Determinism versus Nondeterminism

György Maróti, Szeged (Hungary)

Abstract: This paper is the third item of the sequence of articles, which has been devoted to investigate nondeterminism within automata theory from didactic point of view. The first two articles discussed the teaching of notions and proofs, while this paper deals with the didactic approach of teaching problem solving.

Kurzreferat: Dieser Aufsatz ist der dritte in einer Folge von Artikeln, die der Untersuchung von Nichtdeterminiertheit in der Automatentheorie aus didaktischer Perspektive gewidmet sind. Die ersten beiden Aufsätze beschreiben die Behandlung von Begriffen und Beweisen, während sich dieser Artikel mit didaktischen Strategien zum Problemlösen befasst.

ZDM-Classification: C20, D40, H70

1 Introduction

Our aim is to find CAS based didactic methods for teaching nondeterminism within automata theory. We already discussed the introduction of the notion of nondeterministic automaton (see [5]) which was followed by the second article (see [6]) whose subject was the equivalence of nondeterministic and deterministic automaton.

In this paper we come to the point of teaching automata theoretic problem solving with CAS. According to the rule of activity (see [1]), which states the necessity of the active participation of students in every phase of learning process we choose the method of guided exploring learning (see [1] and [4]). Although the didactic literature divides this method into four phases so that we can emphasize the importance of preparatory phases we prefer to use five phases:

- Motivation,
- collecting experience,
- abstraction, solving the problem,
- integrating the new content into the existing system of knowledge,
- evaluation, summary.

In the motivation phase the main goal is to arouse the student's curiosity. How can we achieve this? Supposing a well-structured curriculum, the present knowledge of students is always in accordance with the exercises and problems to be solved. This may induce the feeling in the students that their knowledge is adequate and absolute. "There is no need for newer knowledge as we can solve every problem we have to".

For that reason one possible tool of motivation is to point out that the present knowledge of the students can only be adequate relative to a specific circle of problems, and never can be absolute. The teacher has to point out that one can always find problems, which are important to be solved and can not be solved by means of the students' present knowledge. We consider both conditions necessary to make students motivated.

All these steps may make the students temporarily unsure which induces inner tension. This is exactly what we need. A student without this healthy inner tension is disinterested. On the other hand this inner tension will be dissolved by the end of the learning process, which gives delight and a real kick.

Our answer for the questions "how to practice" and "how to collect experience" is the usage of CAS. Computer algebra systems like Maple turned out to be adequate tool for modern teaching of mathematics. Being interactive systems CAS means effective device for experiment. The didactic challenge we have to face here is how to organize the spreadsheet?

One possible way is to use PoP-EoP blocks (see [5]) which designate a series of associated commands. The command in PoP (point of practice) usually contains the data (parameters) on which the following calculation depends. This helps the students to alter easily the input parameters and check how the result of calculation changes at EoP (end of practice).

In this paper we consider the automata theory package as a tool for mathematical problem solving. As the subject of our investigation is the relationship of determinism and nondeterminism we focus our attention to this aspect of the matter.

In Section 1 we solve two simple exercises and point out that the way we solve automata theoretic problems often (in fact almost always) leads to nondeterministic automaton. Next in Section 2 we prove that both the deterministic and the nondeterminsitic automaton we created in previous Section are solutions of the exercise 2. The comparison of two proofs reflects to the major differences in the technical usage and also the differences in cognitive efficiency between two types of automata.

2 Problem solving

The aim of this section is twofold. On the one hand we show how we use Maple's automata theory package in problem solving. On the other hand, however, we are interested in what types of automata are resulted in the course of problem solving process. For that reason we solve two simple but hopefully nontrivial exercises. We create not only one solution in order to investigate and compare the properties of resulting automata.

2.1 Exercise 1

Create automaton, which recognizes the language

$$\{x^{3n}y \mid n \geq 0\}$$

Following György Pólya let us begin by solving a simpler problem. Erase the letter y at the end of words to be accepted and consider first the language

$$\{x^{3n} \mid n \geq 0\}$$

This language consists of words like ε , x^3, x^6, ...etc, which can be easily recognized by a three-state automaton.

