Computational Logic Lab II - Solutions to Selected Exercises

Camilo Thorne Sergio Tessaris

11/3/2011

Normal Forms. We will show **(1.iii.b)** and **(1.iv)**. We will consider w.l.o.g. that φ is in negation normal form (this transformation is polynomial!). We also recall that the *size* $|\varphi|$ of a propositional formula φ is the number of symbols it contains (with the exception of parenthesis).

(1.iii.b) For this exercise, we need to define $\mathit{defi}(\cdot)$ by (structural) recursion on φ, namely,

$$\begin{aligned} \mathit{defi}(a) &:= a, \\ \mathit{defi}(\neg a) &:= \neg a, \\ \mathit{defi}(\varphi \wedge \psi) &:= (a_\varphi \Leftrightarrow \varphi) \wedge (a_\varphi \wedge \mathit{defi}(\psi[a_\varphi/\varphi])), \\ \mathit{defi}(\varphi \vee \psi) &:= (a_\varphi \Leftrightarrow \varphi) \wedge (a_\varphi \vee \mathit{defi}(\psi[a_\varphi/\varphi])), \end{aligned}$$

where $\psi[a_\varphi/\varphi]$ denotes the (uniform) substitution of all the occurrences of (sub)formula φ in ψ by the "fresh" atom a_φ[1].

To show that $\mathit{defi}(\varphi) \models \varphi$ we need to prove that, for all assignments μ, if satisfies $\mathit{defi}(\varphi)$, it satisfies φ too. Let μ be a satisfying truth assignment for $\mathit{defi}(\varphi)$. We prove, by (structural) induction on φ, that μ satisfies φ:

(Basis) Let φ be a propositional variable a or its negation $\neg a$. Since by definition $\mathit{defi}(\varphi) = \varphi$, the property trivially holds.

(Inductive step) Let φ be a formula $\psi \wedge \chi$. By assumption, μ satisfies $\mathit{defi}(\psi \wedge \chi) = (a_\psi \Leftrightarrow \psi) \wedge (a_\psi \wedge \mathit{defi}(\chi[a_\psi/\psi]))$ and hence satisfies both $a_\psi \Leftrightarrow \psi$ and $a_\psi \wedge \mathit{defi}(\chi[p_\psi/\psi])$, and a fortiori a_ψ. Therefore, μ satisfies ψ. Finally, since μ satisfies $\mathit{defi}(\chi[a_\psi/\psi]))$ as well, by IH this implies that μ satisfies χ too. The other case is similar. ♣

(1.iv) We exhibit a propositional formula for which the $\mathit{cnf}(\cdot)$ transformation yields an exponentially larger formula. Consider the following formula φ in disjunctive normal form over the $2n$ propositional variables $\{a_1, \ldots, a_{2n}\}$:

$$\varphi \;=\; \bigvee_{i=1}^{n} (l_{i1} \wedge l_{i2}),$$

where the (positive) literals l_{ij}s, for $i \in \{1, \ldots, n\}, j \in \{1, 2\}$, are all possibly distinct propositional variables. The formula φ is of size $|\varphi| = 2n + n + (n-1) = \mathbf{O}(n)$. The

[1] The proposed algorithm, Defi$(\cdot)$ is supposed to *compute* this function, and can be assumed to be sound, complete and terminating for $\mathit{defi}(\cdot)$!

conjunctive normal form of is obtained by distributing conjunctions over disjunctions

$$\mathit{cnf}(\varphi) \;=\; \bigwedge_{i=1}^{2^n}(l_{i1} \vee l_{i2}),$$

but $\mathit{cnf}(\varphi)$ will be of size $|\mathit{cnf}(\varphi)| = (2^n - 1) + 2^n + 2n = 2^{\mathbf{O}(n)} = 2^{|\varphi|}$, i.e., exponential in $|\varphi|$ (just try it on an example!).

On the other hand, recall that the composition $(p \circ p')(\cdot)$ of two polynomials $p(\cdot)$ and $p'(\cdot)$, is a polynomial. We thus need to prove two claims:

i. That $\mathit{defi}(\varphi)$ is polynomial in $|\varphi|$. The basic operations of the $\textsc{Defi}(\cdot)$ algorithm are at most linear in $|\varphi|$. Furthermore, there are polynomially many formulas (i.e., sub-trees) in $\mathit{SF}(\varphi)$. Thus $\textsc{Defi}(\varphi)$ and a fortiori $\mathit{defi}(\varphi)$ is polynomial in $|\varphi|$.

ii. That $\mathit{cnf}(\mathit{defi}(\varphi))$ is polynomial in $|\mathit{defi}(\varphi)|$. The algorithm $\textsc{Defi}(\cdot)$ will introduce $\#(\mathit{SF}(\varphi))$ "definitions" for the formula φ, a number that is polynomial in $|\varphi|$. Finally, the number of connectives (and hence the size) in $\mathit{cnf}(\mathit{defi}(\varphi))$ will be polynomial in $|\mathit{defi}(\varphi)|$, which proves the claim.

