Automated Deduction - SS 2019

Problem Set 2
Due May 2, 2019, 11am

Problem 2.1. (10 points) Consider the following formula:

$$x = y \rightarrow (x = a \vee x = b),$$

where x, y are variables and a, b are constants.

Describe the class of interpretations that makes this formula valid.

Solution:
Note, that the given formula is equivalent to $x = a \vee x = b$.
The class of interpretations that make this formula valid consists of all interpretations I that interpret the sort of x, a, b wih the same domain and this domain contains either:

(i) only one element, in this case $I(a) = I(b)$.

or

(ii) two elements, in this case $I(a) \neq I(b)$.

Problem 2.2. (20 points) Consider two axiomatizable theories $\mathcal{T}_1$ and $\mathcal{T}_2$ such that $\Sigma_{\mathcal{T}_1} = \Sigma_{\mathcal{T}_2} = \Sigma$ and $A_{\mathcal{T}_2} \subseteq A_{\mathcal{T}_1}$. Recall that $\Sigma_{\mathcal{T}_i}$ and $A_{\mathcal{T}_i}$ denote, respectively, a signature and set of axioms defining $\mathcal{T}_i$, for $i = 1, 2$. Let F be a formula over signature Σ.

(a) If F is valid in $\mathcal{T}_1$, it is also valid in $\mathcal{T}_2$? Prove your answer or give a counterexample.

Solution:

No.

Consider standard first-order logic, with signature Σ. Let F be a first-order formula that is satisfiable, but not valid. (For example, the atomic formula $p(x)$, with $p \in \Sigma$ and variable x).

Take $A_{\mathcal{T}_2} = \emptyset$ and $A_{\mathcal{T}_1} = F$. Then, F is valid in $\mathcal{T}_1$. Yet, F is not valid in $\mathcal{T}_2$ as F is a satisfiable but not a valid first-order formula.

(b) If F is valid in $\mathcal{T}_2$, it is also valid in $\mathcal{T}_1$? Prove your answer or give a counterexample.

Solution:

Yes.

Assume F is valid in $\mathcal{T}_2$. That is, for any interpretation J of Σ with $J \models A_{\mathcal{T}_2}$, we have $J \models F$.

Take *an arbitrary $\mathcal{T}_1$-interpretation* I of Σ, that is an arbitrary interpretation I of Σ with such that $I \models A_{\mathcal{T}_1}$. As $A_{\mathcal{T}_2} \subseteq A_{\mathcal{T}_1}$, we have $I \models A_{\mathcal{T}_2}$. From the assumption that F is valid in $\mathcal{T}_2$, we then *obtain $I \models F$*. Thus, F is a valid in $\mathcal{T}_1$.

Problem 2.3. (30 points) For each formula below, decide whether it is satisfiable or not by relying on the decision procedure of the theory or arrays. If the formula is satisfiable, give an interpretation that satisfies the formula.

(a) $read(write(A, i, e), j) = e \ \wedge \ i \neq j \ \wedge read(A, j) \neq e$,

where i, e, j, are integer-valued constants and A is an array constant.

Solution: We use the decision procedure of the theory $\mathcal{T}_A$ of arrays. Using the (read-over-write) axioms of $\mathcal{T}_A$, we consider the following cases.

(a1) Using the (read-over-write-1) axiom, we get :

$$i = j \ \wedge e = e \ \wedge \ i \neq j \ \wedge read(A, j) \neq e.$$

Using a fresh function symbol f_A, the formula is is translated to the following conjunction of literals in $\mathcal{T}_E$:

$$i = j \ \wedge e = e \ \wedge \ i \neq j \ \wedge f_A(j) \neq e.$$

We now apply the congruence closure decision procedure of $\mathcal{T}_E$. The initial partition of the set of subterms of the formula is:

$$\{\{i\}, \{j\}, \{e\}, \{f_A(j)\}\}.$$

According to the literal $i = j$, we merge $\{i\}$ and $\{j\}$ and obtain the partition (no further propagation of congruence classes is possible):

$$\{\{i, j\}, \{e\}, \{f_A(j)\}\}.$$

Using the literal $e = e$, the congruence class $\{e\}$ is kept unchanged and we obtain the final partition (no propagation of congruence classes is possible):

$$\{\{i, j\}, \{e\}, \{f_A(j)\}\}.$$

In our formula we have $i \neq j$ which however contradicts the above partition of congruence classes where i and j are in the same congruence classe. The congruence closure algorithm of $\mathcal{T}_E$ applied to the above formula derives $\mathcal{T}_E$-unsatisfiability. We thus conclude the $\mathcal{T}_A$-unsatisfiability of the formula considered in this branch of the case split.