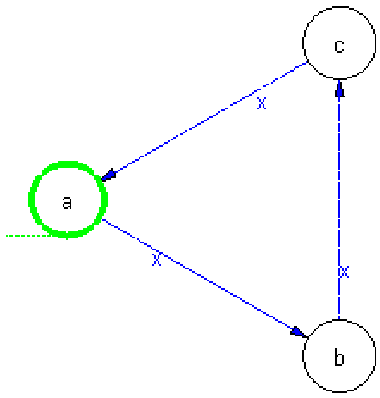
```
> with(aut);
```

[*Analyze*, *Cartesian*, *Compose*, *ConCom*, *ConDet*, *ConIco*, *ConMin*, *ConNet*, *Construct*, Δ, *Rabsorb*, *Reduce*, *Represent*, *Rexpand*, *Rparse*, *Rsimplify*, *Rsubs*, *Runparse*, *Synthesize*, *accept*, *addaut*, *closure*, *delaut*, *disjoint*, *genabc*, *genaut*, *genexp*, *genwords*, *init*, *isequi*, *isword*, *langaut*, *plotaut*, *printaut*, *randaut*, *renaut*, *showaut*]

```
> Xi:=genaut([a,x,b,b,x,c,c,x,a],a,a);
```

$$\Xi := [\{a, c, b\}, \{x\}, \delta, \{a\}, \{a\}]$$

```
> plotaut(Xi);
```

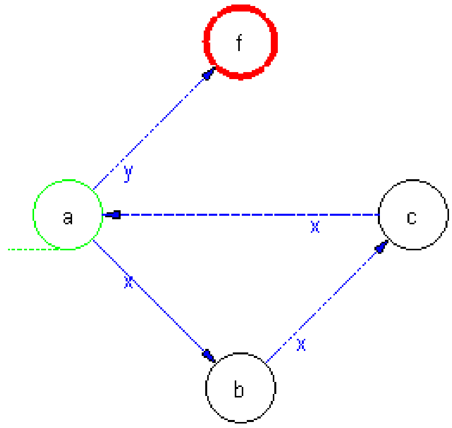


Now we can alter this automaton so that it recognize words y, x^3y, x^6y ...etc. We have nothing else to do than to create a new transition for state a and the input signal y. The result of this transition must be the new final state, which on the same time can be none of the existing states (why?). So we have to establish a new state, say f. As the state a is no more final state, we first delete it from set of final states and then add a new transition and a new final state.

```
> Phi:=addaut(delaut(Xi,final=a),\
      transition=[a,y,f],final=f);
```

$$\Phi := [\{a, f, c, b\}, \{x, y\}, xT, \{a\}, \{f\}]$$

```
> plotaut(Phi);
```

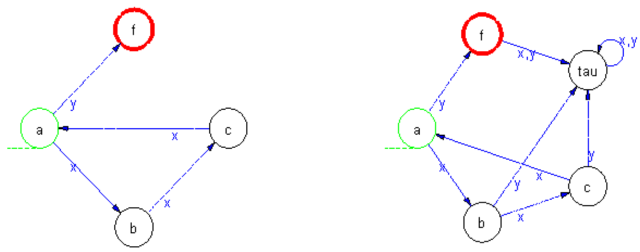


This automaton is the solution of our exercise. Let us consider the characteristic features of this automaton. It is deterministic, but not complete. Indeed, several transitions are not defined. These transitions, however, are irrelevant from the point of view of word recognition. If we insisted on getting a complete automaton as a result of our exercise the resulted automaton would possess more states and a lot of irrelevant transitions. This of course corrupts vividness and clarity.

```
> Omega:=ConCom(Phi);
```

$$\Omega := [\{\tau, a, f, c, b\}, \{x, y\}, \delta, \{a\}, \{f\}]$$

```
> plots[display]({plotaut(Phi,[-
2,0]),plotaut(Omega,[2,0])});
```



Automaton Ω has more state and several redundant transitions, which makes the understanding of its operation difficult. While we are able to decide at a glance e.g. that automaton Φ recognizes all the words y, x^3y, x^6y …etc. the same task needs minutes in case of automaton Ω. The didactic difference between automaton Φ and Ω is even more eye-catching if we want to substantiate that the automata Φ and Ω do not recognize other words.

