# Fickle Fruit: Finite-State Machines

Teacher Notes to support Tenderfoot Unit 5: Theoretical Computers – Fun with finite-state machines

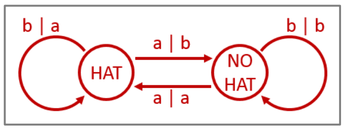
A finite-state machine is not really a mechanical entity, but an abstract set of instructions which a computer can be programmed to follow precisely. The activity, from the MathManiaCS project encourages children to reason about what they observe. Finding ways to record observations naturally leads to State Diagrams and the notion of a Finite-State Machine. More formal exercises are included to develop techniques illustrated in the activity.

**Preparation required:**
For each group: A hat and scarf, Fickle Fruit Vendor Instructions, 6 apples and bananas (or pictures)
For each pupil: Fickle Fruit Class Exercise and Solutions, Happy Robot Input Rules, rough paper
Follow up exercises: Designing Finite-State Machines - Hair Dryer and Vending Machine exercises
Extension: The Perfect Pizza story (all can be provided electronically)

## Fickle, Frustrating and Fancy Fruit

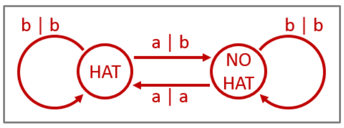

Start the exercise as a class, with one volunteer at the front as a fruit vendor. The vendor should wear a hat. Give the volunteer the Fickle Fruit Instruction card. No-one else should see it. Instruct students to request (in an orderly fashion) either a banana or apple. The vendor responds according to the instructions on the card. Encourage students to keep some record of their observations. When familiar with the routine, split into groups with one student as fruit vendor. They can take it in turns as the activity progresses.

In groups, with the fruit vendor in possession of the Fickle Fruit Instruction card only (and starting with a hat on), challenge students to place orders for *three consecutive apples*. If they get the answer quickly (banana, banana, banana), repeat the exercise with the vendor starting without a hat. Do they fully understand the vendor's behaviour? Try ordering *two bananas then an apple*, starting with the vendor in their current attire. Then try *two apples then a banana*. Bring to a close by asking groups to predict the behaviour of the vendor and share any methods they have used to record the behaviour. Has anyone drawn a state diagram? Swap vendors and use the Frustrating Fruit instructions. They can start with or without the hat. Issue the same challenges. Eventually, someone will realise that you cannot place an order for 3 consecutive apples. Stop the exercise and ask if they are sure?

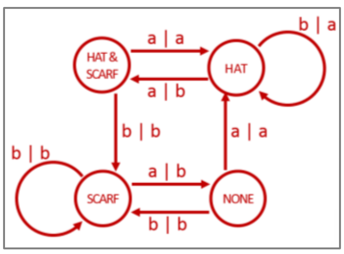



At this stage, we need an organised way of characterising the vendor's behaviour. A slide demonstrates the creation of a state diagram for the **Fickle Fruit** instructions. There are two states for the vendor: Hat and No Hat. We can represent the transition with an arrow. We'll use 'a' and 'b' to represent apple and banana. We can write the input and output on the transition, using a pipe to separate them. So a|b means we asked for an apple and received a banana. The presentation allows this to be completed as a class through questions and answers.

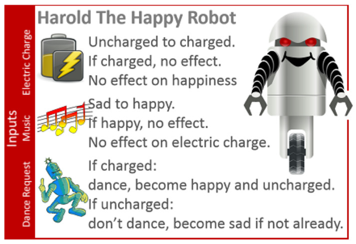
The Fickle Fruit Class exercise asks students to draw the diagram for the Frustrating Fruit vendor. Can the students explain why a three apple order is impossible? The answer is in the presentation. Note that a state diagram is completely deterministic. Every state must have a transition for every possible input. In this case, there are 2 states and 2 possible inputs, so four transitions.