(a2) Using the (read-over-write-2) axiom, we get :

$$i \neq j \wedge read(A, j) = e \ \wedge \ i \neq j \ \wedge read(A, j) \neq e.$$

By introducing a fresh function f_A for the array A, we derive:

$$i \neq j \wedge f_A(j) = e \ \wedge \ i \neq j \ \wedge f_A(j) \neq e.$$

Applying the congruence closure algorithm of $\mathcal{T}_E$, we obtain the following final partition over the congruence classes of the subterms of the above formula:

$$\{\{i\}, \{j\}, \{f_A(j), e\}\}.$$

In our formula we have $f_A(j) \neq e$ which however contradicts the above partition of congruence classes where e and $f_A(j)$ are in the same congruence classe. The congruence closure algorithm of $\mathcal{T}_E$ applied to the above formula derives $\mathcal{T}_E$-unsatisfiability. We thus conclude the $\mathcal{T}_A$-unsatisfiability of the formula considered in this branch of the case split.

As both branches of the case split are unsatisfiability, we conclude that the input formula is $\mathcal{T}_A$-unsatisfiable.

(b) $read(write(write(A,i,e),j,f),k) = read(A,j) \wedge i = k \wedge read(A,k) \neq g \wedge read(A,j) \neq g$, where i, e, j, f, k, g, are integer-valued constants and A is an array constant.

Solution: We use the decision procedure of the theory $\mathcal{T}_A$ of arrays. Using the (read-over-write) axioms of $\mathcal{T}_A$, we consider the following cases.

(b1) Using the (read-over-write-1) axiom, we get:

$$j = k \ \wedge f = read(A,j) \ \wedge \ i = k \ \wedge \ read(A,k) \neq g \ \wedge \ read(A,j) \neq g.$$

Using a fresh function symbol f_A, the above formula is translated to the following conjunction of literals in $\mathcal{T}_E$:

$$j = k \ \wedge \ f = f_A(j) \ \wedge \ i = k \ \wedge \ f_A(k) \neq g \ \wedge \ f_A(j) \neq g.$$

Applying the congruence closure algorithm of $\mathcal{T}_E$, we obtain the following final partition over the congruence classes of the subterms of the above formula:

$$\{\{i,j,k\}, \{f_A(j), f_A(k), f\}, \{g\}.\}$$

The derived congruence classes do not contradict the equalities/disequalities of the formula above, hence we conclude that the formula is $\mathcal{T}_E$-satisfiable. As a result, *the original formula is $\mathcal{T}_A$-satisfiable.*

We use the above partition above to construct a model I that satisfies the input formula, as follows.

Let Z be a sort and we consider the domain Int of integer numbers. We consider array indexes and array elements to be of sort Z. We interpret the sort Z to be the domain Int, that is $Z^I := Int$. We next interpret arrays as functions on $Int \rightarrow Int$; that is for an array A, we have $A^I : Int \rightarrow Int$.

Note that our input formula also contains the constant e. We consider i, j, k, e, f, g to be of sort Z and define:

$$\begin{array}{ll} i^I := 1 & \\ k^I := 1 & \\ j^I := 1 & \\ e^I := 0 & \text{note: the value } e^I \text{ can be set arbitrarily} \\ f^I := 0 & \\ g^I := 2 & \\ A^I(u) := \begin{cases} 0, & \text{if } u \leq 1 \\ 1, & \text{otherwise} \end{cases} & \end{array}$$

This interpretation I is a model of our original input formula.

Problem 2.4. (20 points) Consider the formula:

$$g(f(a-2)) = a+2 \ \wedge \ g(f(b)) = b-2 \ \wedge \ b+1 = a-1,$$

where a, b are constants interpreted over integers, f, g are function symbols, and $+, -, 1, 2$ are interpreted in the standard way over the integers.

(a) Use the Nelson-Oppen decision procedure to determine the satisfiability of the above formula. Use the decision procedures for the theory of uninterpreted functions and use simple mathematical reasoning for deriving new equalities among the constants in the theory of linear integer arithmetic. If the formula is satisfiable, give an interpretation that satisfies the formula.

Solution: The input formula is expressed in the combined theories of $\mathcal{T}_E$ and the theory $\mathcal{T}_Z$ of linear integer arithmetic. The formula is the conjunction of the following literals:

$$\begin{array}{l} g(f(a-2)) = a+2 \\ g(f(b)) = b-2 \\ b+1 = a-1 \end{array} ,$$

so we apply the Nelson-Oppen decision procedure to determine its satisfiability.

