Exam – Computational Logic - Subjects -2013-2014

I Propositional logic

1. Using a proof method:
   a) semantic method (truth table, semantic tableau, conjunctive normal form)
   b) syntactic method (resolution, definition of deduction, the theorem of deduction and its reverse)
   c) direct method (truth table, conjunctive normal form, definition of deduction, the theorem of deduction and its reverse)
   d) refutation method (resolution, semantic tableau)

   prove the validity of some propositional formulas:
   - A2 – the second axiom of propositional logic
   - A3- the third axiom, „modul tollens"
   - the syllogism rule
   - the permutation/ reunion/ separation of the premises law

2. Check the following logical/syntactic consequence:

   $U_1, \ldots, U_n \models V$ ($\vdash$)
   - build the deduction of V from the hypothesis $U_1, \ldots, U_n$ using the axiomatic system;
   - semantic tableau for: $U_1 \wedge \ldots \wedge U_n \wedge \neg V$;
   - resolution for: $FNC(U_1) \wedge \ldots \wedge FNC(U_n) \wedge FNC(\neg V)$.

3. Decide the type (consistent, contingent, inconsistent, tautology) of the propositional formula U and write the models and anti-models of U.
   - from the truth table of U;
   - from the semantic tableau of U => the models of U are provided by the open branches
   - from the semantic tableau of $\neg$ U => the anti-models of U are provided by the open branches
   - from the conjunctive normal form of U => the anti-models of U are provided by the clauses that are not tautologies
   - from the conjunctive normal form of U => the models of U are provided by the cubes that are not inconsistent

4. Prove the inconsistency of a set of clauses using:
   - general resolution + transformations used to simplify the initial set of clauses
   - level saturation strategy

- lock resolution
- linear resolution('unit' / 'input')

5. Check the consistency/inconsistency of a set of clauses using:
   - level saturation strategy
   - lock resolution + level saturation strategy
   - linear resolution - backtracking.

6. The theorems of soundness and completeness of the proof methods:

   The properties of propositional logic: coherence, non-contradiction ,decidability.

   The theorem of soundness for propositional logic:

   If $\vdash U$ **then** $\models U$ (a theorem is a tautology).

   The theorem of completeness for propositional logic:

   If $\models U$ **then** $\vdash U$ (a tautology is a theorem).

   The theorem of deduction and its reverse.

7. Definitions: tautology, theorem, logical consequence, syntactic consequence, logical equivalence, consistent/contingent/valid/inconsistent formula, interpretation, model, anti-model.

   The axiomatic system of propositional logic.

   The axiomatic system of propositional resolution.

8. Propositional reasoning modeling

II First-order (predicate) logic

1. Evaluation of a closed predicate formula under a given (proposed by the student) interpretation, with a finite/infinite domain..
2. Build a model/ anti-model of a closed predicate formula:
   - from the semantic tableau of U => the models of U are provided by the open branches
   - from the semantic tableau of $\neg$ U => the anti-models of U are provided by the open branches
   - a proposed interpretation that evaluates the formula U as true/false is a model/anti-model of U.
3. Check the property of distributivity of a quantifier ($\exists\forall$) over a connective ($\wedge, \vee, \rightarrow, \leftrightarrow$):

Ex: distributivity of „$\exists$" over „$\rightarrow$":

$(\exists x)(A(x) \rightarrow B(x)) \equiv (\exists x)A(x) \rightarrow (\exists x)B(x)$ if and only if

$\models (\exists x)(A(x) \rightarrow B(x)) \leftrightarrow ((\exists x)A(x) \rightarrow (\exists x)B(x))$ if and only if

$\models (\exists x)(A(x) \rightarrow B(x)) \rightarrow ((\exists x)A(x) \rightarrow (\exists x)B(x))$ and $\models ((\exists x)A(x) \rightarrow (\exists x)B(x)) \rightarrow (\exists x)(A(x) \rightarrow B(x))$

4. Using a proof method:
   a) semantic method (semantic tableau)
   b) syntactic method (resolution, definition of deduction, the theorem of deduction and its reverse)
   c) direct method (definition of deduction, the theorem of deduction and its reverse)
   d) refutation method (resolution, semantic tableau)

   prove that some predicate formulas are tautologies/theorems
5. Tranform a predicate formula into prenex, Skolem and clausal normal forms.
6. Check the following logical/syntactic consequence:

   $U_1, \ldots, U_n \models V$ ($\vdash$)

   - build the deduction of V from the hypothesis $U_1, \ldots, U_n$ using the axiomatic system;
   - semantic tableau for: $U_1 \wedge \ldots \wedge U_n \wedge \neg V$;
   - resolution for: $U_1^C \wedge \ldots \wedge U_n^C \wedge (\neg V)^C$.

