COMP 60332

Two hours

Note that the last five pages of the exam paper include definitions from the course

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Automated Reasoning and Verification

Date: Wednesday 15th May 2019

Time: 09:45 - 11:45

Please answer BOTH Questions
Each Question is worth 30 marks

Use a SEPARATE answerbook for each QUESTION

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

1. Orderings, propositional logic, DPLL, state transition systems, LTL.

(a) Which of the following ordered sets are well-founded ?

i. $(\mathbb{N}, >)$,

ii. $(\mathbb{Z}, >)$,

iii. $(\mathbb{Q}^{[1;0]}, >)$ – the set of all rationals in the interval $[1;0]$,

iv. the set of all propositional clauses in the subsumption relation, i.e., $C > D$ if and only if $D \subset C$.

Briefly explain your answer. (4 marks)

(b) Is the following statement true or false? ”There exist a non-empty well-founded ordering without a minimal element.” Briefly explain your answer.

(2 marks)

(c) Briefly explain how validity and equivalence of propositional formulas can be expressed in terms of satisfiability. (2 marks)

(d) What are two main differences between structural CNF transformation and syntactic CNF transformation (based on equivalence rules)? (2 marks)

(e) Apply DPLL to the set of clauses S below, assuming the following sequence of decision literals:

$$\neg q^d \ldots \neg m^d \ldots v^d \ldots$$

Apply backjumping (BJ) and lemma learning (LL) whenever possible and explain how you applied them. Is this set of clauses satisfiable? (Note, the formal definition of DPLL rules is given at the end of the exam paper.)

(10 marks)

$\Vert$
$p \vee m \vee u$
$q \vee \neg p$
$\neg v \vee p \vee \neg s$
$s \vee \neg v \vee p$

(f) Is LTL formula $\Diamond\Box F$ equivalent to one of the following formulas ?

i. $\Box(F \rightarrow \bigcirc F)$

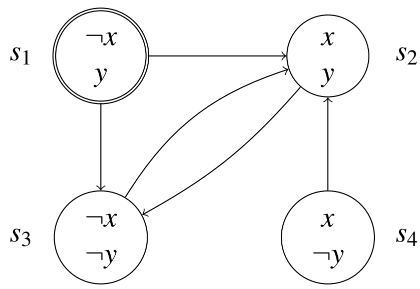
ii. $\Diamond F \wedge \Box(F \rightarrow \bigcirc F)$

Briefly explain your answer in each case. (4 marks)

(g) Consider the transition system with the state transition graph shown below.

i Write down 1) the set of all states reachable from the inital state and 2) a symbolic representation of this set. (2 marks)

ii Write down 1) the set of all single step transitions starting from the initial state and 2) a symbolic representation of this transition set. (2 marks)



(h) Explain the main advantage of k-induction compared to bounded model checking. (2 marks)

2. First-order logic, resolution, redundancy, model construction, orderings.

   (a) Consider a first-order language with the predicate symbols Bur, Bal, C, S and W, and a supply of variables $x, y, z, \ldots$. Suppose the predicate symbols have the following interpretation.

$Bur(x)$	means that x is a burglar
$Bal(x)$	means that x is a balaclava
$C(x)$	means that x is a car
$S(x,y)$	means that x steals y
$W(x,y)$	means that x wears y

   Express each of the following sentences as a first-order formula. (4 marks)

   i) Everyone who steals is a burglar.

   ii) A burglar wearing a balaclava steals a car.

   iii) Not every burglar wears a balaclava.