We introduce the following new constants:

$$\begin{array}{lll} c_1 & \text{to denote} & a-2 \\ c_2 & \text{to denote} & a+2 \\ c_3 & \text{to denote} & b-2 \end{array}$$

By separating reasoning in the various theories, we should satisfy the following set of literals in two theories:

- In $\mathcal{T}_Z$:

$$\begin{array}{l} c_1 = a-2 \\ c_2 = a+2 \\ c_3 = b-2 \\ b+1 = a-1 \end{array}$$

- In $\mathcal{T}_E$:

$$\begin{array}{l} g(f(c_1)) = c_2 \\ g(f(b)) = c_3 \end{array}$$

From the $\mathcal{T}_Z$-theory literals, the arithmetic reasoner will derive the following *shared equality*:

$$b = c_1.$$

From this shared equality and the $\mathcal{T}_E$-theory literals, the congruence closure algorithm of $\mathcal{T}_E$ will derive the *shared equality*:

$$c_2 = c_3$$

Using the two shared equalities $b = c_1$ and $c_2 = c_3$ in conjunction with the $\mathcal{T}_Z$-theory literals, the arithmetic reasoner of $\mathcal{T}_Z$ will derive the (shared) equality $2 = -4$, that is a contradiction.

Hence, the input formula is unsatisfiable.

(b) Encode the above formula as an input to the Z3 SMT solver and evaluate Z3 on your encoding, using `http://rise4fun.com/Z3`. Interpret the result of Z3. Provide the electronic version of your Z3 encoding together with your solution.

Problem 2.5. (20 points) Consider the formula:

$$read(write(A, a, f(b)), c+1) = f(d) \ \wedge \ f(b) \neq f(d-1) \ \wedge \ (b+1 = d \ \vee \ c = a-1),$$

where A is an array constant with integer elements, a, b, c, d are constants interpreted over integers, f is a function symbol, and $+, 1$ are interpreted in the standard way over the integers.

(a) Use the Nelson-Oppen decision procedure to determine the satisfiability of the above formula. Use the decision procedures for the theory of uninterpreted functions and the theory of arrays, and use simple mathematical reasoning for deriving new equalities among the constants in the theory of linear integer arithmetic. If the formula is satisfiable, give an interpretation that satisfies the formula.

Solution: The input formula is expressed in the combined theories of $\mathcal{T}_E$, $\mathcal{T}_A$ and the theory $\mathcal{T}_Z$ of linear integer arithmetic. The formula contains a clause consisting of two literals. Therefore, to decide satisfiability of the input formula, we use DPLL($\mathcal{T}_E \cup \mathcal{T}_A \cup \mathcal{T}_Z$).

We first name every atom by a propositional variable:

$$\begin{array}{ll} p_1: & read(write(A, a, f(b)), c+1) = f(d) \\ p_2: & f(b) = f(d-1) \\ p_3: & b+1 = d \\ p_4: & c = a-1 \end{array}$$

Our input formula is thus represented by the following set S of propositional clauses:

$$\begin{array}{l} p_1 \\ \neg p_2 \\ p_3 \vee p_4 \end{array}$$

(DPLL$_1$) We apply DPLL on S. DPLL returns satisfiability of S and reports the set of literals:

$$p_1, \neg p_2, p_3, p_4$$

satisfying S.
The set T of $\mathcal{T}_E \cup \mathcal{T}_A \cup \mathcal{T}_Z$-theory literals corresponding to $p_1, \neg p_2, p_3, p_4$ are:

$$\begin{array}{l} read(write(A, a, f(b)), c+1) = f(d) \\ f(b) \neq f(d-1) \\ b+1 = d \\ c = a-1 \end{array}$$

We introduce the following new constants:

$$\begin{array}{lll} c_1 & \text{to denote} & f(b) \\ c_2 & \text{to denote} & c+1 \\ c_3 & \text{to denote} & f(d) \\ c_4 & \text{to denote} & d-1 \end{array}$$

By separating reasoning in the various theories, we should satisfy the following set of literals in two theories:
– In $\mathcal{T}_A$:

$$read(write(A, a, c_1), c_2) = c_3$$

– In $\mathcal{T}_E$:

$$\begin{array}{l} c_1 = f(b) \\ c_3 = f(d) \\ f(b) \neq f(c_4) \end{array}$$

– In $\mathcal{T}_Z$:

$$\begin{array}{l} c_2 = c+1 \\ c_4 = d-1 \\ b+1 = d \\ c = a-1 \end{array}$$

From the $\mathcal{T}_Z$-theory literals, the arithmetic reasoner derives the *shared equalities*:

$$c_4 = b \qquad \text{and} \qquad c_2 = a.$$

Using these equalities in conjunction with the $\mathcal{T}_E$-theory literals, the congruence closure algorithm of $\mathcal{T}_E$ will set the terms $f(b)$ and $f(c_4)$ in the same congruence class, yielding thus a contradiction in $\mathcal{T}_E$ (as $f(b) \neq f(c_4)$). Hence, we report *unsatisfiability of* T.

