Australian National University

Intellectual Property Overview

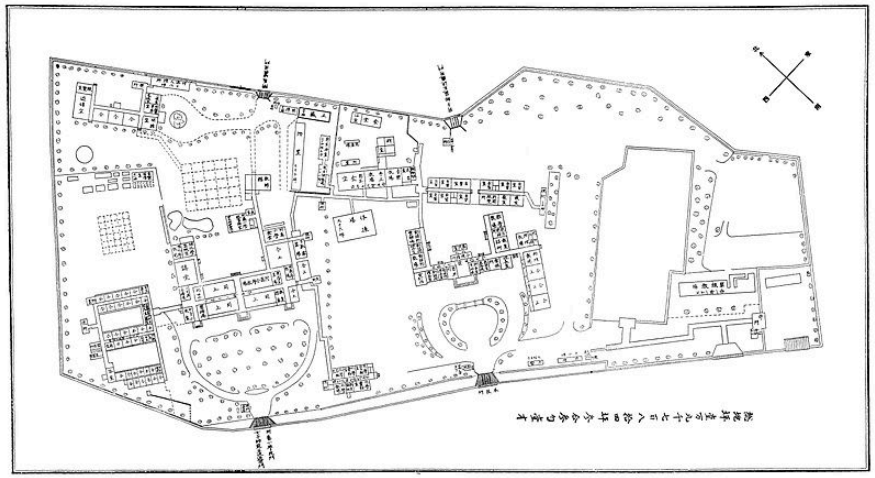
Michael Curtotti

Overview

- What is Intellectual Property
- The idea of property
- Examples and characteristics of property
- Extension of the notion of property to creativity and invention
- Major flavours of intellectual property: copyright, patents, trademarks
- IP within the ANU environment

What is property?

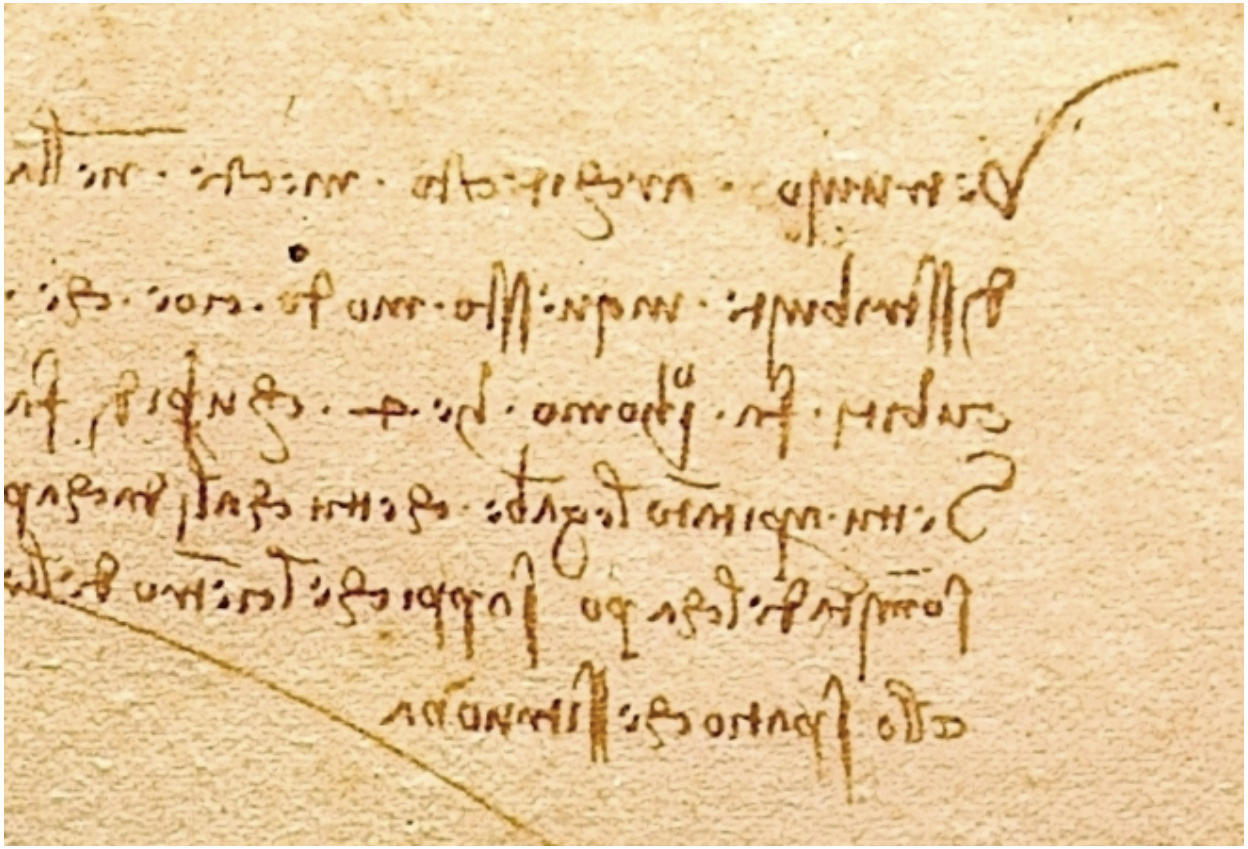
- We can think of types



We can think of characteristics & operations

- Owner
- Some kind of boundary – or way of delimiting it from other property
- You can't own air unless you put it in a bottle
- Certain operations you can carry out on property
- Use it
- Give/sell (transfer it to another owner)
- Lease/licence it

Before Intellectual Property



As Jefferson would put it:

If nature has made any one thing less susceptible than all others of exclusive property, it is … an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, … and like the air in which we breathe, … incapable of confinement or exclusive appropriation. Inventions then cannot, in nature, be a subject of property.

How to put an idea in as bottle: Intellectual Property

- We abstract the idea of property from physical things and apply the characteristics and functions of property to the “products of the mind”
- Still true that intellectual property does not give anyone property in an “idea” per se.
- IP comes in several flavours which are quite different share the properties/functions of property discussed above

Primary types: Copyright, Patents, Trademarks

	Copyright	Patents	Trademarks
Boundary Condition	The way an original idea is expressed	The way of achieving a useful result	A word, image, sound, smell, colour
Default owner	Creator	Inventor	Person registering the trademark

IP in software

- In fact can have complex IP environment even in a single piece of software
- Who owns images? Who owns the look and feel?
- Who owns the code?
- What about external libraries?
- What about the name of the software? Who owns that? Is the name free to be used?
- Is any part of the software patented or patentable?