   (b) For each of the following statements state whether it is true or false. In each case explain your answer. (8 marks)

   i) The transformation of first-order logic formulas to clausal form is not unique (i.e., for the same formula it can produce different sets of clauses).

   ii) It is possible to find an interpretation that satisfies the set

$$N = \{P(x), \neg P(f(y))\}$$

   of clauses. If true, give one. If false, say why.

   iii) Every Herbrand model of $P(a)$ is a Herbrand model of $\neg P(g(x)) \lor P(x)$.

   iv) The substitution $\sigma = \{x/f(a,a), y/a, u/f(a,a)\}$ is a unifier of the atoms

$$Q(x, f(y,a)) \quad \text{and} \quad Q(u,u).$$

(c) i) Find a total ordering $\succ$ on the ground atoms A_1, A_2, A_3, A_4, A_5, such that the associated clause ordering $\succ_C$ orders the following clauses like this:

$$
\begin{array}{llr}
 & \neg A_2 \vee A_1 & (C_1) \\
\succ_C & \neg A_4 \vee \neg A_2 \vee \neg A_2 \vee A_3 & (C_2) \\
\succ_C & A_3 \vee \neg A_3 & (C_3) \\
\succ_C & A_3 \vee \neg A_5 & (C_4) \\
\succ_C & A_5 \vee \neg A_4 & (C_5) \\
\succ_C & \neg A_5 & (C_6)
\end{array}
$$

Justify your answer. Note that the definition of $\succ_C$ is given at the end of the exam paper. (3 marks)

ii) Let $N = \{C_1, \ldots, C_6\}$ be the set of clauses in 2(c)i. State which of the clauses in N are redundant with respect to $\succ_C$. Justify why these clauses are redundant in N. (3 marks)

iii) Find the candidate model $I_N^{\succ}$ for the set $N = \{C_1, \ldots, C_6\}$ (and nothing else). Briefly explain your answer. (2 marks)

(d) Let N be the following set of clauses.

1. $\neg P(x,y) \vee \neg P(y,z) \vee Q(z) \vee Q(x)$
2. $P(x, f(x))$
3. $\neg Q(f(f(x))) \vee \neg Q(x)$

Let $\succ$ be a total and well-founded ordering on ground atoms such that:

if the atom A contains more symbols than B, then $A \succ B$.

Let S be the selection function which selects one negative literal among the literals containing the deepest term in all clauses which contain negative literals.

i) Use $Res_S^{\succ}$ (where $\succ$ and S are as specified) to derive the empty clause or obtain a set of clauses saturated up to redundancy and justify each step. Note that the definition of $Res_S^{\succ}$ is given at the end of the exam paper. (4 marks)

ii) In your derivation indicate both the maximal literals and selected literals in every clause. (4 marks)

iii) In your derivation also indicate which of the clauses (if any) are redundant, and why. (1 mark)

iv) Is N satisfiable, or not? Justify your answer. (1 mark)

Examination definition sheet

Structural transformation. Lemma: $F[G]$ is satisfiable $\Leftrightarrow F[n_G] \wedge (n_G \leftrightarrow G)$ is satisfiable. provided n_G is a (fresh) propositional variable not occurring in $F[G]$. n_G can be seen as a name for G. *Structural CNF Transformation:* introduce names recursively for every non-literal subformula in the original formula.

DPLL rules.

Unit Propagate (UP):

$$U \parallel S, \;\; \Rightarrow_{\mathrm{UP}} \;\; U\ell \parallel S \qquad \text{if } \begin{cases} I_U \models \neg C, \text{ for } C \vee \ell \in S \\ \ell \text{ is undefined in } I_U \end{cases}$$

Decide (D):

$$U \parallel S \;\; \Rightarrow_D \;\; U\ell^d \parallel S \qquad \text{if } \{ \; \ell \text{ is undefined in } I_U$$

Backtrack (B)

$$U\ell^d V \parallel S \;\; \Rightarrow_B \;\; U\bar{\ell} \parallel S \qquad \text{if } \begin{cases} I_{U\ell^d V} \models \neg C, \text{ for } C \in S, \\ V \text{ contains no decision literals} \end{cases}$$

Unsat ($\bot$)

$$U \parallel S \;\; \Rightarrow_\bot \;\; \bot \parallel S \qquad \text{if } \begin{cases} I_U \models \neg C, \text{ for } C \in S, \\ U \text{ contains no decision literals} \end{cases}$$

Backjumping (BJ)

$$U\ell^d V \parallel S \;\; \Rightarrow_{BJ} \;\; Ue \parallel S \qquad \text{if } \begin{cases} I_{U\ell^d V} \models \neg C, \text{ for } C \in S, \\ U \wedge S \models e, \\ \text{e is undefined in } U. \end{cases}$$

Lemma Learning (LL):

$$U \parallel S \;\; \Rightarrow_{LL} \;\; U \parallel S \cup \{C\} \qquad \text{if } \begin{cases} S \models C \\ C \text{ is set-reduced} \end{cases}$$

LTL semantics.

Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula. F is true on π, denoted by $\pi \models F$, defined by induction on F as follows. For all $i = 0, 1, \ldots$ denote by π_i the sequence of states $s_i, s_{i+1}, s_{i+2} \ldots$ (note that $\pi_0 = \pi$).

- $\pi \models \top$ and $\pi \not\models \bot$.
- $\pi \models p$ if $s_0 \models p$.
- $\pi \models F_1 \wedge \ldots \wedge F_n$ if for all $j = 1, \ldots, n$ we have $\pi \models F_j$;
- $\pi \models F_1 \vee \ldots \vee F_n$ if for some $j = 1, \ldots, n$ we have $\pi \models F_j$.
- $\pi \models \neg F$ if $\pi \not\models F$.
- $\pi \models F \rightarrow G$ if either $\pi \not\models F$ or $\pi \models G$;
- $\pi \models F \leftrightarrow G$ if either both $\pi \not\models F$ and $\pi \not\models G$ or both $\pi \models F$ and $\pi \models G$.
- $\pi \models \bigcirc F$ if $\pi_1 \models F$;
- $\pi \models \Diamond F$ if for some $i = 0, 1, \ldots$ we have $\pi_i \models F$;
- $\pi \models \Box F$ if for all $i = 0, 1, \ldots$ we have $\pi_i \models F$.
- $\pi \models F \,\mathbf{U}\, G$ if for some $k = 0, 1, \ldots$ we have $\pi_k \models G$ and $\pi_0 \models F, \ldots, \pi_{k-1} \models F$.

Two LTL formulas F and G are called equivalent, denoted $F \equiv G$, if for every path π we have $\pi \models F$ if and only if $\pi \models G$.

Herbrand models. The *Herbrand universe* T_Σ (over Σ) is the set of all ground terms over Σ.

A *Herbrand interpretation* I (over Σ) is a set of ground atoms over Σ.

Truth in I of *ground formulae* is defined inductively by:

$$\begin{array}{rl}
I \models \top & \qquad\qquad I \not\models \bot \\
I \models A & \text{iff } A \in I, \text{ for any ground atom } A \\
I \models \neg F & \text{iff } I \not\models F \\
I \models F \wedge G & \text{iff } I \models F \text{ and } I \models G \\
I \models F \vee G & \text{iff } I \models F \text{ or } I \models G
\end{array}$$

Truth in I of any *quantifier-free formula* F with free variables $x_1, \dots, x_n$ is defined by:

$$I \models F(x_1, \dots, x_n) \text{ iff } I \models F(t_1, \dots, t_n), \text{ for every } t_i \in \mathrm{T}_\Sigma$$

Truth in I of any *set N of clauses* is defined by:

$$I \models N \quad \text{iff} \quad I \models C, \text{ for each } C \in N$$

Construction of candidate models. Let $N, \succ$ be given.

For all ground clauses C over the given signature, the sets I_C and Δ_C are inductively defined with respect to the clause ordering $\succ$ by:

$$I_C := \bigcup_{C \succ D} \Delta_D$$

$$\Delta_C := \begin{cases} \{A\}, & \text{if } C \in N, \; C = C' \vee A, \; A \succ C' \text{ and} \\ & I_C \not\models C \\ \emptyset, & \text{otherwise} \end{cases}$$

We say that C *produces* A, if $\Delta_C = \{A\}$.

The *candidate model* for N (wrt. $\succ$) is given as

$$I_N^{\succ} := \bigcup_{C \in N} \Delta_C.$$

We also simply write I_N, or I, for $I_N^{\succ}$, if $\succ$ is either irrelevant or known from the context.

