = b[y] + 3$ can be reformulated into an equivalent array property.	☐	☑

spare page