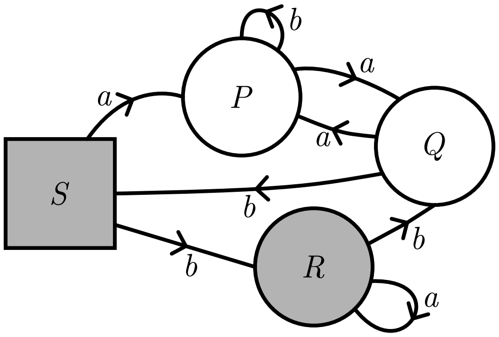
Problem of the Month

Solution to Problem 2: Manuals for Robots

November 2024

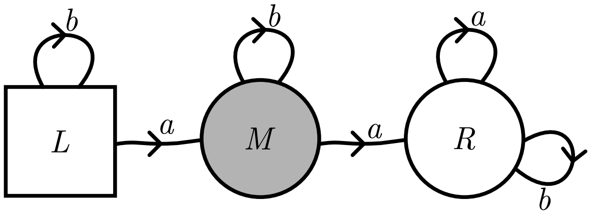
1. Here is Course 2, with some labels on the places.



The string abb ends at place P. The string aab ends at place S. The string $baaaa$ ends at place R. The string $abba$ ends at place Q.

The finishing places of the course are S and R. Therefore aab and $baaaa$ are in the manual, and abb and $abba$ are not in the manual.

2. (a) The manual for this course is the set of strings with exactly one a.



To justify this statement, name the places L, M, R (for left, middle, and right) from left to right respectively, as above. Note that L is the starting place, and M is the only finishing place.

There are a few important properties to notice about this course.

- The only way to move from place L to place M is through the a arrow.
- The only way to move from place M to place R is through the a arrow.
- There is no way to move to place L from another place.
- There is no way to move from place R to any other place.

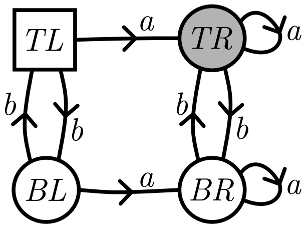
Suppose now that a string contains zero a's. Then the robot ends at place L so the string is not in the manual.

Suppose a string contains at least two a's. After the first two a's, the robot is at place R. Since there is no way to move from place R, the robot stays at place R until the string is completed. Therefore the string is not in the manual.

Finally, suppose a string contains exactly 1 a. Until the robot encounters the a, it stays at place L. The a moves the robot to place M. The rest of the string consists only of b's, so the robot ends at place M. We can conclude the string is part of the manual.

Therefore the manual consists precisely of those strings with exactly one a.

(b) The manual is the set of strings that has at least one a and an even number of b's.



Let's justify this. Label the places TL, TR, BL, BR for top left, top right, bottom left, and bottom right as above. The starting place is TL and the only finishing place is TR. Here are the important properties of this course.

i. If the robot is in the top row (TL or TR), then moving along an arrow labelled b moves the robot to the bottom row (BL or BR). This is the only way to move from the top row to the bottom row.

ii. If the robot is in the bottom row, then moving along an arrow labelled b moves the robot to the top row. This is the only way to move from the bottom row to the top row.

iii. If the robot is in the left column (TL or BL), then moving along an arrow labelled a moves the robot to the right column (TR or BR). Further, this is the only way to move from the left column to the right column.

iv. Once the robot is in the right column, it stays in the right column.

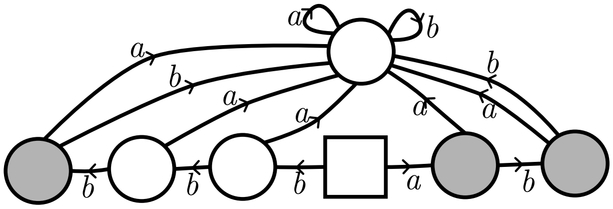
Suppose we have a string in the manual. Since the robot starts in the left column and finishes in the right column, the string must contain at least one a by Properties (iii) and (iv). Since the robot starts and ends in the top row, by Properties (i) and (ii), there must be an even number of b's in the string.

On the other hand, suppose there is a string with an even number of b's and at least one a. By Properties (i) and (ii), the robot must move between the top and bottom rows an even number of times. Since it starts at the top row, it must end at the top row. By Properties (iii) and (iv), since the robot has traversed at least one arrow labelled a, it must end up in the right column. We conclude that the robot ends in the place TR, and therefore the string belongs to the manual.