2.2 Exercise 2.

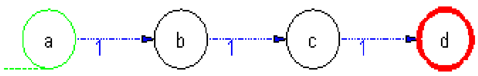
Create automaton, which recognizes a word over the alphabet {0,1} if and only if it contains three consecutive 1s .

Similar to the solution of previous exercise we divide exercise 2 into simpler subproblems. The recognition of three consecutive 1s is easy task.

```
> Xi:=genaut([a,1,b,b,1,c,c,1,d],a,d);
```

$$\Xi := [\{a, c, d, b\}, \{1\}, \delta, \{a\}, \{d\}]$$

```
> plotaut(Xi,line);
```

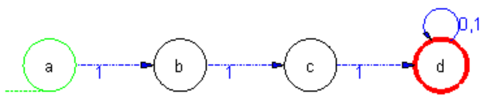


How can we apply this temporary automaton to solve our original problem? The words we have to recognize must contain a subword 111, which means it can have an arbitrary suffix v. It is easy to alter automaton Ξ so that it can recognize words of the form 111v. As Ξ does not have any defined transition for state d, we are free to specify two new transitions δ (d, 0) = δ (d, 1) = d.

```
> Phi:=addaut(Xi,transition={[d,0,d],\
              [d,1,d]});
```

$$\Phi := [\{a, c, d, b\}, \{0, 1\}, xT, \{a\}, \{d\}]$$

```
> plotaut(Phi,line);
```

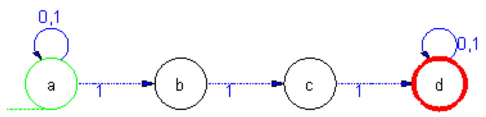


A word having subword 111 can contain not only an arbitrary suffix but also an arbitrary prefix. We already have solution for suffix, so we have every reason to do the same for prefix.

```
> Phi:=addaut(Phi,transition={[a,0,a],\
              [a,1,a]});
```

$$\Phi := [\{a, c, d, b\}, \{0, 1\}, xT, \{a\}, \{d\}]$$

```
> plotaut(Phi,line);
```

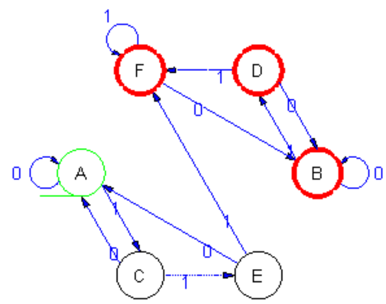


This automaton, which solves problem 2 is neither complete nor deterministic. If we insisted on obtaining a deterministic automaton we would result in the following one.

```
> Omega:=renaut(ConDet(Phi),sta=A);
```

$$\Omega := [\{D, E, F, A, B, C\}, \{0, 1\}, \delta, \{A\}, \{D, F, B\}]$$

```
> plotaut(Omega);
```



Ω has more states and more transitions. As all of the states take part in word recognition we can leave none of them in the same way as we did in the previous session. We draw attention, however, this does not mean that there is no equivalent automaton with fewer states.

2.3 Summary

We solved two problems of the same characteristic. In both cases we had to find automaton which recognizes a given language. We reached the solution in such a way that we begun with an automaton, which recognized a language being related with the given language and we altered this automaton as long as we obtained the automaton, required. This method led us to deterministic but not complete automaton in the first case and nondeterministic and non-complete automaton in the second case.

