A discussion of Question B11 in Glynn Winskel's notes

Thomas Forster

February 21, 2013

The Question

"Define the length of a Boolean proposition by structural induction as follows:

$|a| = 1$,
$|\top| = 1$,
$|\bot| = 1$,
$|A \wedge B| = |A| + |B| + 1$,
$|A \vee B| = |A| + |B| + 1$,
$|\neg A| = |A| + 1$.

Define a translation which eliminates disjunction from Boolean expressions by the following structural induction:

$tr(a) = a$, $tr(\top) = \top$, $tr(\bot) = \bot$,
$tr(A \wedge B) = tr(A) \wedge tr(B)$,
$tr(A \vee B) = \neg(\neg tr(A) \wedge \neg tr(B))$,
$tr(\neg A) = \neg tr(A)$.

Prove by structural induction on Boolean propositions that

$$|tr(A)| \leq 3|A| - 1,$$

for all Boolean propositions A."

Discussion

This is a beautiful question, co's it touches several important points. It tests your understanding of structural induction; it tests your ability to do the fiddly manipulation necessary to perform the inductive step; it underlines the importance of having a sufficiently strong induction hypothesis, and finally it makes a point about dereferencing.

So: we have a propositional language—a recursive datatype of formulæ—which starts off with three propositional letters ("literals") 'a', '$\top$' and '$\bot$'. We

then build up compound formulæ by means of the constructors '$\wedge$', '$\vee$' and '$\neg$'. We have a *length* function defined on objects in the datatype of formulæ, written with two vertical bars as in the question, which is roughly what you think it is—so that the length of a literal is 1, and the length of a conjunction (or a disjunction) of two formulæ is one plus the sum of their lengths, and the length of the negation of a formula is one plus the length of the formula. Evidently the question-designer thought that the length of a '(' or a ')' is zero!

One tends naturally to write the second half of the preceding paragraph with expressions like

$$|A \wedge B| = |A| + |B| + 1.$$

This looks fair enough, and in some sense it is, but we need to be clear about the conventions we are using. The letter 'A' by itself is a single symbol, so a pedant might insist that $|A| = 1$. This is wrong of course: the letter 'A' is not a formula, but a variable ranging over formulæ... when looking for the length $|A|$ of A we have to *see through* the variable all the way to the value it takes—and that value is a formula. All this is well and good, but it can cause some confusion when we start thinking about expressions like: $|A \vee B|$. The constructor '$\vee$' is something we put between two formulæ to make a new formula; we don't put it between two *names* of formulæ or between two *pointers* to formulæ! Until we have a convention to make our practice OK, writing things like '$|A \vee B|$' should generate a `syntax error` warning. If you look back to the first page you will find that i wrote

> "...length of a literal is 1, and the length of a conjunction (or a disjunction) of two formulæ is one plus the sum of their lengths..."

...and this is syntactically correct. When we wrote '$|A \wedge B|$' we should really have written '| the conjunction of A and B|'.

There are two ways of dealing with this. One is to have explicit names for the constructors, as it might be 'conjunction of ...' and 'disjunction of ...' and 'negation of ...' This makes huge demands on our supply of alphanumerics. The other solution is to have a kind of `environment` command that creates an environment within which [deep breath]

> *contructors applied to* **pointers** *to objects construct* **pointers** *to the objects thereby constructed.*

Inside such a context things like '$|A \vee B|$' have the meaning we intend here. There is a culture within which this environment is created by the '$\ulcorner$' symbol (LaTeX: `\ulcorner`) and closed by the '$\urcorner$' symbol (LaTeX: `\urcorner`). In fact people tend to leave things out.

Thus we should/should have posed the question as:

"Define the length of a Boolean proposition by structural induction as follows:

$|a| = 1$,
$|\top| = 1$,
$|\bot| = 1$,
$|\ulcorner A \wedge B \urcorner| = |A| + |B| + 1$,
$|\ulcorner A \vee B \urcorner| = |A| + |B| + 1$,
$|\ulcorner \neg A \urcorner| = |A| + 1$.

Define a translation which eliminates disjunction from Boolean expressions by the following structural induction:

$tr(a) = a$, $tr(\top) = \top$, $tr(\bot) = \bot$,
$\ulcorner tr(A \wedge B) = tr(A) \wedge tr(B)$,
$tr(A \vee B) = \neg(\neg tr(A) \wedge \neg tr(B))$,
$tr(\neg A) = \neg tr(A) \urcorner$.

