Exam – Computational Logic - Subjects -2023-2024

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
   - the distributivity of a connective over another connective

2. Check the following logical/syntactic consequence:
   $U_1, \dots, U_n \models V$ ($\vdash$)
   - build the deduction of V from the hypothesis $U_1, \dots, U_n$ using the axiomatic system and the definition of deduction;
   - semantic tableau for: $U_1 \wedge \dots \wedge U_n \wedge \neg V$;
   - resolution for: CNF($U_1 \wedge \dots \wedge U_n \wedge \neg V$).

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;
   - the models of U are provided by the open branches of the semantic tableau of U;
   - the anti-models of U are provided by the open branches of the semantic tableau of $\neg$U
   - the anti-models of U are provided by the clauses of CNF(U) which are not tautologies
   - the models of U are provided by the cubes of DNF(U) which 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.

Lecture 4 slide 8

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:
   - the models of U are provided by the open branches of the semantic tableau of U
   - the anti-models of U are provided by the open branches of the semantic tableau of $\neg$U
   - a proposed interpretation which evaluates the formula U as true/false is a model/ anti-model of U.

Lecture 5 slide 24

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 tableaux method)
   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 tableaux method)

   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:

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

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

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, logic 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 conjunctive/disjunctive canonical form of a Boolean function (of 2,3,4 variables) given by its table of values.
   Examples 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, Veitch/Karnaugh diagrams, Moisil’s method. A Boolean function is given:
   - in disjunctive canonical form (DCF) 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 disjunctive canonical form (DCF) using the expressions for the minterms:
     $$f(x_1,x_2,x_3,x_4)= x_1x_2\overline{x_3}x_4 \vee x_1x_2x_3\overline{x_4} \vee x_1x_2\overline{x_3}\,\overline{x_4} \vee x_1\overline{x_2}x_3\overline{x_4} \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}$$
     $$\vee\ \overline{x_1}\,\overline{x_2}x_3\overline{x_4} \vee x_1\overline{x_2}x_3x_4 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4 \vee \overline{x_1}\,\overline{x_2}x_3x_4;$$
   - by an expression:
     $$f(x_1, x_2, x_3) = x_3(\overline{x}_1 \vee x_2) \vee x_1(x_2 \vee \overline{x}_2\overline{x}_3) \vee \overline{x}_1\overline{x}_2\overline{x}_3,$$
     or
     $$f(x, y, z) = x(\overline{y} \oplus z) \vee y(\overline{x} \oplus z) \vee \overline{x}(\overline{y} \downarrow z) \vee (\overline{x} \downarrow y)z\ ;$$
     -apply transformations (distributivity, replace ↓, ⊕,...) to obtain the DCF
   - 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 DCF is built
   - by its values 1:
     $f_1(1,1,1,1) = f_1(1,1,0,1) = f_1(0,1,1,1) = f_1(1,1,0,0) = f_1(0,1,0,0) = f_1(0,0,0,0)=$
     $=f_1(0,0,0,1) = f_1(0,0,1,1) =1;$ DCF 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 DCF is built

Simplification of Boolean functions of 2, 3, 4 variables, given in CCF, using Veitch/Karnaugh diagrams and a dual simplification algorithm.

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

5. Examples of logic circuits used in the hardware: *encode*r, *decoder, comparator*, *adder*, *subtractor*.
   The combinational logic circuit corresponding to the *electronic display of the decimal digits using 7 segments (LEDs)*.