In general we can assert that in most cases the natural human thinking results in non-complete and nondeterministic automata in the course of automata theoretic problem solving. As for non-completeness the reason for this is that people are originally lazy. Yes, we do not like needless work and prefer to reach the result with the most small effort. On the same time irrelevant states and transition do not help us to comprehend the operation of resulting automata, which means the cognitive efficiency of complete and deterministic automata is lower.

As for nondeterminism the cognitive structure of students seems to be closer to nondeterministic model of finite state machines than that of deterministic machines. We prefer to imagine an automaton as a directed graph (graphic representation) and we look for different directed paths in this graph.

3 Proofs

In this section we prove that both automata Φ and Ω are solutions of exercise 2. Our aim is to present two proofs, which gives the possibility to compare them and to assert some general didactic remarks.

3.1 Automaton Φ : nondeterministic case

Let w=u111v where u and v are arbitrary words over the alphabet {0,1}. One possible operation of Φ is that it remains in state a while processing word u, next it reads the three 1s and moves to the state f where it processes the suffix v. In this way automaton Φ recognizes the words w. On the other hand if an input words does not have three consecutive 1s then automaton Φ is not able to reach the state f in the course of processing w. This yields Φ does not recognizes other words.

3.2 Automaton Ω : deterministic case

According to the fact that automaton Ω has got a more complicated inner structure this proof is not so easy as in the nondeterministic case. Let w an input word of the form u111v. Without the loss of generality we can assume we indicated the first occurrence of three consecutive 1s in w, which yields

1) u does not have three consecutive 1s and
2) the last signal of u is 0.

We need two observations, which will be useful later on..

Observation 1

For every input word u with properties 1) and 2)

$$\Delta(\Omega, A, u) = A.$$

We prove the observation by induction on the length of u. As the statement is obvious for the one letter word 0, let us assume that the length of u is greater then 1 and that the statement is true for every words whose length is less the that of u.

Because of 1) and 2) u=$u_1 z0$, where $z \in \{\varepsilon, 1, 11\}$ and u_1 satisfies 1) and 2). As the length of u_1 is less then that of u using the induction hypothesis we obtain $\Delta(\Omega, A, u_1) = A$.

On the other hand $\Delta(\Omega, A, z) \in \{A, C, E\}$ and hence

$\Delta(\Omega, A, u) =$
$= \Delta(\Omega, A, u_1 z0) =$ # u=u[1]z0
$= \Delta(\Omega, \Delta(\Omega, A, u_1), z0) =$ # definition of Δ
$= \Delta(\Omega, A, z0) =$ # induction hypothesis
$= \Delta(\Omega, \Delta(\Omega, A, z), 0) =$ # $\Delta(\Omega, A, 0) = A$
$\Delta(\Omega, C, 0) = A$
$\Delta(\Omega, E, 0) = A$
$= A$ and this proves Observation 1.

Observation 2

For every state $s \in \{B, D, F\}$ and for every input word w the state $\Delta(\Omega, s, w) \in \{B, D, F\}$

The proof of observation 2 is evident.

Now let us consider an input word $w = u1111v$ where u fulfills 1) and 2). Then by Observation 1 we obtain $\Delta(\Omega, A, u) = A$. Next

$\Delta(\Omega, A, w) =$
$= \Delta(\Omega, A, u111v) =$ # w=u111v
$= \Delta(\Omega, \Delta(\Omega, A, u), 111v) =$ # definition of Δ
$= \Delta(\Omega, A, 111v) =$ # Observation 1
$= \Delta(\Omega, F, v)$ # specification of Ω

By Observation 2

$$\Delta(\Omega, F, v) \in \{B, D, F\}$$

which means $\Delta(\Omega, F, v)$ is final state. This proves that the word w is accepted by automaton Ω.

Conversely, assume Ω recognizes an input word w. We have to show that w contains three consecutive 1s. As

$$\Delta(\Omega, F, w) \in \{B, D, F\}$$

the input word w must be of the form $w = u1v$, where

$$\Delta(\Omega, A, u) = E.$$

If u does not contain 0 that u must coincide with 11 and this yields $w = u1v = 111v$. Therefore w contain three consecutive 1s.

