Vending machines: A modelling example

Peter Osmon
King's College, London

Throughout the last century the mathematics of the continuum underpinned the science and technology of the developed world. Today's developed world is increasingly dominated by the artefacts and processes of information technology and it is discrete mathematics that underpins this technology. A finite state machine description of the behaviour of vending machines, in the form of state transition diagrams and state transition tables, is used as an example to demonstrate that modelling numerous artefacts of today's everyday world would be within reach of many 15-19 year old learners if the curriculum were to give more emphasis to discrete mathematics

Keywords: mathematics applications, modelling, discrete mathematics, finite state machines, state transition diagram, state transition table.

Introduction

This presentation is one of a series where the overall aim is to make the case for an updated curriculum- one with less emphasis on the continuum and more on discrete mathematics. The argument for this change is essentially that while during the nineteenth and twentieth centuries continuum mathematics underpinned the science and technology of the developed world, now in the twenty-first century our civilisation is becoming IT-dominated and the mathematics that underpins it is discrete. Moreover mathematics performs this underpinning role through *modelling* and this is what applied mathematics should mean in the curriculum.

To elaborate this: looking for patterns and building models with them is how we understand the world around us. Mathematics is the science of patterns, and so can help with model-making and hence with our understanding of the world. Applied mathematics is model-making and using in the context of either the everyday world or some professional discipline such as science or engineering. But learners in school have limited knowledge of (a) mathematics (b) application domains (principally science and their everyday world), and these limitations narrow the range of models they can hope to appreciate.

However, with respect to their limited mathematics knowledge, quite a lot of the relevant discrete mathematics (sets, relations, logic, events, algorithms, sequential machines) needed for understanding the behaviour of typical artefacts and processes of our everyday twentieth-century world could, with a reformed curriculum, be within reach of learners aged 15-19. In today's mathematics curriculum, discrete mathematics does not emerge as a distinct branch of the subject until university. The topic Finite State Machines (FSMs) is an example of this accessible mathematics. See for example Rosen (2007, 796-798).

An example of model-making in today's everyday world

Vending machines of all kinds are part of our everyday environment. If we ignore the detail of what a particular instance dispenses, it is clear that they exhibit similar behaviours: there is a common behaviour pattern or at least a common family of patterns. In what follows I aim to show that describing these behaviour patterns is potentially within the reach of school mathematics. Finite State Machines (FSMs) is the particular discrete mathematics topic needed for describing vending machine behaviour. Figure 1 outlines the process of modeling the behaviour of a vending machine by designing an appropriate FSM.

Before getting further into the example I should introduce the term "state". State is an intuitive concept that helps us understand the behaviour of entities, usually systems, over time. Thus we speak of the state of the weather, of the economy, of London's transport network, of our health. "State" can be described mathematically. A familiar example is the parabolic trajectory of a projectile subject to vertical acceleration due to gravity. Its state at any moment is described by values of its position and velocity variables (x,y,vx,vy). The projectile has a *continuum* of states. Note that I have chosen not to use mathematical subscripts, considering instead that the abbreviated state name vx is more appropriate for learners.

Now consider the behaviour of vending machines- these familiar entities have sets of *discrete* states- rather than a continuum like the projectile. This kind of behaviour is characteristic of the artefacts and systems in the IT dominated world in which we all now live.

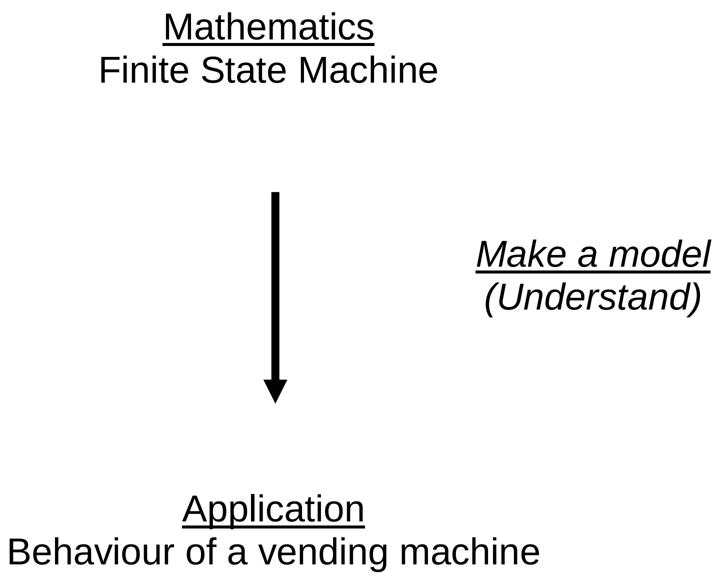

Figure 1. Designing an FSM to model the behaviour of a vending machine.

