COMP 60332

Two hours

Examination definition sheet is available on pages 6 to 10 of this examination paper.

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Automated Reasoning and Verification

Date: Wednesday 31st May 2017

Time: 14:00 - 16:00

Please answer any THREE Questions from the FOUR Questions provided

Use a SEPARATE answerbook for each QUESTION.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

Answer any *three* of the four questions

Note that a definition sheet is included at the back of the exam paper.

1. (Orderings, structural CNF transformation, splitting)

   (a) Give an example of: i) a well-founded ordering; ii) a non well-founded ordering. Explain your answers. (2 marks)

   (b) Consider two orderings: $(X_1, \succ_1)$ and $(X_2, \succ_2)$.

      i. Define the lexicographic combination $\succ_{\text{lex}}$ of $\succ_1$ and $\succ_2$. (2 marks)

      ii. Show that if $\succ_1$ and $\succ_2$ are well-founded then $\succ_{\text{lex}}$ is also well-founded. (4 marks)

   (c) Show how to transform the following formula into clausal normal form using structural transformation:

   $$(\ldots((x_1 \rightarrow x_2) \rightarrow x_3) \rightarrow \ldots) \rightarrow x_k$$

   How many clauses the resulting clausal normal form contains? (7 marks)

   (d) Consider the following formula:

   $$\neg(p \leftrightarrow (q \vee \neg r)) \wedge (p \rightarrow \neg r) \wedge (q \rightarrow r)$$

   Apply the splitting algorithm to this formula and draw a splitting tree. Is this formula satisfiable? If this formula is satisfiable give a model of this formula. (5 marks)

2. (Propositional formalisation, DPLL, LTL, model checking)

   (a) Consider propositional variables $p_1, \ldots, p_n$. Express the following properties in propositional logic:

   i. at least two variables are true,
   ii. at most two variables are true,
   iii. exactly two variables are true.

   (3 marks)

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

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

   Apply backjumping (BJ) and lemma learning (LL) whenever possible. Write down explanations for backjumping and lemma learning. Is this set of clauses satisfiable? (10 marks)

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

   (c) Using LTL formulas express the following path properties:

   i. Always, if F holds then always after that $\neg F$ holds.
   ii. Starting from some state F always holds.
   iii. Sometime in the future F holds and until then $\neg G$ holds.
   iv. F holds in at least two states.
   v. $\neg F$ holds infinitely often.

   (5 marks)

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

3. (Translation from English to first-order logic, clausal form, unification, Herbrand interpretations)

   (a) Consider a first-order language with one unary predicate symbol S, three binary predicate symbols K, D, $\approx$, two constants a, b, and a supply of variables $x, y, z, \ldots$. Assume that

$S(x)$	means x is a sportscar	$x \approx y$	means x and y are identical
$K(x,y)$	means x knows y	a	means Adam
$D(x,y)$	means x drives y	b	means Ben

   Express each of the following sentences as formulas. (4 marks)

   i) Ben knows someone who drives a sportscar.
   ii) Not everyone knows someone who drives a sportscar.
   iii) The only one Adams knows who drives a sportscar is Ben.

   (b) i) Transform the following formula to clausal form.

   $$\neg\left[\exists x(P(x) \wedge \forall y(Q(y) \rightarrow S(x,y))\right]$$

   Justify every step in the transformation. (5 marks)

   ii) Is the formula satisfiable? Briefly explain why. (2 marks)

   (c) Consider the following atoms

   $$P(x, f(x,y)) \qquad P(g(a), z)$$

   and suppose $\sigma = \{x/g(a),\, y/a,\, z/f(a,a)\}$, where x, y, z denote variables.

   i) Is σ a unifier of the two atoms? Explain your answer. (2 marks)
   ii) Is σ a most general unifier of the two atoms? Explain your answer. (1 mark)
   iii) If your answer was no, apply our unification algorithm based on the $\Rightarrow_U$-rules to compute the most general unifier. (Note that the $\Rightarrow_U$-rules are given at the end of the exam paper.) (2 marks)

   (d) Assume the signature Σ is given by one constant a, one unary function symbol f and one unary predicate symbol P.

   Consider the following Herbrand interpretation.

   $$I = \{P(f(a)\}$$

   Determine if the following clauses hold in I. Briefly explain your answers. (4 marks)

   i) $I \models \neg P(a)$
   ii) $I \models P(f(a))$
   iii) $I \models \neg P(x) \vee P(f(x))$
   iv) $I \models \neg P(x) \wedge P(f(y))$

4. (Bookwork, model construction, orderings, ordered resolution, subsumption deletion)

   (a) Give a brief explanation of **two** of the following. (4 marks)

      i) first-order clause
      ii) Herbrand interpretation
      iii) refutationally complete
      iv) tautology deletion

   (b) Let N be the following set of ground clauses.

$$\begin{array}{rl} 1. & \neg Q \vee \neg Q \vee P(f(b)) \vee R \\ 2. & P(b) \vee P(a) \vee P(b) \\ 3. & Q \\ 4. & \neg P(a) \vee Q \\ 5. & \neg P(a) \vee \neg Q \vee P(f(a)) \\ 6. & \neg P(b) \end{array}$$

      i) Assume the ordering on atoms be defined by

$$P(f(b)) \succ P(f(a)) \succ P(b) \succ P(a) \succ R \succ Q.$$

      Sort the clauses in N with respect to $\succ_C$. (Note that $\succ_C$ is defined at the end of the exam paper.) (2 marks)

      ii) Construct the candidate model $I_N^{\succ}$ as described in lectures for the set N above (and nothing else). (4 marks)

   (c) 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 N be the following set of clauses.

$$\begin{array}{l} P(x, f(x)) \vee P(f(f(x)), x) \\ \neg P(x, y) \vee \neg P(f(x), z) \end{array}$$

   Use ordered resolution $Res^{\succ}$, where $\succ$ is an atom ordering defined as above (no literal is selected), to either derive the empty clause or obtain a saturated set of clauses.

   In your derivation indicate the maximal literals in every clause and justify each step. (6 marks)

   (d) For each of the following statements state whether it is true or false. In each case give a brief explanation. (4 marks)

      i) $C = P(x, z) \vee \neg R(z, b)$ subsumes $D = P(f(y), y) \vee \neg R(y, b) \vee P(y, y)$
      ii) $C = P(x, z) \vee \neg R(z, b)$ subsumes $D' = P(f(y), y) \vee \neg R(a, b) \vee P(y, y)$

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):

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

Decide (D):

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

Backtrack (B)

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

Unsat ($\perp$)

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

Backjumping (BJ)

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

Lemma Learning (LL):

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

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 $s_0 \models p$.
- $\pi \models F_1 \land \ldots \land F_n$ if for all $j = 1, \ldots, n$ we have $\pi \models F_j$;
- $\pi \models F_1 \lor \ldots \lor F_n$ 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 $x_1, \ldots, x_n$ is defined by:

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

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:

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

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:	$f(\ldots) \doteq g(\ldots), E \Rightarrow_U \bot$	
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:	$x \doteq t, E \Rightarrow_U \bot$	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 $\sigma = \mathrm{mgu}(A, B)$ 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 $\sigma = \mathrm{mgu}(A, B)$ 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