# Lógica en la Informática / Logic in Computer Science
# January 16th, 2019. Time: 2h30min. No books or lecture notes.

**Note on evaluation:** eval(propositional logic) = max{ eval(Problems 1,2,3,4), eval(partial exam) }.
eval(first-order logic) = eval(Problems 5,6,7).

**1a)** Let $F$ and $G$ be propositional tautologies. Is it true that, for every propositional formula $H$, we have $H \models F \wedge G$? Prove it using only the definitions of propositional logic.

**Answer:**

| | |
|---|---|
| $F$ and $G$ propositional tautologies $\Longrightarrow$ | by definition of tautology |
| forall $I$, $\quad I \models F$ and $I \models G \Longrightarrow$ | by definition of $\models$ |
| forall $I$, $\quad eval_I(F) = 1$ and $eval_I(G) = 1 \Longrightarrow$ | by definition of min |
| forall $I$, $\quad min(eval_I(F), eval_I(G)) = 1 \Longrightarrow$ | by definition of $eval_I(\wedge)$ |
| forall $I$, $\quad eval_I(F \wedge G) = 1 \Longrightarrow$ | by definition of $\models$ |
| forall $I$, $\quad I \models F \wedge G \Longrightarrow$ | from the meaning of if-then |
| forall $I$, $\quad$ if $I \models H$ then $I \models F \wedge G \Longrightarrow$ | by definition of logical consequence |
| $H \models F \wedge G$ | |

**1b)** Is it true that the formula $p$ is a logical consequence of the set $S$ of three clauses $\{\ p \vee q \vee r,\quad \neg q \vee r,\quad \neg r\ \}$? Prove it in the simplest and shortest way you know. You may use any well-known property of propositional logic, even without proving that property.

**Answer:** yes. We know that resolution is correct. So if by resolution from $p \vee q \vee r$ and $\neg q \vee r$ we obtain the clause $p \vee r$, then $S \models p \vee r$. In fact, therefore $S \equiv S \cup \{p \vee r\}$. Similarly, from $p \vee r$ and $\neg r$ we obtain the clause $p$.

**2)** Let $Res(S)$ denote the closure under resolution of a set $S$ of propositional two-literal clauses. Which three properties of $Res(S)$ do you find essential to prove that 2-SAT is polynomial? Answer in three lines like this:
1. ...
2. ...
3. ...

**Answer:**
1. $Res(S)$ only contains 2-literal clauses (cannot get larger clauses by resolution from 2-literal clauses).
2. Only a quadratic number of 2-literal clauses exist, so $|Res(S)|$ is quadratic and can be computed in polynomial time.
3. $S$ insat iff empty clause in $Res(S)$.

**3)** Given a propositional CNF, that is, a set of propositional clauses $S$, explain in two lines your best method to decide wether $S$ is a tautology.

**Answer:** $S$ is a tautology iff all clauses $C$ in $S$ are tautologies. A clause is a tautology iff it contains some predicate symbol $p$ and its negation $\neg p$. So the best method is to check this: linear time.

**4)** Write the clauses needed for expressing $x_1 + \ldots + x_4 \leq 1$ using the ladder encoding. (Please write them in a clean and ordered way; give no explanations.)

**Answer:**

| | | |
|---|---|---|
| $\neg x_1 \vee a_1$ | $\neg a_1 \vee \neg x_2$ | $\neg a_1 \vee a_2$ |
| $\neg x_2 \vee a_2$ | $\neg a_2 \vee \neg x_3$ | $\neg a_2 \vee a_3$ |
| $\neg x_3 \vee a_3$ | $\neg a_3 \vee \neg x_4$ | |

**5)** Let $F$ be the folowing formula of first-order logic with equality:
$\forall x\,\forall y\,\forall z\; f(x, f(y,z)) = f(f(x,y), z) \;\; \wedge \;\; \forall x\; f(e,x) = x \;\; \wedge \;\; \forall x\; f(i(x), x) = e \;\; \wedge \;\; \forall x\,\forall y\; f(x,y) = f(y,x)$.
Any model of $F$ is called a *conmutative group* (where $e$ is the *neutral element* for $f$ and $i$ its *inverse*).
**5a)** Give a well-known example of a conmutative group with *infinite* domain. Please write it as clean and simple as possible; give no explanations.