Intellectual Property in the ANU Environment

- ANU Intellectual Property Policy
- By default students own their own IP but …
  - If you are working on a project with ANU employees your IP may get mixed together with ANU IP
  - If your project gets external funding you may be asked to assign IP
  - Under the ANU IP policy can share in benefits of commercialisation (maybe)

Next

- In the next lectures we will begin to explore specific types of intellectual property in more detail

Copyright

Michael Curtotti

Overview

- What is Copyright?
- What does it protect?
- Kinds of copyright matter
- Examples of copyright matter in software
- Practical protection of copyright
- Licensing of copyright (kinds of licences)

What is copyright?

- Copyright is the exclusive right of the owner of copyright in copyrightable matter to do the things that copyright allows you to do (does that help?)
- The core exclusive right is the right to “copy” but that doesn’t really give a full sense of the rights of a copyright holder
- Copy, publish, adapt, licence, import, broadcast, perform in public

What copyright does & doesn't protect

- Doesn't protect ideas
- Protects an original way an idea is expressed (for example the way words are used in a poem or the way a textbook is written).
- Copyright doesn't prevent a person understanding an idea and writing it in their own words

Kinds of Copyright Matter

- Literary Works – this includes books, plays, poetry, furniture catalogue, it also includes tables (think data) and it includes computer code (source code and object code)
- ***"computer program "*** means a set of statements or instructions to be used directly or indirectly in a **computer** in order to bring about a certain result. (Copyright Act)
- But also includes anything essential to operation of the program

More kinds of copyright matter

- a painting, sculpture, drawing, engraving or photograph, whether the work is of artistic quality or not;
- a building or a model of a building, whether the building or model is of artistic quality or not
- a work of artistic craftsmanship whether or not mentioned
- Dramatic work, including choreography and scenario or script for a film
- Publication of copyright matter (published work)
- Sound recordings, films, broadcasts

Ownership and Duration

- Default is that the author of copyright matter owns it, but employer owns material created by employee in course of employment
- Essential to know who owns copyright
- Most types of copyright last for the life of the author plus 70 years.
- J.R.R. Tolkien died in 1973 – copyright on the book The Hobbit runs out in 2043.
- The movie has separate copyright. The script for the movie also has separate copyright.

Some Exclusive Rights of the Owner

- to reproduce the work in a material form;
- to publish the work
- to perform the work in public;
- to communicate the work to the public;
- to make an adaptation of the work;
- in the case of a computer program, to enter into a commercial rental arrangement in respect of the program.
- (infringement of copyright involves some kind of copying – i.e. it is conscious)

Moral Rights

- Not really copyright but live in the Copyright Act
- They are the right
  - To be attributed as the author of a work
  - Not to have a work falsely attributed
  - Not to have the work treated in a derogatory fashion (i.e. in a way that diminishes the reputation of the author)

Fair Dealing – Important Qualifications to Copyright for university students

- Fair dealing for purposes of study or research
- Factors: (a) was it for study and research (b) the nature of the work (c) the possibility of obtaining the work quickly at ordinary commercial price; (d) effect on the market of value for work (e) how much was copied.
- See section 40 for details
- Fair dealing for the purposes of criticism or review / parody or satire
- Reporting the news

Examples of copyright in computer software

- Computer program (the code)
- Data
- Text files part of the program
- Images
- Sounds
- Films
- Layout design

Licensing & other commercialisation options

- Licensing is how copyright is often dealt with (commercialised)
- Can licence the right to use the software (e.g. purchased copy of old versions of microsoft office)
- Can provide the software as a service (e.g. newest versions of microsoft office)
- Can of course sell/transfer

Australian National University

Patents

Michael Curtotti

Overview

- What are patents?
- What do they protect?
- Kinds of patentable material
- Examples of patents in hardware and software
- Licensing and commercialising patents

What are patents

- Patents are a registered form of intellectual property which protects the right of an inventor of a patentable invention to exploit the invention
- Balance between encouraging inventors to innovate in the hope of future profit and not creating unreasonable monopolies which hold back diffusion of innovation
- Patents give an exclusive right to exploit – generally for a period of 20 years
- Can be registered in different jurisdictions

What kinds of things can be patented

- a new manner of manufacture
- that is novel as compared with the prior art base and
- involves an inventive step and
- is useful and
- was not secretly used in the patent area
- (Patent law highly technical and only patent attorneys can prepare a patent application)