7. Definitions: substitutions, the most general unifier of 2 atoms - algorithm.

8. Prove the inconsistency of a set of predicate clauses using:
   - general resolution
   - level saturation strategy
   - lock resolution
   - linear resolution('unit' or 'input')

9. The theorems of soundness and completeness of the proof methods:

   The properties of propositional logic: coherence, non-contradiction, semi-decidability (Church).

   The theorem of soundness for first-order logic:

   If $\vdash U$ **then** $\models U$ (a theorem is a tautology).

   The theorem of completeness for first-order logic:

   If $\models U$ **then** $\vdash U$ (a tautology is a theorem).

   The theorem of deduction and its reverse.

10. Definitions: tautology, theorem, logical consequence, syntactic consequence, logical equivalence, consistent/contingent/valid/inconsistent formula, interpretation, model, anti-model.

    The axiomatic system of first-order logic.

    The axiomatic system of first-order resolution.

11. Transformation of a natural language sentence into a predicate formula.
   Predicate reasoning modeling.

III Boolean algebras, Boolean functions, logical circuits

1. Boolean algebra: definition+examples
   Using “nand”/”nor” express the operations “and”, “not”, “or”.
   Definitions: Boolean function, “minterm”,“maxterm”, „factorization”, ”maximal monom”, „central monom”, „simplification of a Boolean function”.
2. Build the canonical conjunctive/disjunctive form of a Boolean function (of 2,3,4 variables) given by its table of values.
   Exemples of minterms and maxterms (of 2,3,4 variabiles): notations,expressions, tables of values.
3. Simplification of Boolean functions of 2, 3, 4 variables using Quine’s method or Veitch/Karnaugh diagrams.
   A Boolean function can be given:
   - in canonical disjunctive form using the standard notations for the minterms:
     $$f(x_1,x_2,x_3)= m_0 \vee m_3 \vee m_4 \vee m_5 \vee m_6 \vee m_7;$$
   - in canonical disjunctive form using the expressions for the minterms:
     $$f(x_1,x_2,x_3,x_4)= x_1x_2\bar{x}_3x_4 \vee x_1x_2x_3\bar{x}_4 \vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$$
     $$\vee \bar{x}_1\bar{x}_2x_3\bar{x}_4 \vee x_1\bar{x}_2x_3x_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3x_4 \vee \bar{x}_1\bar{x}_2x_3x_4;$$
   - by an expression:
     $$f(x_1, x_2, x_3) = x_3(\bar{x}_1 \vee x_2) \vee x_1(x_2 \vee \bar{x}_2\bar{x}_3) \vee \bar{x}_1\bar{x}_2\bar{x}_3,$$
     or
     $$f(x, y, z) = x(\bar{y} \oplus z) \vee y(\bar{x} \oplus z) \vee \bar{x}(\bar{y} \downarrow z) \vee (\bar{x} \downarrow y)z;$$
     -apply transformations (distributivity, replace ↓, ⊕,...) to obtain the canonical disjunctive form

   - by its table of values,

x	*y*	*z*	*f*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

     - from the values 1 of the function the canonical disjunctive form is built
   - by its values 1:
     $f_1(1,1,1,1) = f_1(1,1,0,1) = f_1(0,1,1,1) = f_1(1,1,0,0) = f_1(0,1,0,0) = f_1(0,0,0,0)=$

$=f_1(0,0,0,1) = f_1(0,0,1,1) =1;$

- the canonical disjunctive form is built

- by its values 0: $f_1(0,1,0) = f_1(0,1,1) = f_1(1,0,1) = 0,$
  - from the values 1 of the function the canonical disjunctive form is built

4. Using basic and derived gates draw the logical circuit corresponding to a Boolean function given by a Boolean expression.
   Write the expression of the Boolean function which models the functionality of a logical circuit with basic and derived gates.

5. Examples of logical circuits used in the hardware: the „coder”, the „decoder”, the „comparison circuit”, the „addition circuit”.