FSM notation: State Transition Diagrams (STDs)

An STD is a bubble and arrow diagram that describes the behaviour of an entity over time. Bubbles represent states which are given names. The entity has a finite set of states S = {S0, S1, S2, etc}. By convention, S0 is the initial state. When speaking generally we call the current-state Scs. State names are written in the upper half of the bubble. Characteristic of a state is its set of outputs O. A state's output is written in the lower half of the bubble.

Arrows show the possible transitions from Scs to a next-state Sns. A state has a set of Inputs I = {I1, I2, etc} to *choose* which of several possible transitions occurs. An arrow is labelled with the particular input which selects that transition. That is there is a next-state function: Scs x I = Sns. (Understanding FSM models may be helped by assuming that States have duration and Transitions are instantaneous- the mathematics has nothing to say about such matters.)

Figure 2 aims to clarify STD notation. It shows a state bubble and transition arrows into the state, from possible previous-states, and out of the state, into the various possible next-states.

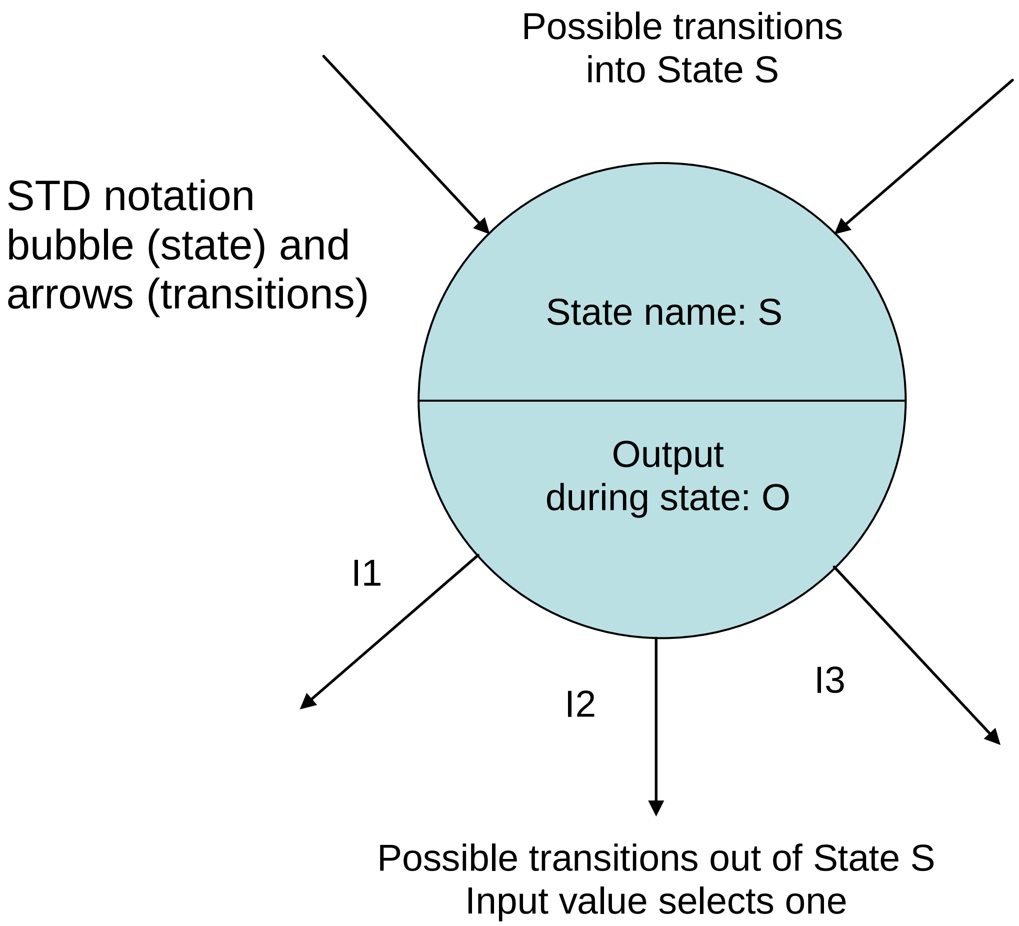

Figure 2. STD notation: a state bubble- containing the State name and Outputs during that state- and possible Transitions in and out of that state with the Input values that select them.