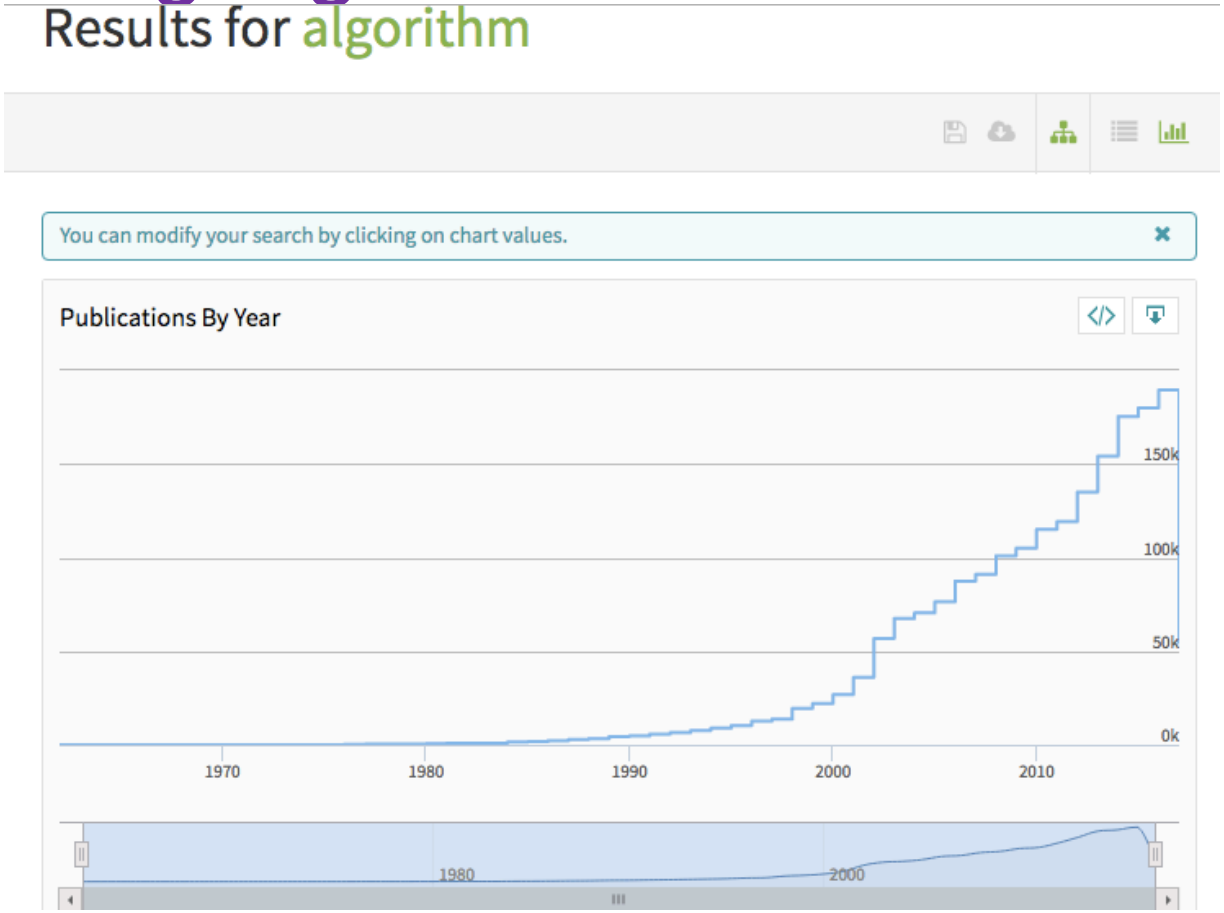
Examples

- Biotechnology (but can't patent a human being or part of a human being)
- Software - generally mathematical algorithms not patentable but an application of a mathematical algorithm can be patentable (is it economically useful)
- Business methods

Software

- Huge numbers of software patents issued every year
- Patent trolls aggregate patents in an effort to find a large target to sue
- Large companies hold huge patent portfolios to protect themselves from hostile patent claims
- There is controversy about software patenting

Number of patent documents containing algorithm



Why patent?

- IP gold – but expensive to get
- Investors are looking for exclusivity (and safety against hostile patent claims)
- Not realistic without external investment for most start ups

Publish, Perish and at your own risk

- If you use or publish your patentable idea – its no longer patentable
- It doesn't matter if you don't know there is a patent – if you infringe you are still liable

Commercialisation & Licensing

- Patents can be assets within a company
- Shares in a company can then be sold and increase in value as the patent is commercialised
- Patents can be licensed to downstream suppliers who pay the inventors/owners for the right to use a patent in a particular context (e.g. a territory or a particular market)

Australian National University

Trademarks

Michael Curtotti

Overview

- What are trademarks?
- What do they protect?
- Examples and kinds of trademarks
- Practical trademark protection
- Licensing trademarks

What are trademarks

- Trade marks are a registrable form of intellectual property protecting the name or identity of a business or product/service
- A trade mark can be:
  - A word
  - An image
  - A combination of word and image
  - A sound
  - A colour
  - A smell

Some familiar examples

- Trade mark above and:

Number
1283373

Words
UNIVERSITY HOUSE

Image description
GROUP OF MEN IN VARIOUS POSES IN TROPHY CUP

Status
● Registered/Protected

Priority date
04 Feb 2009 (Lodgement)

Classes
9, 14, 16, 25, 26, 43

Words
ANU

Image description
STARS, BOOMERANG & STRIPES WAVY ON SHIELD ATOP SCROLL

Status
● Registered/Protected

Priority date
02 Aug 2001 (Lodgement)

Classes
14, 16, 25, 35, 38, 41, 42

Kind
Word, Device

Number
851016

Words
MoodGYM ; Mood Gym ; MOODGYM

Number
1293511

Words
AUSTRALIAN NATIONAL UNIVERSITY

Status
● Registered/Protected

Priority date
08 Apr 2009 (Lodgement)

Classes
41, 42

Number
1293510

Words
ANU

Status
● Registered/Protected

Priority date
08 Apr 2009 (Lodgement)

Classes
41, 42

Kind
Word

What do they protect

- The right to use the trademark in the course of trade or commerce
- Cannot use any deceptively similar mark in trade or commerce
- Doesn't prevent use of mark in other contexts – e.g. use of the words "Australian National University" in an academic paper.

Some other characteristics

- They are about turning the goodwill of an enterprise (which shows up as an asset in accounting) into property as far as the law is concerned
- They last ten years – but can be renewed indefinitely
- Need to be registered in different jurisdictions
- Their value can represent a significant proportion of commercial value – e.g the brand Coca Cola (for coloured flavoured sugar water)

Practical Trade Mark Protection

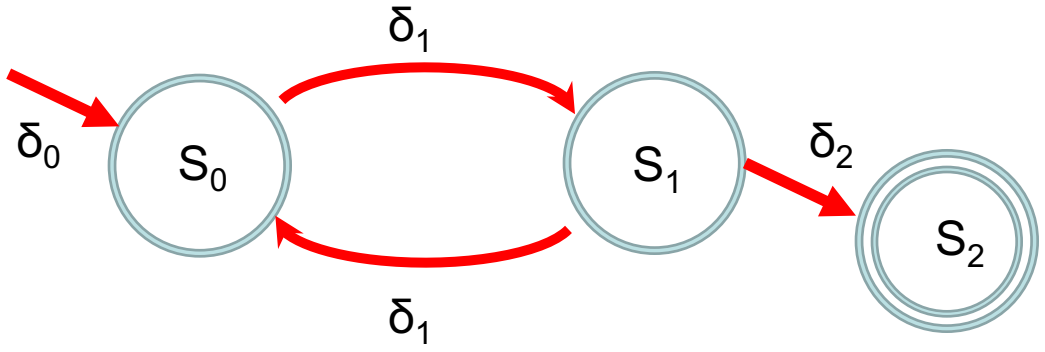
- Fairly easy to arrange your own trade mark.
- Can seek the help of a trade mark attorney or lawyer
- IP Australia has a searchable database of trademarks
- As long as someone hasn't already registered the mark you can register it in Australia
- Nowadays often go hand in hand with obtaining an unique domain name (an separate process)

What do you do with them

- Can use directly on your own products or services
- A common use case is licensing others with the permission to use a trade mark
- (e.g. subcontractors)
- Use it or lose it

Australian National University

Finite State Machines - Introduction



Michael Curtotti

Overview

- Why use Finite State Machines?
- What are they?
- Deterministic and Non-Deterministic Finite State Machines & Turing Machines

Why use Finite State Machines?