(DPLL$_2$) We next add $\neg p1 \vee p_2 \vee \neg p_3 \vee \neg p_4$ to the initial set S of propositional clauses and obtain the set S' of propositional clauses:

$$\begin{array}{l} p_1 \\ \neg p_2 \\ p_3 \vee p_4 \\ \neg p1 \vee p_2 \vee \neg p_3 \vee \neg p_4 \end{array}$$

DPLL reports that the literals:

$$p_1, \neg p_2, p_3, \neg p_4$$

satisfy S'. The set T' of $\mathcal{T}_Z \cup \mathcal{T}_E \cup \mathcal{T}_A$-theory literals corresponding to this model of S' is:

$$\begin{array}{l} read(write(A, a, f(b)), c + 1) = f(d) \\ f(b) \neq f(d - 1) \\ b + 1 = d \\ c \neq a - 1 \end{array}$$

Similarly as before, we apply the Nelson-Oppen decision procedure on the above set T' of literals and derive *unsatisfiability of* T' (by deriving a contradiction in $\mathcal{T}_E$).

(DPLL$_3$) We next add $\neg p1 \vee p_2 \vee \neg p_3 \vee p_4$ to the initial set S of propositional clauses and obtain the set S'' of propositional clauses:

$$\begin{array}{l} p_1 \\ \neg p_2 \\ p_3 \vee p_4 \\ \neg p1 \vee p_2 \vee \neg p_3 \vee \neg p_4 \\ \neg p1 \vee p_2 \vee \neg p_3 \vee p_4 \end{array}$$

DPLL reports that the literals:

$$p_1, \neg p_2, \neg p_3, p_4$$

satisfy S''. The set T'' of $\mathcal{T}_Z \cup \mathcal{T}_E \cup \mathcal{T}_A$-theory literals corresponding to this model of S' is:

$$\begin{array}{l} read(write(A, a, f(b)), c + 1) = f(d) \\ f(b) \neq f(d - 1) \\ b + 1 \neq d \\ c = a - 1 \end{array}$$

Similarly as before, we apply the Nelson-Oppen decision procedure on the above set T'' of literals.

We introduce the following new constants:

$$\begin{array}{lll} c_1 & \text{to denote} & f(b) \\ c_2 & \text{to denote} & c+1 \\ c_3 & \text{to denote} & f(d) \\ c_4 & \text{to denote} & d-1 \end{array}$$

By separating reasoning in the various theories, we should satisfy the following set of literals in two theories:

– In $\mathcal{T}_A$:

$$read(write(A, a, c_1), c_2) = c_3$$

– In $\mathcal{T}_E$:

$$\begin{array}{l} c_1 = f(b) \\ c_3 = f(d) \\ f(b) \neq f(c_4) \end{array}$$

– In $\mathcal{T}_Z$:

$$\begin{array}{l} c_2 = c+1 \\ c_4 = d-1 \\ b+1 \neq d \\ c = a-1 \end{array}$$

From the $\mathcal{T}_Z$-theory literals, the arithmetic reasoner derives a *shared equality*:

$$c_2 = a.$$

Using this equality in conjunction with the $\mathcal{T}_A$-theory literals, the congruence closure algorithm of $\mathcal{T}_A$ derives a *shared equality*:

$$c_1 = c_3.$$

No further shared equalities are derived in either of the theories, hence T'' *is satisfiable, implying that our input formula is satisfiable*.

We construct a model I of T'' as follows.

Let Z be a sort and we consider the domain Int of integer numbers. We consider array indexes and array elements to be of sort Z. We interpret the sort Z to be the domain Int, that is $Z^I := Int$. We next interpret arrays as functions on $Int \to Int$; that is for an array A, we have $A^I : Int \to Int$.

We set $f : Z \to Z$ and interpret $f^I : Int \to Int$. We consider $a, b, c, d, c_1, c_2, c_3, c_4$ to be of sort Z. We define:

$$\begin{array}{l} a^I := 0 \\ b^I := 1 \\ c^I := -1 \\ d^I := 3 \\ c_1^I := 1 \\ c_2^I = 0 \qquad\qquad \text{as } c_2^I = a^I \\ c_3^I = 1 \\ c_4^I = 2 \\ f^I(u) = \begin{cases} 1, \text{if } u \in \{1, 3\} & \text{as } c_1^I = f^I(b^I) \text{ and } c_3^I = f^I(d^I) \\ 2, \text{if } u = 2 & \text{as } f^I(c_4^I) = f^I(d^I - 1) \\ 3, \text{otherwise} \end{cases} \\ A^I(u) := 1, \text{ for all } u \in Int. \end{array}$$

This interpretation I is a model of T''.

The interpretation I yields a model of our original formula, by "omitting" the interpretations of c_1, c_2, c_3, c_4 from I.

(b) Encode the above formula as an input to the Z3 SMT solver and evaluate Z3 on your encoding, using `http://rise4fun.com/Z3`. Interpret the result of Z3. Provide the electronic version of your Z3 encoding together with your solution.