FSM application example: drinks vending machine

Now, to be specific, consider drinks vending machines as our everyday example, and first consider a very simple machine that accepts 50p coins and offers a choice of two drinks: Cola or Orange.

This machine has four states, as follows.

S0. Initial state: Waiting

Output: “Insert 50p coin”

Input of coin- causes transition to-

S1. Choosing drink

Output: “Cola/Orange”

Choice of Input buttons- causes a transition to ***either***-

S2. Dispensing Cola *or* S3. Dispensing Orange
Output: "Please wait"
Input: dispensing finished- causes transition to-
S4. Drink is ready
Output: "Take drink"
Input: removal of drink- causes transition back to S0

From this information we can construct the STD for this simple machine. It looks like the diagram in Figure 3.

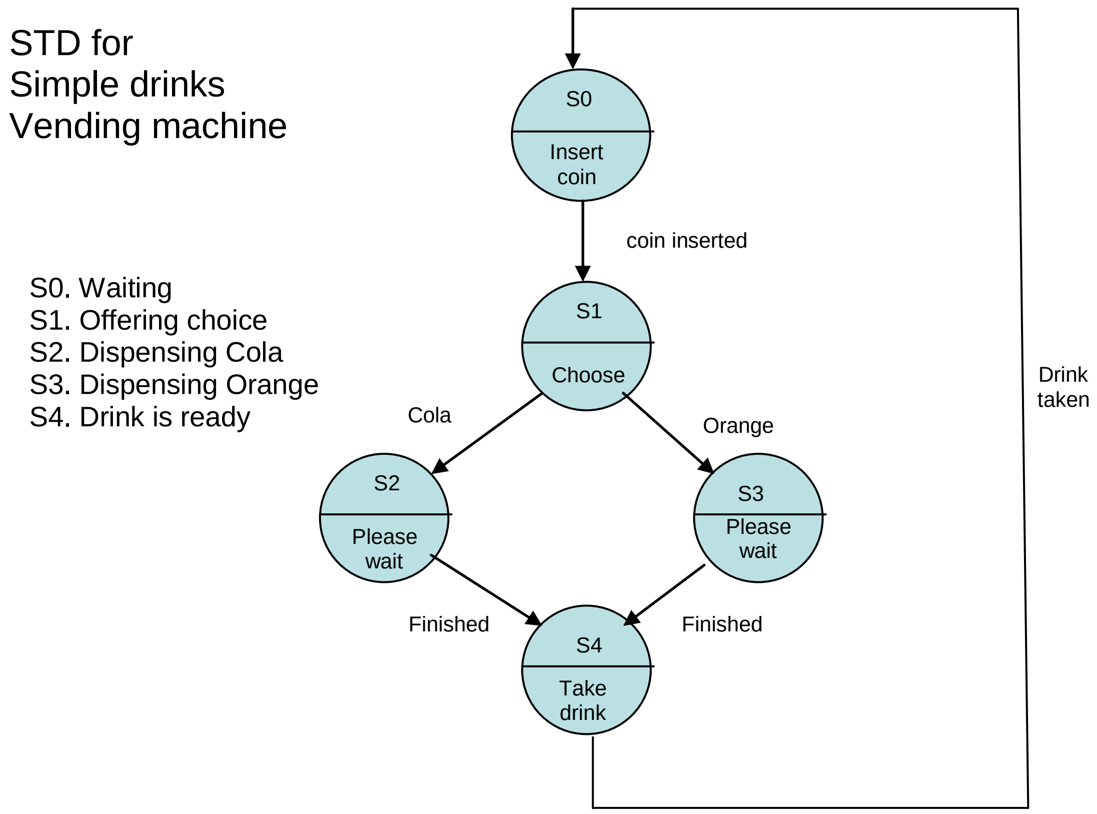

Figure 3. STD for the simple drinks vending machine

A State Transition Tables (STT) is an alternative notation for describing FSM behaviour. The Table below describes the STT for the simple drinks machine. As the table demonstrates, a STT is actually two tables: the output table and the next-state table, corresponding to the machine's output function and next-state function respectively.