- We are primarily interested in FSM's as a programming construct.
- They are a useful representation for many problems addressed by computer science including hardware
- Easy to code
- Intuitive
- Flexible

Formal Definition

- Without output - quintuple defined by

  $FSM = \{ \Sigma, S, s_0, \delta, F\}$

- Where:

  Σ = a finite non-empty set of input symbols

  S = a finite non-empty set of states

  s_0 = the initial state of the FSM

  δ = a transition function $\delta: \delta \times \Sigma \rightarrow S$

  F = a finite set of final states (can be empty)

- With output 2 additional elements: Θ (set of output symbols) and $\lambda : S \rightarrow \Theta$

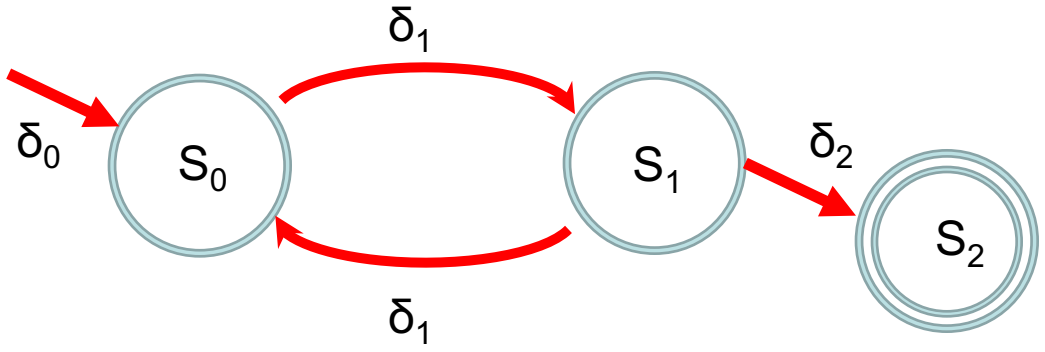
Deterministic and Non-Deterministic Finite State Machines

- For deterministic FSM each transition δ can lead to only one new state S.
- Non-deterministic FSM a transition δ results in more than one possible state for a given input.
- $\delta: \delta \times \Sigma \rightarrow \boldsymbol{P}(S)$
- NFSM can be translated to DFSM but may result in a much larger DFSM
- Turing Machine and RAM can be modelled with FSMs. Turing Machine can be thought of as two interacting FSMs – tape and control unit

Some useful references & next

- References:
- Ch. 3 & 4 of John E. Savage, Models of Computation, cc http://cs.brown.edu/people/jes/book/
- Jurafsky & Martin, Speech and Language Processing, 2nd edition, sections 2.2 & 2.3
- Russell & Norvig, Artificial Intelligence a Modern Approach, 2nd Edition, pp 64 et seq
- Next:
  - concrete examples of finite state machines and approaches to representing them

Finite State Machines - Examples

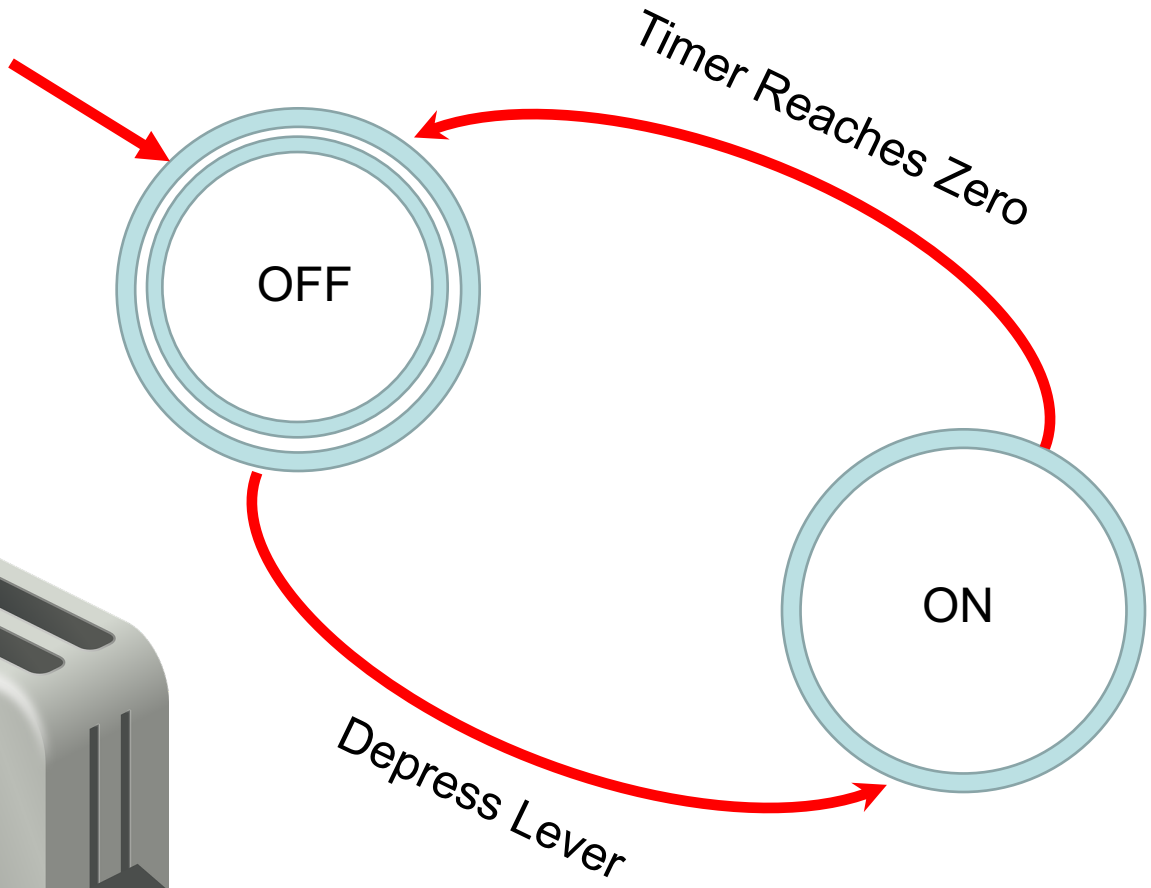


Michael Curtotti

Overview

- So far we've described finite state machines primarily as a mathematical construct
- Look at some examples and their representation
  - Toaster as a finite state machine
  - Tape recorder
  - Vending machine
  - Regular expressions & eight tile puzzle
- Different graphical and tabular FSM representations