**Answer:** $D_I$ is the integers, $f_I(n,m) = n + m$ (the addition of integers), $i_I(n) = -n$, and $e_I = 0$.

**5b)** Give an *as simple as possible* example of a conmutative group with a *finite* domain. Please write it as clean and simple as possible; give no explanations.

**Answer:** $D_I = \{a\}$. Then the functions can only be: $f_I(a,a) = a$, $i_I(a) = a$, and $e_I = a$.

**6)** Formalize and prove by resolution that sentence $D$ is a logical consequence of the other three:
$A$: Everybody loves his father and his mother.
$B$: John is stupìd.
$C$: When someone is stupid, at least one of his parents is stupid too.
$D$: There are stupid people that are loved by someone.
Mandatory: use function symbols $f(x)$ and $m(x)$ meaning "father of $x$" and "mother of $x$".

**Answer:**

| | |
|---|---|
| $A$: | $\forall x\; Loves(x, f(x)) \wedge Loves(x, m(x))$ |
| $B$: | $IsStupid(John)$ |
| $C$: | $\forall x\; IsStupid(x) \rightarrow (IsStupid(f(x)) \vee IsStupid(m(x)))$ |
| $\neg D$: | $\neg(\exists x\,\exists y\; IsStupid(x) \wedge Loves(y,x))$ |

In clausal form, these become:

| | |
|---|---|
| $A1.$ | $Loves(x, f(x))$ |
| $A2.$ | $Loves(x, m(x))$ |
| $B.$ | $IsStupid(John)$ |
| $C.$ | $\neg IsStupid(x) \vee IsStupid(f(x)) \vee IsStupid(m(x))$ |
| $\neg D.$ | $\neg IsStupid(x) \vee \neg Loves(y,x)$ |

By resolution we obtain:

| | | | |
|---|---|---|---|
| 6. | $IsStupid(f(John)) \vee IsStupid(m(John))$ | $B + C$, | $\sigma = \{x = John\}$ |
| 7. | $\neg Loves(y, f(John)) \vee IsStupid(m(John))$ | $\neg D + 6$, | $\sigma = \{x = f(John)\}$ |
| 8. | $\neg Loves(y, f(John)) \vee \neg Loves(y', m(John))$ | $\neg D + 7$, | $\sigma = \{x = m(John)\}$ |
| 9. | $\neg Loves(y', m(John))$ | $A1 + 8$, | $\sigma = \{y = John, x = John\}$ |
| 10. | `empty clause` | $A2 + 9$, | $\sigma = \{y' = John, x = John\}$ |

**7)** Consider a 1-ary function symbol $f$ and a 3-ary predicate symbol $P$ and a first-order interpretation $I$ with a finite domain $D_I = \{a, b\}$ and the (finite) definition of the functions $f_I$ and $P_I$. Answer in a few words: Is it decidable whether $I$ satisfies a given formula $F$ (over $f$ and $P$)? If so, what do you think is the complexity of this? (hint: any relationship with 3-SAT?).

**Answer:**
Yes, this is decidable: evaluating a given $F$ in a given first order interpretation $I$ is obviously a finite process if $D_I$ is finite.

About the complexity: it is NP-hard even for this simple set of symbols. Let $I$ be the interpretation where $D_I = \{a, b\}$, $f_I(a) = b$, $f_I(b) = a$, and $P_I(x,y,z) = 1$ iff at least one of its arguments is $a$. Then we can express 3-SAT as a problem of checking $I \models F$, in the following way:

$(\overline{x_7} \vee x_8 \vee \overline{x_2}) \;\wedge\; \ldots$ is satisfiable IFF $I \models \exists x_1 \exists x_2 \ldots \exists x_n \;\; P(f(x_7), x_8, f(x_2)) \;\wedge\; \ldots$

Hence checking $I \models F$ cannot be easier than 3-SAT. Here $a$ and $b$ act as true and false, $f_I$ as negation and $P_I$ says if a clause is true.
Note: in fact checking $I \models F$ is P-space-complete, i.e., it is believed to be even harder than NP-complete problems.