Therefore, $\mathit{defi}(\varphi)$ is polynomial in φ, as desired. ♣

A Note on Recursion and Induction. Recursion intuitively refers to the expedient by which sets or functions are constructed or defined over some given base sets or functions, by calling themselves finitely many times until, resp., the base sets or base functions are reached. Induction, on the other hand, consists in checking whether a property over a recursive set is "hereditary", viz., invariant under the recursive set's construction operators and thus holds over all its members [3].

The classical example of recursion in mathematics are positive integers $\mathbb{N}$, defined from $\{0\}$ by finitely iterating the successor function $\mathit{succ}(\cdot)$, integer-valued recursive functions (such as $\mathit{fact}(\cdot)$, and the principle of complete induction, wherein if a property P holds for 0 and is invariant under $\mathit{succ}(\cdot)$, then it holds for all $n \in \mathbb{N}$. Inductive sets, structural recursion and structural induction generalize these notions and principles to arbitrary mathematical objects.

Definition 1 (Inductive set). Let $A \subseteq E$ be a set and $f_1, \ldots, f_n$ a family of operations over E of arity, resp., $\mathit{ar}(1), \ldots, \mathit{ar}(n)$. An *inductive set* A^+ over A is the smallest set $\subseteq E$ s.t.

- for all $a \in A$, $a \in A^+$, and
- for all $\alpha_1, \ldots, \alpha_{\mathit{ar}(n)} \in A^+$, $f_i(\alpha_1, \ldots, \alpha_{\mathit{ar}(n)}) \in A^+$, for $i \in \{1, \ldots, n\}$.

Theorem 1 (Structural recursion theorem). *Let A^+ be an inductive set over some set $A \subseteq E$, closed under the operations $f_1, \ldots, f_n$ over E of arity $\mathit{ar}(1), \ldots, \mathit{ar}(n)$. Let $g\colon A \times \cdots \times A \to A$ be a function of arity k, and $s\colon A^+ \times \cdots \times A^+ \to A^+$ a function of arity p. Then $h\colon A^+ \times \cdots \times A^+ \to A^+$ is the unique function of arity k s.t., for $i \in \{1, \ldots, n\}$,*

$$h(a_1, \ldots, a_k) := g(a_1, \ldots, a_k), \text{ and}$$
$$h(f_i^1(\alpha_1^i, \ldots, \alpha_{\mathit{ar}(i)}^i), \ldots, f_i^k(\alpha_1^i, \ldots, \alpha_{\mathit{ar}(i)}^i)) := s(h^1(\alpha_1^i, \ldots, \alpha_k^i), \ldots, h^p(\alpha_1^i, \ldots, \alpha_k^i)).$$

Theorem 2 (**Structural induction principle**). *Let A^+ be an inductive set over a set A closed under operations $f_1, \ldots, f_n$ of arity $ar(1), \ldots, ar(n)$. Let P be a property over A^+ (i.e., $P \subseteq A^+$). If, for all $a \in A$, P is true over a (i.e., $a \in P$), and for all $i \in \{1, \ldots, n\}$ and all $\alpha_1, \ldots, \alpha_{ar(i)}$, the fact that P holds over $\alpha_1, \ldots, \alpha_{ar(i)}$ (i.e., $\alpha_1, \ldots, \alpha_{ar(i)} \in P$) implies that P is true over $f_i(\alpha_1, \ldots, \alpha_{ar(i)})$ (i.e., $f_i(\alpha_1, \ldots, \alpha_{ar(i)}) \in P$), then, for all $\alpha \in A^+$, $\alpha \in P$ (i.e., $P = A^+$).*

Examples of inductive sets are recursive data structures such as lists or trees, or recursive syntactic structures such as propositional formulas. An example of a (recursive) function defined by structural recursion is the $\mathit{defi}(\cdot)$ function defined above. An example of an "hereditary" property that can be shown to hold by structural induction over propositional formulas is, e.g., that they possess an even number of parentheses [1].

Inductive sets, structural recursion and structural induction can be generalized further by invoking ordinals, ordinal recursion and the principle of well-founded induction and (ordinal) transfinite induction of axiomatic set theory (ordinal theory) in pure mathematics [2].

References:

[1] Dirk Van Dalen. *Logic and Structure.* Springer, 1997.

[2] Herbert B. Enderton. *Elements of Set Theory.* Academic Press, 1977.

[3] René Lalément. *Logique, réduction, résolution.* Dunod, 1997.