Toaster as Finite State Machine



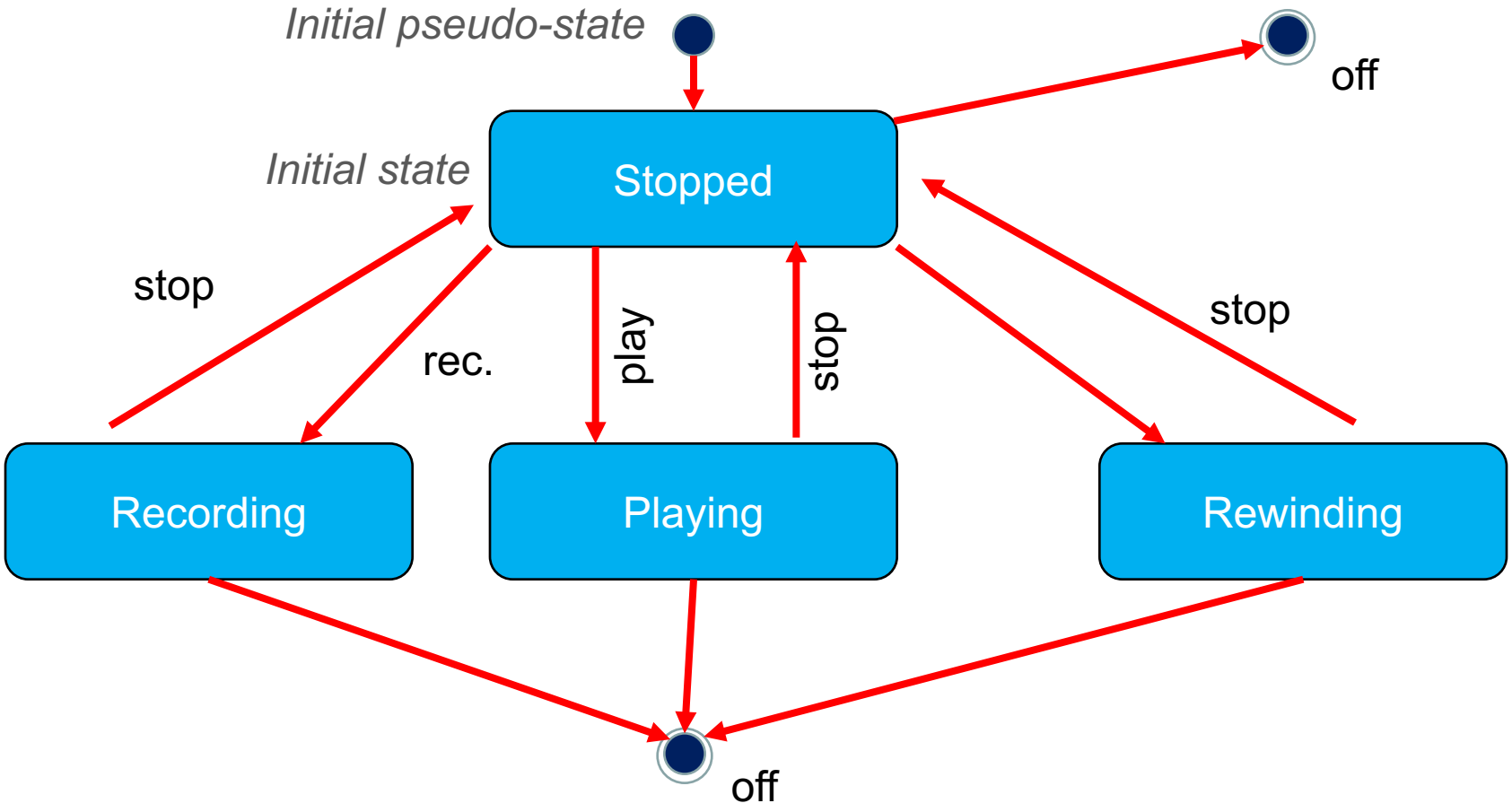
ex. State Transition Table Representation

STATE / INPUT	OFF	ON
Depress Lever	ON	-
TIMER Reaches 0	-	OFF

ex. State Transition Table Representation with outputs

STATE / INPUT	OFF	ON	OUTPUT
Depress Lever	-	ON	-
TIMER Reaches 0	OFF	-	Eject toast

