Computer Science Tripos 2018 Paper 2 Question 10
A Discussion Answer

Thomas Forster

April 20, 2022

This is a nice question.
(a) (i) obviously not regular.

(a) (ii) This is very cute. Every time you change from reading as to reading bs you get an ab and every time you change from reading bs to reading as you get a ba. Since the language only contains as and bs a string has the same number of abs as bas iff the first character is the same as the last. So it's regular. (How many states *exactly* do you need to check whether or not the first character is the same as the last?)

Notice that this argument works to show that the language is regular only beco's the alphabet contains no character other than 'a' and 'b'. If it had any other characters it would fail to be regular for the same reason that (i) is not regular. It's easy to show that if you start with a regular language and delete from its words all occurrences of a fixed character then the result is regular. This example shows that if you try to go in the other direction it doesn't work. And this despite the fact that an interleaving of two regular languages is regular!!!

(a) (iii) This is $a^n b^n$ wot ain't regular.

(a) (iv) This is $a^* b^*$ wot is regular.

(a) (v) Try my thought-experiment. What do you have to keep track of, and how many bits do you need to keep track of it? You need to keep track of the number of as mod 3 (and that's just under 2 bits) and the number of bs mod 7 (and that's just under 3 bits) so definitely do-able by a DFA.

(b) is bookwork. I don't need this!!

But i suppose i should say something. The rectype of regular expressions over a language has two binary constructors, juxtaposition and disjunction. Binary constructors bring with them the danger that you night need brackets, at least if they aren't associative, beco's you then need to distinguish between $A * (B * C)$ and $(A * B) * C$. Actually juxtaposition and disjunction are both associative so that isn't an immediate problem. However we do have *two* constructors, and we have to distinguish between $(A|B)C$ and $A|(BC)$, so we do need brackets after all. And once you have brackets your language ceases to be regular, co's you have to keep track of how many you have opened!