Current-state	Output	Input/Next-state
S0 Waiting	Insert coin	Coin-inserted/S1
S1 Offering choice	Choose	Cola/S3 Orange/S4
S2 Dispensing Cola	Please wait	Finished/S4
S3 Dispensing Orange	Please wait	Finished/S4
S4 Drink is ready	Take drink	Drink-taken/S0

STTs are more compact than STDs: they can describe, on a single page, behaviour with more states and transitions. While it is generally harder to comprehend behaviour from a tabular description, drawing large and complicated STDs can become tedious, even with the aid of special software.

A more elaborate vending machine: more functionality and more states

Now consider a drinks vending machine with more functionality: drinks still cost 50p each, but this machine accepts 5p, 10p, 20p, and 50p coins, and gives change. Further, the machine offers a choice of five hot drinks - tea, coffee, strong-coffee, mocha, chocolate - and also offers choices of additives - unsweetened/sugar/double-sugar and black/milk/double-milk. When it comes to describing its behaviour with an FSM, this means not just more states but more complicated connections- quite a lot to get one's head round. How to proceed?

Our vending machine problem provides an opportunity to introduce the following two heuristics which are helpful in many problem solving situations (Polya 1945):

(A) Divide-and-conquer

"Factorise" the problem into parts: a *Payment part* and a *Drinks-and-Additives-Choices part*.

(B) <u>Easier-problem-first</u> (applies to both parts):

Payment part: let the complications in progressively. We have already considered a 50p coin only machine, so now accept a range of coins- but no change given, and then, at the next stage, give change.

Choices part: let the complications in progressively. We have already considered a binary choice machine, now provide a five-way choice of drinks, and finally introduce two levels of additives choice.

Following in the tradition of mathematics textbooks, the task of working out a description of the more elaborate vending machine- as either a STD or a STT- is left to the reader.

Some other applications of finite state machines

Vending machines in today's world dispense a great variety of goods and services besides drinks- perhaps the most common is the automatic teller machine (ATM) or "hole in the wall" outside banks that dispenses money, or one's bank account information, in response to input information supplied by a magnetic strip on one's debit card, supplemented by choices input by keypad.

ATMs differ from most vending machines in that they are not self-contained within a cabinet- not localised- but rely on electronic communications with a, generally remotely located, bank database. Perhaps this also behaves as a finite state machine. Communication between a pair of FSMs- the output from one providing the input for the other and vice versa- is a more challenging modelling problem.

Vending machines of various kinds are by no means the only applications of finite state machines in our everyday world. Other examples include control of traffic lights, lifts, and traditional combination locks. In a rather different context- word processing- FSMs can perform syntax checking. A problem example that demonstrates this application, while not taking too long to work out, is devising an FSM that checks that every left-hand bracket in a sentence or mathematical expression containing nested pairs of brackets has a matching right-hand one.

My experience presenting a range of such examples to first year undergraduates, who have no more than GCSE level mathematics, is that they get quite interested in the problems and can manage to solve them; also, that this is a class of problems that seem well-suited to collaborative group working.

Conclusions

This paper is about meaningful teaching of applied mathematics to 15-19 year olds. I have argued that this means teaching modelling and that, to make models, learners need knowledge of some application domain- and in practice this has to be either science or else the everyday world- together with the relevant mathematics.

During the nineteenth and twentieth centuries, continuum mathematics (calculus especially) underpinned the classical physics base of much nineteenth and twentieth century industry as well as many everyday artefacts. Now, in the twenty-first century, it is becoming apparent that today's business and industry as well as today's everyday artefacts and processes, rest on information technology- which in turn is underpinned by discrete mathematics (sets, logic, relations, algorithms, events, sequential machines) rather than the formerly dominant continuum mathematics. But in today's mathematics curriculum, discrete mathematics does not emerge as a distinct branch of the subject until university.

In this paper, by focusing on a particular topic in discrete mathematics, namely finite state machines, I have sought to demonstrate that, were the curriculum to be reformed to give appropriate recognition to the contemporary importance of discrete mathematics, there are many familiar everyday artefacts and processes that would then be accessible to younger learners to model.

Reference

Rosen, K. H. 2007. *Discrete mathematics and its applications*. Sixth edition. New York: McGraw-Hill,

Polya, G. 1945. *How to solve it*. Princeton: Princeton University Press