Simple UML State Diagram – Tape Recorder

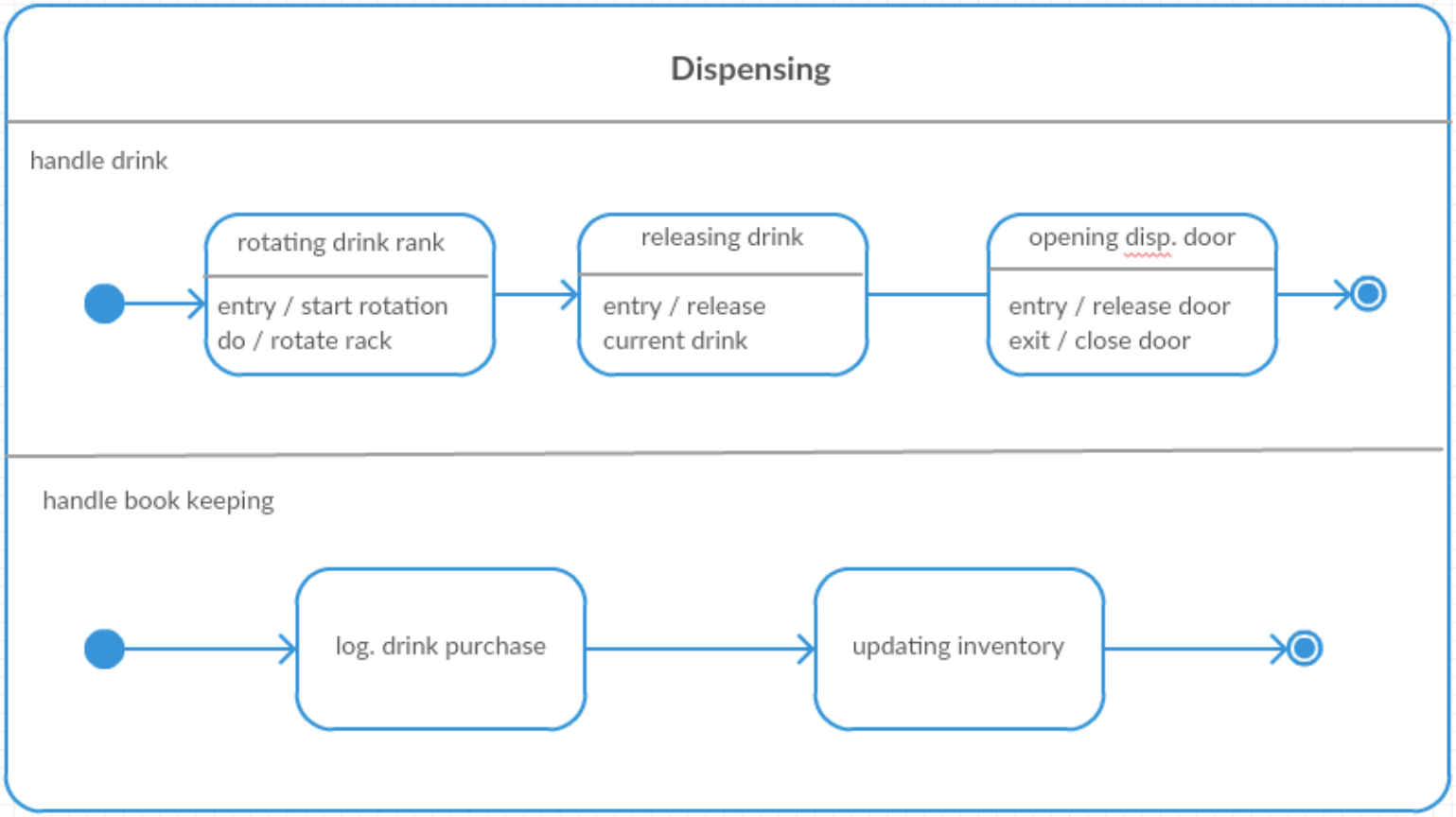


From Christopher Fox, Intro. to Software Engineering Design …, 2007, p 397
(not all inputs/transition strings shown)

Tape Recorder State Transition Table

State / Input	STOPPED	PLAYING	RECORDING	REWINDING
Play	PLAYING	-	-	-
Stop	-	STOPPED	STOPPED	STOPPED
Rewind	REWINDING	-	-	-
Record	RECORDING	-	-	-
Off	-	OFF	OFF	OFF

Vending Machine UML State Diagram

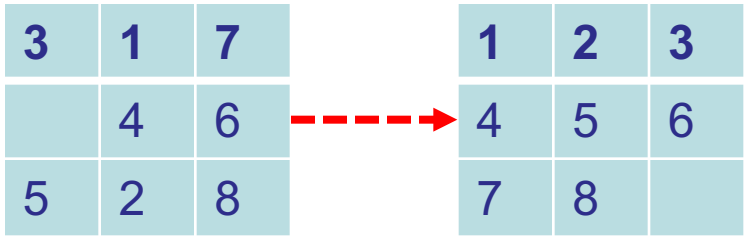


c.f. Dan Pilone, UML 2.0 Pocket Reference Reilly Media 2006, p 97

Regular Expression and FSM

- FSMs and Regular Expressions are Interchangeable
- Regular expression to recognise sheep language (Jurafsky and Martin)

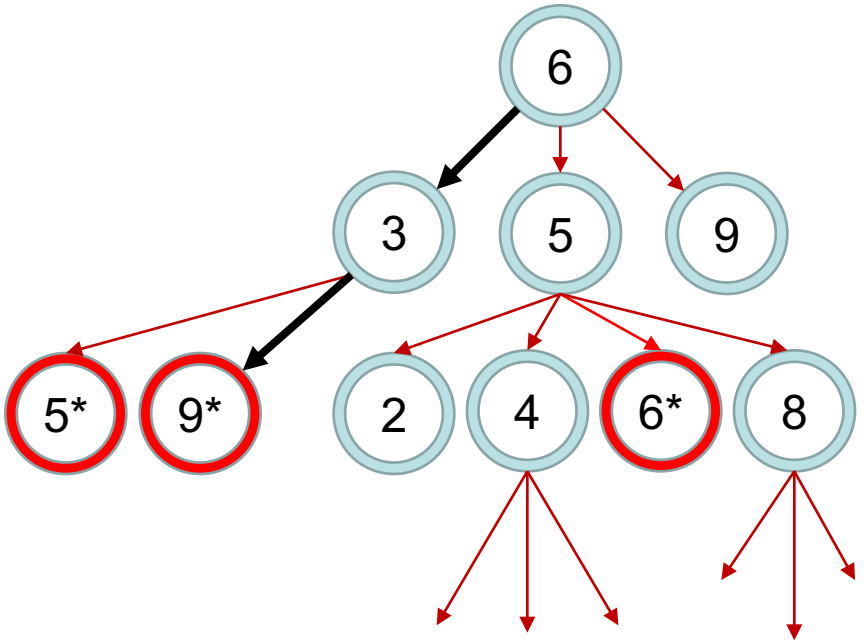
Tile Puzzles and Search Spaces



- 9!/2 ≈181K reachable states
- 15 tile ⟹ 1.3 trillion
- 24 tile ⟹ 10^{25}

- A “toy” problem – but real world states potentially far larger
- How would we represent 8-tile states in a program?
- Could we usefully use a state representation?
- How would we represent possible transitions?
- Tree representation of search space (behaviour trees)
- How would we optimally solve the search problem?

empty	δ	Next - empty
sq1	L,U	[2,4]
sq2	R,L,U	[1,3,5]
sq3	R,U	[2,6]
sq4	D,L,U	[1,5,7]
sq5	D,R,L,U	[2,4,6,8]
sq6	D,L,U	[3,5,9]
sq7	D,L	[4,8]
sq8	D,R,L	[5,7,9]
sq9	U,L	[6,8]



sq1	sq2	sq3	sq4	sq5	sq6	sq7	sq8	sq9
5	6	2	1	3	*	8	4	7
5	6	*	1	3	2	8	4	7
5	6	7	1	3	2	8	4	*
…	…	…	…	…	…	…	…	…

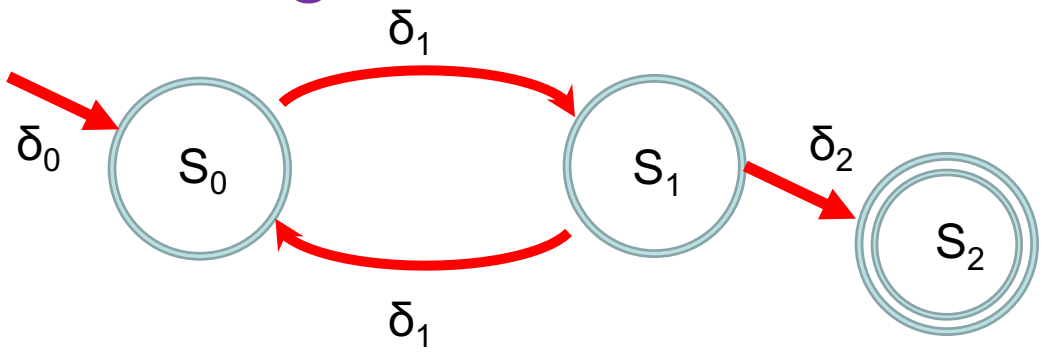
- Breadth 1st vs Depth 1st
- Not repeat seen states
- Limited depth first
- A* search (cost minimisation)