On the other hand if u contains 0 then choosing the last occurrence of 0 in u we obtain $u = z011$. Indeed, $\Delta(\Omega, A, u) = E$ yields that $\Delta(\Omega, A, z)$ must belong to set $\{\mathrm{A}, \mathrm{C}, \mathrm{E}\}$ and hence

$\Delta(\Omega, A, u) =$
$= \Delta(\Omega, A, z0v) =$ # u=z011
$= \Delta(\Omega, \Delta(\Omega, A, z), 0) =$ # definition of Δ
$= A$ # specification of Ω

Again using the equality $\Delta(\Omega, A, u) = E$ we obtain that u must end with two consecutive 1s. Hence

$$w = u1v = (z011)1v = z0(111)v$$

which ends the proof that w contains three consecutive 1s.

3.3 Summary

These two proofs clearly show the difference between the work with nondeterministic and deterministic automata. In case of nondeterministic automata our proof was short, clear and elegant. The simpler inner structure of nondeterministic automata leads to simpler proof of statements. This serves additional argument for our former remark that nondeterministic automata are closer to cognitive structure of human being.

On the same time informal arguments may be risky as it is easy to make errors or to overlook certain cases. For that reason we need the formal and algebraic way of proofs, which is served by deterministic automata.

Indeed, deterministic automata can be considered as special (unary) algebra and this gives the possibility to treat them by effective algebraic tools. Although these tools are heavily formalized they are far from being needless. In the contrary, formal proofs help us to mould exact mathematical thinking and to widen our practice for symbolic computation.

4 Conclusion

Historically, the usual way for introduction the notion of finite automata is to begin with deterministic automata as the mathematical model of finite state machines (see [3]). The definition and the discussion are highly algebraic based on symbolic representation of automata. First the students learn how to handle deterministic automata with algebraic tools how to describe their operation and how to use them to recognize words and languages. After that the notion of nondeterministic automaton is introduced as a generalizaton of deterministic one. This treatment is based on symbolic representation, which makes the understanding of nondeterministic automata very difficult. The teaching of nondeterministic automata means serious didactic challenge for all lecturers.

On the other hand we have seen that if we choose graphical representation instead of symbolic one then nondeterminsitic automata become to be easy and simple to handle. Indeed, nondeterminsitic automata turned out to be closer to intuitive human thinking. On the other hand nondeterminsitic automata are exponentially succinct than deterministic ones, which means exponentially less states and less transitions. This simpler inner structure allows much simpler and clearer treatment of proofs of different statements.

5 References

[1] A. Ambrus, Bevezetés a Matematikadidaktikába, Eötvös kiadó,1995

[2] K.J. Fuchs, Computer Algebra Systems in Mathematic Education, International Symposium - Anniversary of Pollack Mihály College of Engineering, 2002

[3] Kozen: Automata and Computability. - Springer-Verlag, New York, 1997.

[4] D. Mackie, Using Computer Algebra to Encourage a Deep Learning Approach to Calculus, http://www.math.uoc.gr/~ictm2/Proceedings/pap415.pdf

[5] Gy. Maróti, Didactic Approach for Teaching Nondeterminism in Automata Theory, ZDM, 2003

[6] Gy. Maróti, CAS based approach for discussing subset construction, ZDM, 2003

[7] M.B. Monagan, K.O Geddes, K.M. Heal, G.Labahm, S.M.Vorkoetter, J. McCaron. P.DeMarco: Maple 8 Advanced Programming Guide -Waterloo Maple , 2002

[8] National Council of Teachers of Mathemetics (NCTM), Principles and Standards for School Mathematics, 2000

Autor

György Maroti, – Epigramma Kft.,Szent-Györgyi Albert u. 2., H-672 (Hungary)

E-mail: maroti@epigramma.hu