Therefore the manual consists exactly of those strings with at least one a and an even number of b's.

3. There are many possible correct courses for this question. We will give one correct answer for each part.

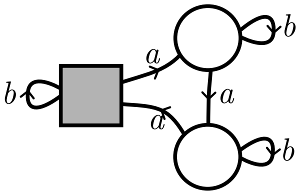
(a) Here is such a course:



Let's justify ourselves. There are three finishing places, and for each one there is only one way to get from the starting place to that finishing place. The three paths are a, ab, and bbb. Therefore the manual for this course is the set of three strings $\{a, ab, bbb\}$.

Notice how there is a place above which acts as a kind of garbage can for strings that don't belong to the desired manual. We can imitate this construction in general to construct a course with any finite set of strings as its manual!

(b) Here is such a course:



To see why this is the case, notice that every time an a appears in a string, the robot moves one place clockwise. Since there are 3 places, the robot returns to the starting place (which is also the only finishing place) exactly when there has been a multiple of 3 a's in the string.

Of course, there is nothing special about the number 3 here. Fix a positive integer k. We can imitate this construction to build a course whose manual is the set of all strings with a multiple of k appearances of the letter a.

4. (a) Notice that if a string has length 1, then the robot will visit two, not necessarily distinct, places - the starting place and some other place. If a string has length 2, then the robot will visit three, not necessarily distinct, places on its journey. If a string has length k, then the robot will visit $k + 1$ not necessarily distinct places on its journey. Since there are k places in our course, if a robot follows the instructions of a string of length at least k, then there must be some place which the robot visits twice (by the pigeonhole principle).

Let w be a string of length at least k in the manual. Let's call the first place that is visited twice p. Let's split w up into three substrings. The first substring, x, will be the string that takes the robot from the starting place to p for the first time. Note that x may be empty if p is the starting place. Then there is some substring y that takes the robot from p back to p for the second time. Note that $y \neq e$ since we are waiting for the robot to visit p again. Let z be the rest of the string w, which takes the robot from p (which has now been visited twice by the robot) to one of the finishing places. Note that z may be empty. While travelling along the course according to the instructions in z, the robot may visit p again, but we don't care!

We now have that $w = xyz$, where $y \neq e$ and y takes the robot from p to p. The string x takes the robot from the starting place to p, and the string z takes the robot from p to a finishing place. Therefore the string xz takes the robot from the starting place to a finishing place. Alas, xz is also in the manual!

(b) Let $w = xyz$, where x, y, and z are constructed as in the solution to part (a). Notice that since y takes the robot from place p to place p, then y^n also takes the robot from place p to place p. It simply repeats the path n times.

Consider now the string xy^nz for any natural number n. The string x takes the robot from the starting place to p, the string y^n takes the robot from p to p, and the string z takes the robot from p to a finishing place. Therefore a robot following the instructions given by the string xy^nz travels from the starting place to a finishing place, so xy^nz is in the manual.

(c) Suppose the set of strings of perfect-square length is the manual for some course, and suppose that course has k places. Choose any string w of length k^2. Since the length of w is a perfect square, it is in the manual. Since $k^2 \geq k$, by part (b) we can write $w = xyz$ for some substring $y \neq e$ so that xy^nz is in the manual for every natural number n. Let l be the length of w and let $d \geq 1$ be the length of y. Then the sequence of words in the manual

$$xyz, xy^2z, xy^3z, \ldots$$

have lengths

$$l, l+d, l+2d, \ldots$$

respectively. Said another way, the lengths of the words $xyz, xy^2z, xy^3z, \ldots$ form an arithmetic sequence. Therefore, in order to complete the problem, we have to prove the following fact:

Fact: In an arithmetic sequence, there is at least one number that is not a perfect square.

Proof. Let l and d be integers with $d \geq 1$, and consider the arithmetic sequence $l, l+d, l+2d, \ldots$. Let $t \geq d$ be an integer. Then

$$(t+1)^2 - t^2 = 2t + 1 \geq 2d + 1 > d.$$

Exactly one of every d consecutive numbers greater than l is in the arithmetic sequence. Choose t to be any integer satisfying $t \geq d$ and $t^2 \geq l$. Since the difference between $(t+1)^2$ and t^2 is greater than d, there must be some number in the arithmetic sequence strictly between t^2 and $(t+1)^2$. However, there are no perfect squares between t^2 and $(t+1)^2$, completing the proof. □

Great! We have just shown that at least one of the lengths of $xyz, xy^2z, xy^3z, \ldots$ is not a perfect square, but they are all in the manual. Therefore, there is no course whose manual consists exactly of all strings with perfect square lengths.