Useful references & next

- References:
  - Dan Pilone, UML 2.0 Pocket Reference Reilly Media 2006
  - Christopher Fox, Introduction to Software Engineering Design, 2007, Chapter 13
  - Jurafsky & Martin, Speech and Language Processing, 2nd edition, sections 2.2 & 2.3
  - Russell & Norvig, Artificial Intelligence a Modern Approach, 2nd Edition, pp 64 et seq
- Next:
  - state design pattern as a useful programming construct in the context of a simple game agent

Australian National University

Finite State Machines – Implementing a State Design Pattern

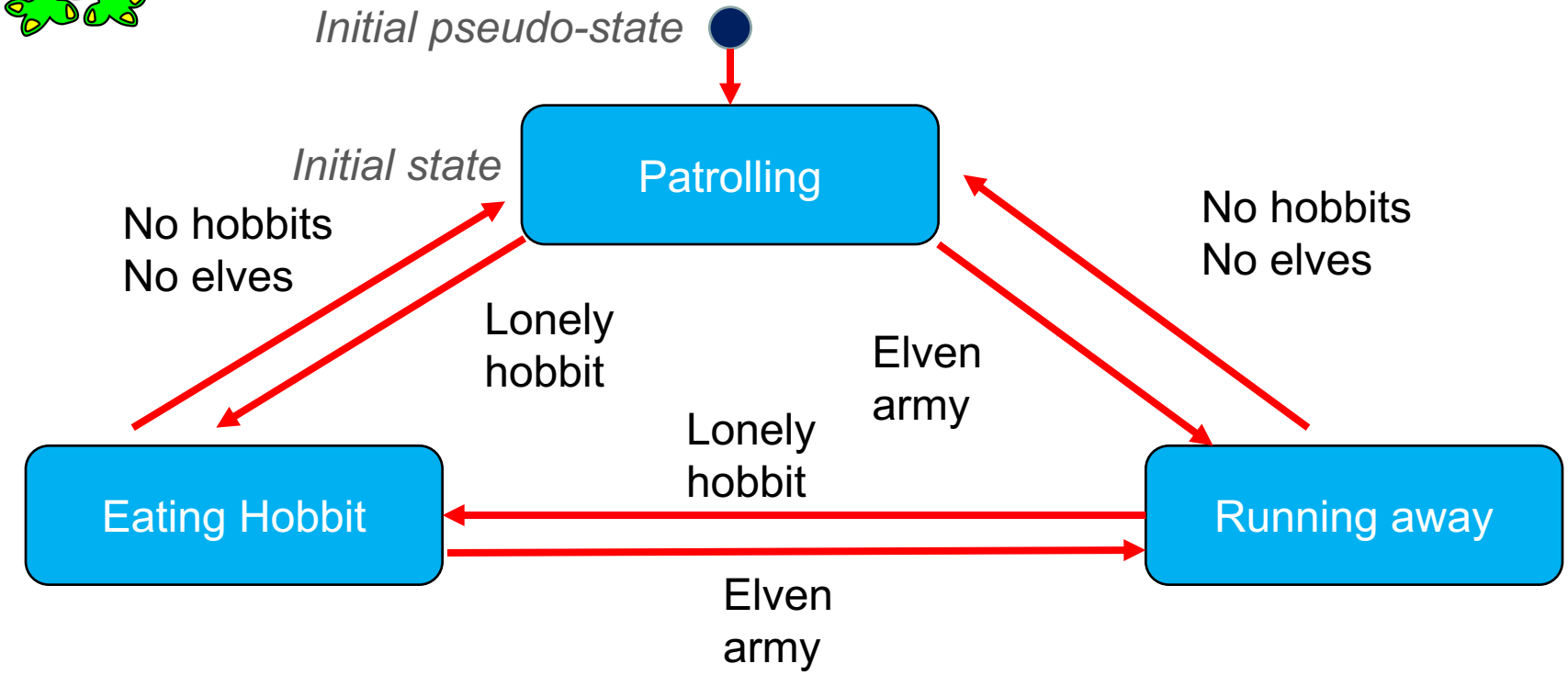


Michael Curtotti

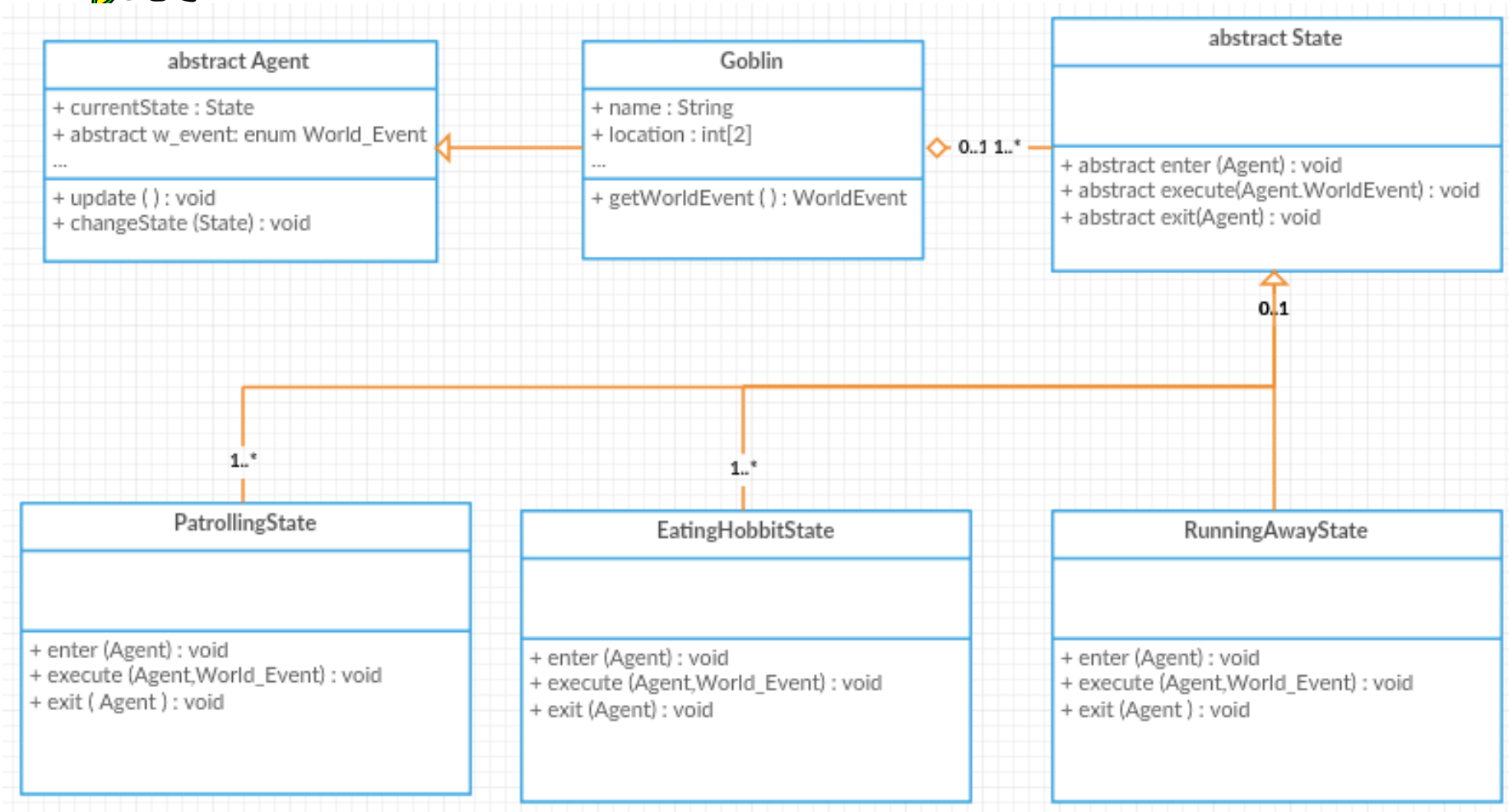
Overview

- Coding a FSM
- Agent based programming example
- Start with a naïve approach
- Simple State Design Pattern implementation