Prove by structural induction on Boolean propositions that

$$|tr(A)| \leq 3|A| - 1,$$

for all Boolean propositions A."

The above use of corner quotes illustrates how there is no restriction that says that the scope of the corner quotes has to live entirely inside a single formula. I use corner quotes in what follows, but (although—i *think*–i have put them in correctly) they can be inserted correctly in more than one way.

The Proof by Structural Induction

We aspire to prove by structural induction on the recursive datatype of formulæ that

$$(\forall A)(|tr(A)| \ \leq \ 3 \cdot |A| - 1)$$

The base case we verify easily. The induction step has three cases

$\neg$ If $|tr(A)| \ \leq \ 3 \cdot |A|$ what is $|\ulcorner tr(\neg A) \urcorner|$? $\ulcorner tr(\neg A) = \neg tr(A) \urcorner$ so $\ulcorner |tr(\neg A)| = |\neg tr(A)| \urcorner$, and $|\ulcorner \neg tr(A) \urcorner|$ is $|tr(A)| + 1$ which is certainly $\leq \ 3 \cdot |\ulcorner \neg A \urcorner|$.

$\wedge$ If $|tr(A)| \ \leq \ 3 \cdot |A|$ and $|tr(B)| \ \leq \ 3 \cdot |B|$ what is $|\ulcorner tr(A \wedge B) \urcorner|$?

$\ulcorner tr(A \wedge B) \urcorner$ is $\ulcorner tr(A) \wedge tr(B) \urcorner$. By induction hypothesis $|tr(A)| \ \leq \ 3 \cdot |A| - 1$ and $|tr(B)| \ \leq \ 3 \cdot |B| - 1$ so $\ulcorner |tr(A) \wedge tr(B)| \urcorner \ \leq \ (3 \cdot |A| - 1) + (3 \cdot |B| - 1) + 1$. The final '+1' is for the '$\wedge$'. This rearranges to

$|\ulcorner tr(A) \wedge tr(B) \urcorner| \ \leq \ 3 \cdot (|A| + |B|) - 1$

but $|A| + |B| \ \leq \ |\ulcorner A \wedge B \urcorner|$ whence

$\ulcorner |tr(A) \wedge tr(B)| \ \leq \ 3 \cdot (|A \wedge B|) - 1 \urcorner$ and finally

$\ulcorner |tr(A \wedge B)| \ \leq \ 3 \cdot (|A \wedge B|) - 1 \urcorner$.

$\bigvee$ If $|tr(A)| \leq 3 \cdot |A|$ and $|tr(B)| \leq 3 \cdot |B|$ what is $|tr(A \vee B)|$? $\ulcorner tr(A \vee B) \urcorner$ is $\ulcorner \neg(\neg tr(A) \wedge \neg(tr(B))) \urcorner$. What is the length of this last expression? Clearly it's going to be $|tr(A)| + |tr(B)|$ + one for the outermost '$\neg$' + one for the '$\neg$' attached to $tr(A)$ + one for the '$\neg$' attached to $tr(B)$ + one for the '$\wedge$' ... giving $|tr(A)| + |tr(B)| + 4$. By induction hypothesis $|tr(A)| \leq 3 \cdot |A| - 1$ and $|tr(B)| \leq 3 \cdot |B| - 1$ so we have

$\ulcorner |tr(A \vee B)| \leq (3 \cdot |A| - 1) + (3 \cdot |B| - 1) + 4 \urcorner$. We can rearrange this to

$\ulcorner |tr(A \vee B)| \leq 3 \cdot (|A| + |B|) - 1 - 1) + 4 \urcorner$ and further to

$\ulcorner |tr(A \vee B)| \leq 3 \cdot (|A| + |B|) + 2 \urcorner$.

Now $|A| + |B| = \ulcorner |A \vee B| \urcorner - 1$ so we can substitute getting

$\ulcorner |tr(A \vee B)| \leq 3 \cdot (|A \vee B| - 1)) + 2 \urcorner$ and rearrange again to get

$\ulcorner |tr(A \vee B)| \leq 3 \cdot |A \vee B| - 1 \urcorner$ as desired.

A final thought ... I wouldn't mind betting that quite a lot of thought went into this question. We've proved $|tr(A)| \leq 3 \cdot |A| - 1$ so we've certainly also proved the weaker claim $|tr(A)| \leq 3 \cdot |A|$. However wouldn't stake my life on our ability to prove the weaker claim by induction. You might like to try ... i'm not going to!