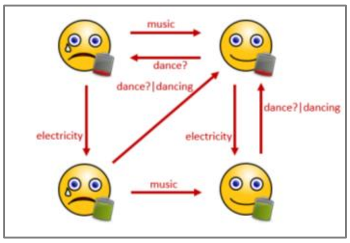

The final exercise is more challenging. The Fancy Fruit vendor has both a hat and a scarf. Challenge students to complete the state diagram on the exercise sheet. As they don't know the possible states in advance, complete a rough draft first. The solution (on a slide) illustrates a further issue. The transitions outgoing from the Hat & Scarf state are identical to those from the None state. This is an example of a redundant state. The diagram can therefore be simplified, collapsing the two states into one.



# Harold The Happy Robot



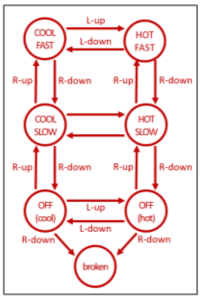

Once familiar with state diagrams, as a group design a FSM from a given specification - the behaviour of "Harold, the happy robot". At any time, Harold is either **happy** or **sad**. He is also either **charged** or **not-charged** with electricity. There are three types of **inputs** that Harold might get: electricity, music or a 'dance' request. The rules that describe Harold's behaviour are fairly natural. They are reproduced on Happy Robot Card, so each student can have a copy.

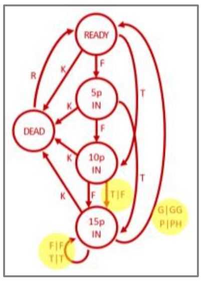



Deciding the states is the key starting point for state diagrams. Is dancing part of a state or is it an output? We'll make it an output (the only output), in response to a dance request input. So any output is associated with the transition, like the Fruit diagrams. Guide discussion towards discovering the states, with 'dancing' not part of any in this model: Sad / Uncharged, Happy / Uncharged, Happy / Charged and Sad / Charged. Discuss transitions from each in turn. The diagram builds in stages, question prompts given in the slide notes.

# Designing Finite-State Machines

Once students have grasped the principles they could investigate everyday appliances. There are two exercises, a hair dryer and vending machine. In the case of the hair dryer, point out that there are no outputs on the transitions. The transitions represent the input (L-up/down or R-up/down). As the output is constant, it makes sense for the state to represent the output e.g. blowing cool air fast. There are two conventions for representing FSM's. Representing *output on the state*, as we did with the hexahexaflexagon, and are now doing with the hair dryer, is known as a Moore machine. Representing *output on the transition*, as we did with the Fickle Fruit machines is known as a Mealy machine. Whilst this detail is not needed at KS3, it does feature at A level. A state diagram will use one or other representation, the choice usually dictated by the task it represents. It is possible to express any Moore machine as a Mealy machine, and vice-versa.




The vending machine is more challenging. Discussion about representing output is helpful. The output is a discrete drink, before returning to a ready state so best represented on a transition – a Mealy machine. Deciding on states and types of machine can be difficult, often requiring intuition that comes from experience. Be prepared with some good hints. The solution to the Slush Dispenser is shown. To check understanding the slide animation highlights transitions showing output to use as discussion prompts.

FSM's can be very useful for modelling the design of interfaces. Homework might develop diagrams for setting the time or alarm on a digital watch. By modelling an interface, design errors can quickly be identified. If the FSM that describes a device is complicated it is a warning that the interface will be difficult to navigate. This is a huge area in Computer Science: HCI or Human Computer Interaction. Cs4fn host a video about setting microwaves (goo.gl/Bv4LZU), pointing out implications for interfaces in serious settings, such as hospitals. A reprint from SwitchedON, the CAS magazine, included in the resources looks at how software interfaces can be expressed as a FSM. It highlights the importance for software developers in separating the processing requirements from the interface behaviour and includes suggestions for further exercises.

One final, challenging exercise is also included in the resources. From the now defunct MegaMath website, the Perfect Pizza is a story in which the behaviour of the Babuie, the pizza maker can be explained by a finite -state machine. Slide notes include discussion prompts. A last suggestion, Manufactoria, (goo.gl/B28pKZ) is a very challenging game based on finite-state machines - probably one for teachers, rather than students!