- Discuss extensions of the pattern

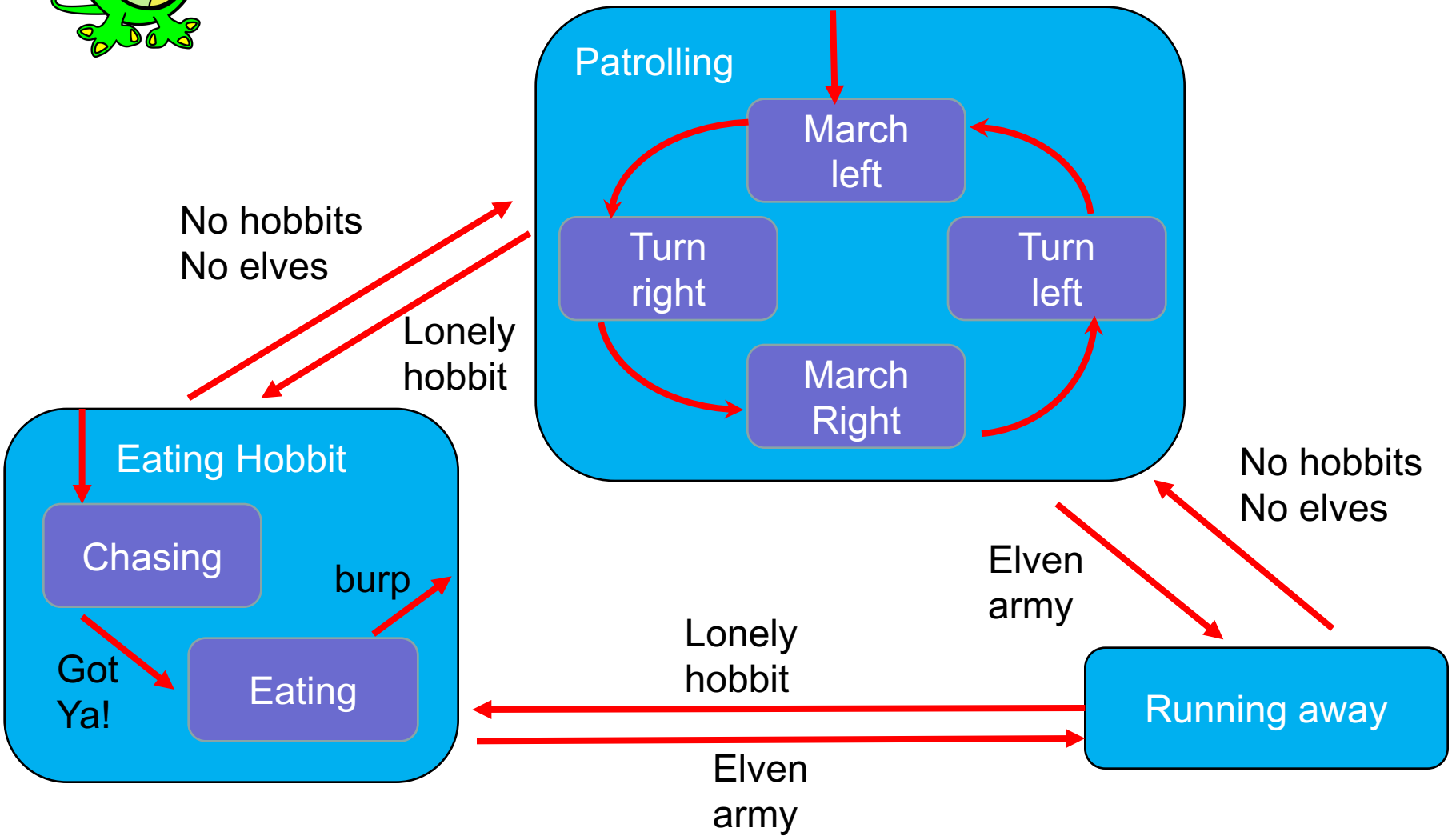
Goblin FSM – UML



Basic State Pattern for Goblins



Hierarchical Goblin FSM



Handling hierarchy and extensions

- One approach - recursive code for state changes
- External state transition table/xml
- Storing previous state(s)
- Using open source FSM library
- Dealing with multiple simultaneous events: elves & hobbits
- Dealing with collective state change: Sauron orders attack on Minas Tirith
- Dealing with internal states of agent (sleep, hunger, health, death)

References

- Matt Buckland, Programming Game AI by Example, Wordware Publishing 2005, pp 43 et seq
- Ian Millington and John Funge, Artificial Intelligence for Games, Morgan Kaufmann, 2nd Edition , 2009, pp 309 et seq