CS 2742 (Logic in Computer Science) – Winter 2014
Lecture 23

Antonina Kolokolova

March 21, 2014

Recall the notion of recursive definitions from last time:

Definition 1. *A recursive definition consists of:*

1) ***Base of recursion****: a statement that certain objects belong to a set.*
2) ***Recursion:*** *a collections of rules indicating how to form new set objects from those already known to be in the set.*
3) ***Restriction****: A statement that no objects belong to the set other than those coming from the base and the recursion rules.*

In this lecture, we will look at some examples (some repeating from the previous time) and show how to prove properties of recursively defined sets using structural induction.

Example 1. (Propositional formulas.)
Here we will give a formal definition of formulas of propositional logic.
Base of the recursion: propositional variables $p, q, r, \ldots$ and constants F, T are propositional formulas.
Recursion: If ϕ and ψ are propositional formulas, so are $\neg\phi$, $\phi \vee \psi$, $\phi \wedge \psi$, $\phi \rightarrow \psi$ and $\phi \leftrightarrow \psi$, as $(\neg\phi)$, $(\phi \vee \psi)$ and so on.
Restriction: ... and nothing else is a propositional formula.
For example, $(p \vee \neg q) \wedge T$ is a propositional formula, because it is made out of a $\wedge$ of $(p \vee \neg q)$ and T, and both of them are propositional formulas: T because it satisfies the base of induction, and $(p \vee \neg q)$ because it is a $\vee$ of two formulas p and $\neg q$, the first of which again satisfies the base case, and the second is a $\neg$ of a formula which is a base case.

Example 2. (Arithmetic expressions)
Base of the recursion: rational numbers and variables $x, y, z, \ldots$ are arithmetic expressions.

Recursion: For any two arithmetic expressions A and B, $A+B$, $A-B$, $A*B$, A/B, $(A+B)$, $(A-B)$, $(A*B)$, (A/B) are arithmetic expressions.
Restriction:... and nothing else.
For example, $3+5*x$ is an arithmetic expression.

7.1 Structural induction

The version of induction usually used to prove properties of objects in a recursively-defined set.

Definition 2. *(Structural induction)*

1) ***Base case:*** *prove that "simplest" elements (basis of recursion) satisfy the property.*
2) ***Induction step:*** *prove that each of the rules used to construct a more complex element preserves the property.*

Example 3. Here we will show that in any arithmetic expression (defined as above) the number of elements (such as variables or numbers) differs from the number of connectives (such as $+, -, *$ and $/$) by 1. Here, if a number or a variable occurs twice (as in $3-x/3$) we count it twice. More precisely, given an expression A, if it has $el(A)$ elements and $con(A)$ connectives, then $el(A) = con(A)+1$. For example, in the expression $3+5*x$ there are two connectives $+$ and $*$, and three elements $3, 5$ and x.

The proof proceeds by structural induction. For the base case, in expressions consisting of just a number or just a variable there is one element (that number or a variable) and no connectives. So in the base case $el(A) = 1$, $con(A) = 0$ and thus $el(A) = con(A)+1$ holds.

Now we will show that as long as an expression is constructed according to the rules, and out of valid expressions which satisfy (here is our induction hypothesis) the property we are trying to prove, the new constructed expression will also satisfy this property. That is, if C is constructed out of A and B according to the rules, and $el(A) = con(A)+1$ and $el(B) = con(B)+$, then $el(C) = con(C)+1$. But notice that all the rules here look like "expression followed by a connective followed by another expression". So $el(C) = el(A)+el(B)$ (because the new expression will have all the numbers and variables of the old ones, and nothing else – counting each occurrence of a number or variable as a separate element here). Similarly, $con(C) = con(A)+1+con(B)$, where that 1 comes from the connective that connects A and B. Thus, $con(C) = (el(A)-1)+1+(el(B)-1) = (el(A)+el(B)-1-1+1 = el(C)-1$, which is exactly what we were trying to show. We could also go through a separate proof for each of $+, -.*, /$ and the cases with parentheses (which we do not count here at all), but all these cases would be the same. So, for example, combining the expression 3 with the expression $5*x$ using the connective $+$ gives us the expression $3+5*x$ with 1+2=3 elements and 0+1+1=2 connectives.

For a simpler example, consider two possible definitions of even natural numbers.

Example 4. Consider the following two recursive definitions of a set $S \subseteq \mathbb{N}$:

1)
- Base of recursion: $0 \in S$ and $2 \in S$.
- Recursion: if $a, b \in S$, then $a + b \in S$.

2)
- Base of recursion: $0 \in S$.
- Recursion: if $a \in S$, then $a + 2 \in S$.

To complete both definitions, we need to add the third part, restriction: there are no elements in S other than ones obtained from the elements in the base of the recursion by applying the recursion rule. Thus, in the first case we will say that there are no elements in S other than ones obtained from 0 and 2 by repeatedly summing elements from S, and in the other case no elements other than ones obtained from 0 by repeatedly adding 2.

Now, what are the elements in these sets? It is often useful to think about elements that the set contains after a fixed number of applications of the rule to the elements already in the set. In fact, many induction proofs of properties of recursive sets that are not structural induction proofs would indeed proceed by the number of rule application as the n in the predicate. Usually such induction proofs will use the strong induction to be able to talk about combining elements obtained on different "levels", where "level" corresponds to all elements obtained within a given number of rule applications.

Let us look at the first several levels of the definitions above. Consider the first definition. On level 0 (no rule applications) we have 0 and 2 in the set. On level 1, we can combine 0 with 0 to obtain 0, then 0 with 2 to obtain 2 and finally 2 with 2 to obtain 4. Note that it is essential here that we can reuse an element, so our a and b from the recursion can be the same. Now, what do we get on the second level of the recursion? We again obtain $0, 2, 4$ by, for example, using $a = 0$ and iterating b through $0, 2, 4$ that we have by now. We can also obtain 6 by setting $a = 2$ and $b = 4$ (or $a = 4$ and $b = 2$, in this example it does not matter). Finally, we can get 8 by setting $a = b = 4$. So after two levels of applying the recursion rule, we get elements $\{0, 2, 4, 6, 8\}$ in S. Exercise: what are the elements on the second level using the second definition?

Both of these recursive definitions give us the same set, that of even natural numbers. However, one needs to use structural induction to prove that indeed every natural number in the set is even (and strong induction to prove that every even natural number is in the set).

For example, for the second definition, the proof by structural induction can proceed as follows:

- Predicate $P(n)$: $\exists m \in \mathbb{N}$ such that $n = 2m$.
- Base case: 0 is even, since setting $m = 0$ get $0 = 2 * m = 2 * 0 = 0$.
- Induction step: Suppose that $a \in S$ is even, that is, $\exists m \in \mathbb{N}$ such that $a = 2m$. This is our induction hypothesis. Now, let b be an element obtained by applying the rule: there is only one rule $a + 2$, and it involves only one element, a, in our case. We want to show that $\exists m' \in \mathbb{N}$ such that $b = 2m'$. Now, $b = a + 2 = 2m + 2 = 2 * (m + 1)$. So setting $m' = m + 1$ gives us the desired $b = 2m'$, which says that b is even.

Since by restriction there are no elements in S other than ones defined from 0 by the rule $a + 2$, by structural induction every element in S is even.