(d) As above, suppose there is a course with manual consisting exactly of all palindromes, and suppose the course has k places. Then the string $w = a^kba^k$ is in the manual and has length $2k + 1 > k$. By Part (b), we can write $w = xyz$ with $y \neq e$ and xy^nz in the manual for all natural numbers n.

By the construction of x, y, and z in the solution to Part (a), the substring xy takes the robot from the starting place to the second occurrence of some place. By the pigeonhole principle, this has to happen within the robot visiting $k+1$ places, which happens after moving across k arrows. It follows that xy has length at most k.

Therefore to write $a^kba^k = xyz$ with $y \neq e$ and the length of xy at most k, we must have that $x = a^t$ and $y = a^s$ with $s \geq 1$ and $s + t \leq k$. This is because any substring at the beginning of the string a^kba^k of length at most k must just be a string of a's. It follows that $z = a^{k-s-t}ba^k$.

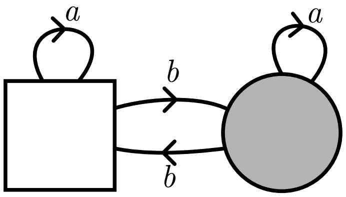
Now $xy^2z = a^ta^{2s}a^{k-s-t}ba^k$. Since $t + 2s + k - s - t = s + k > k$, xy^2z is not a palindrome but it is in the manual. We have a contradiction and can conclude that there is no course whose manual is the set of all palindromes.

Problem of the Month

Curious Context for Problem 2: Manuals for Robots

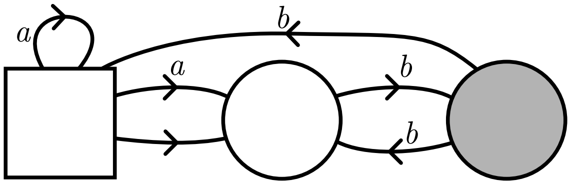
November 2024

Regular languages.
In November's problem you were asked to study the set of strings that form the manual for a particular course. In computer science, these courses are called *Deterministic Finite Automata*, or DFAs. A set of strings which arises as the manual for a DFA is called a *regular language*. For example, the manual for the DFA below is the set L of all strings with an odd number of bs. So L is an example of a regular language.



In Problem 4 you showed that the set of strings with length equal to a perfect square, and the set of palindromes, are both not regular languages. The tool used to prove that these are not regular languages is presented in Problem 4(b), and is known as the *pumping lemma*. The pumping lemma was first proved in 1959 by Michael Rabin and Dana Scott [1]. They used the pumping lemma to prove that the set of all strings of the form a^nba^n, where n is a natural number, is not a regular language.

Nondeterministic finite automata.
In the definition of a DFA, we could relax the condition on the arrows, and allow any number of arrows labelled a, and any number of arrows labelled b to leave from any place (instead of exactly one of each). We can even have some arrows without a label at all, allowing you to move between places without using a letter. Such courses are called *Nondeterministic finite automata*, or NFAs. Notice that in a DFA, once you have a string, your path through the arrows is completely determined. In an NFA, you may have some choices. Here is an example of an NFA:



On first glance, it appears that NFAs allow you to have much more interesting languages (or "manuals" in the language of the POTM). Remarkably, this is not the case! In the same paper that introduced the pumping lemma, Rabin and Scott proved that any language attained from an NFA can be attained from a DFA. Said another way, if a set of strings is the manual for some NFA, then it's the manual for some DFA and is therefore a regular language.

Regular expressions.
Regular expressions are a tool used in computer programming to search through a list of words

and find those which satisfy some condition. For example, you can use regular expressions to search through a list of words and return those which start and end with the letter ‘c’. Regular expressions are a great tool to study patterns in written pieces of work, or to get a computer to solve a crossword by brute force! When you instruct a computer to use a regular expression, the computer creates an NFA and runs each word in your list through the NFA, checking if it’s part of the manual.

Chomsky hierarchy.

In linguistics, there is a notion of computational complexity of a particular language, captured by something called the *Chomsky hierarchy.* Roughly, it measures how much computational power is needed to parse a sentence in that language. There are four types of grammars in the hierarchy, and regular languages are the simplest type (what is referred to as Type-3). The other types of grammars, in increasing order of complexity, are called context-free, context-sensitive, and recursively enumerable.

References

[1] Rabin, M. O. and Scott, D. *Finite automata and their decision problems.* IBM J. Res. Develop. 3 (1959), pp. 114-125.