Orderings. Let $(X, \succ)$ be an ordering. The *multi-set extension* $\succ_{\text{mul}}$ of $\succ$ to (finite) multi-sets over X is defined by

$$
\begin{aligned}
S_1 \succ_{\text{mul}} S_2 \ \text{ iff } \ & S_1 \neq S_2 \text{ and} \\
& \forall x \in X,\ \text{if } S_2(x) > S_1(x) \text{ then} \\
& \quad \exists y \in X:\ y \succ x \text{ and } S_1(y) > S_2(y)
\end{aligned}
$$

Suppose $\succ$ is a total and well-founded ordering on ground atoms. $\succ_L$ denotes the *ordering on ground literals* and is defined by:

$$
\begin{array}{lll}
[\neg]A & \succ_L & [\neg]B, \quad \text{if } A \succ B \\
\neg A & \succ_L & A
\end{array}
$$

$\succ_C$ denotes the *ordering on ground clauses* and is defined by the multi-set extension of $\succ_L$, i.e. $\succ_C = (\succ_L)_{\text{mul}}$.

Maximal literals. Let $\succ$ be a total and well-founded ordering on ground atoms.

A ground literal L is called *[strictly] maximal* wrt. a ground clause C iff

$$\text{for all } L' \text{ in } C: \quad L \succeq L' \quad [L \succ L'].$$

A non-ground literal L is *[strictly] maximal* wrt. a (ground or non-ground) clause C iff there exists a ground substitution σ such that

$$\text{for all } L' \text{ in } C: \quad L\sigma \succeq L'\sigma \quad [L\sigma \succ L'\sigma].$$

If L is [strictly] maximal wrt. a clause C then we say that L is *[strictly] maximal in* $L \vee C$.

The $\Rightarrow_U$-rules of the unification algorithm.

Orientation:	$t \doteq x, E$	$\Rightarrow_U$	$x \doteq t, E$ if $t \notin X$
Trivial:	$t \doteq t, E$	$\Rightarrow_U$	E
Disagreement/Clash:	$f(\ldots) \doteq g(\ldots), E$	$\Rightarrow_U$	$\perp$
Decomposition:	$f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n), E$	$\Rightarrow_U$	$s_1 \doteq t_1, \ldots, s_n \doteq t_n, E$
Occur-check:	$x \doteq t, E$	$\Rightarrow_U$	$\perp$ if $x \in var(t), x \neq t$
Substitution:	$x \doteq t, E$	$\Rightarrow_U$	$x \doteq t, E\{x/t\}$ if $x \in var(E), x \notin var(t)$

Ordered resolution with selection calculus $Res_S^{\succ}$. Let $\succ$ be an atom ordering and S a selection function.

Ordered resolution with selection rule:
$$\frac{C \vee A \qquad \neg B \vee D}{(C \vee D)\sigma}$$

provided $\sigma = \text{mgu}(A, B)$ and

(i) $A\sigma$ strictly maximal wrt. $C\sigma$;

(ii) nothing is selected in C by S;

(iii) either $\neg B$ is selected,
or else nothing is selected in $\neg B \vee D$ and $\neg B\sigma$ is maximal wrt. $D\sigma$.

Ordered factoring rule:
$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

provided $\sigma = \text{mgu}(A, B)$ and

(i) $A\sigma$ is maximal wrt. $C\sigma$ and

(ii) nothing is selected in C.

Redundancy. Let N be a set of ground clauses and C a ground clause. C is called *redundant* wrt. N, if there exist $C_1, \ldots, C_n \in N$, $n \geq 0$, such that

(i) all $C_i \prec C$, and

(ii) $C_1, \ldots, C_n \models C$.

A general clause is *redundant* wrt. N if each ground instance $C\sigma$ of C either belongs to $G_\Sigma(N)$ or is redundant wrt. $G_\Sigma(N)$.

N is called *saturated up to redundancy* (wrt. $Res_S^{\succ}$) iff every conclusion of an $Res_S^{\succ}$-inference with non-redundant clauses in N is in N or is redundant (i.e.

$$Res_S^{\succ}(N \setminus Red(N)) \subseteq N \cup Red(N),$$

where $Red(N)$ denotes the set of clauses redundant wrt. N).

END OF EXAMINATION