ITPPV Homework 7

due 23:59 CET Friday April 3, 2020

1 Regular Expressions and String Matching

Consider regular expressions as an inductive datatype according to the following definition:

$$r ::= \mathbf{0} \mid \mathbf{1} \mid c \mid r + r \mid r \cdot r \mid r^*$$

Encode this datatype in HOL4 in the most general way possible, i.e., let c be an arbitrary type rather than an ASCII character or similar.

Next, consider a matching relation $\lhd$ between strings s as lists of arbitrary characters and regular expressions r (ϵ is the empty list/string):

$$\frac{}{\epsilon \lhd \mathbf{1}} \qquad \frac{}{c \lhd c} \qquad \frac{s \lhd r_1}{s \lhd r_1 + r_2} \qquad \frac{s \lhd r_2}{s \lhd r_1 + r_2}$$

$$\frac{s \lhd r_1 \quad s' \lhd r_2}{s\, s' \lhd r_1 \cdot r_2} \qquad \frac{}{\epsilon \lhd r^*} \qquad \frac{s \lhd r \quad s' \lhd r^*}{s\, s' \lhd r^*}$$

Encode this relation in HOL4, and prove that it holds as expected for some simple but not completely trivial list of characters, for example: $aaaabbbb \lhd a^* \cdot b^*$

Finally, encode the following relation, which states roughly that "the string s matches r when the character c is added to the beginning of s":

$$\frac{cs \lhd r}{s \lhd_c r}$$

2 Function Returning Lists of Restricted Regular Expressions

For this task, restrict the previous notion of regular expression and relations to a fragment:

$$r ::= \mathbf{0} \mid \mathbf{1} \mid c \mid r + r$$

Figure 1 on page 7 in the paper "Proof-Directed Debugging Revisited for a First-Order Version"[1] contains a definition of a function † that, given a regular expression r and a character c (written $r\dagger_c$) returns a set of regular expressions. Your task is to implement this function in HOL4, but returning lists rather than sets. Then, you should specify the correctness of your function and prove it correct. Informally, the function can be specified as follows:

input a regular expression r and a character c (in some completely arbitrary alphabet)

output a *sound* and *complete* list of regular expressions for strings that match r after their leading character c has been removed

More formally, suppose $l = r\dagger_c$. Then, for all s such that $s \lhd_c r$, there must exist some $r' \in l$ such that $s \lhd r'$. Moreover, if $r'' \in l$, then for all s such that $s \lhd r''$, it must be the case that $s \lhd_c r$.

[1] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.122&rep=rep1&type=pdf

3 Obtaining Verified Executable Code

Obtain verified executable code for your HOL4 implementation of the list version of the function †—either through CakeML or the EmitML module bundled with HOL4. Check that the code compiles properly (either with the CakeML bootstrapped compiler[2] or PolyML). For CakeML boilerplate, see the CakeML lecture slides[3] and code for the proof system discussed in the lecture[4].

4 Optional: Handle Complete Regular Expressions

Using the Proof Directed Debugging paper as a guide, extend the previous verified function and code to the complete definition of regular expressions and matching. Note that this will require using a non-trivial well-founded relation on input for termination, as outlined in the paper.

[2]`https://github.com/CakeML/cakeml/releases/download/v1009/cake-x64-64.tar.gz`
[3]`https://kth-step.github.io/itppv-course/lectures/lec10.pdf`
[4]`https://github.com/palmskog/fitch/tree/master/cakeml`