Lect1: Introduction(s)

Tokyo Tech.
Intro. to Comp. & Data
Lecture week1

Brief introduction(s)
to computation and data mining

1. Introduction to computation.

   From Computer Science 1 (undergrad. 1st year course)

2. Introduction to data mining
   (also machine learning and AI).

   From Computer Science 2 (undergrad. 1st year course)

3. On Exercise #1.

1. Introduction to computation

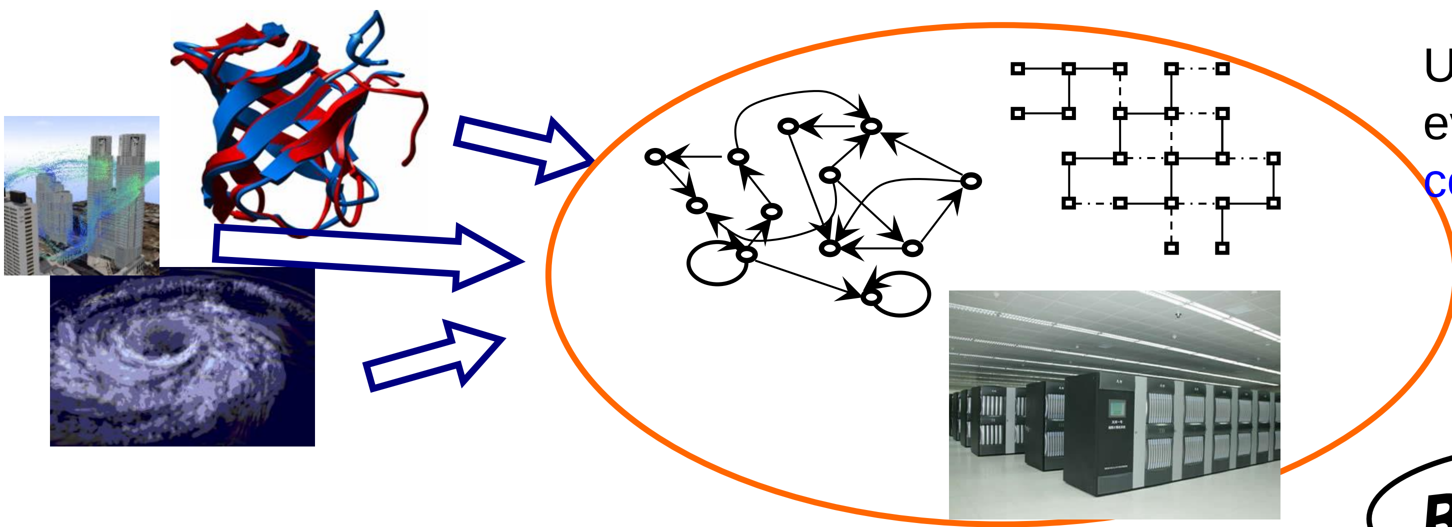
- What is computation?
- Why computation?

Let us start why do we need to study computation.

The importance of ***CompView***.

↑

Computational Approach to XXX



Understand everything through computation

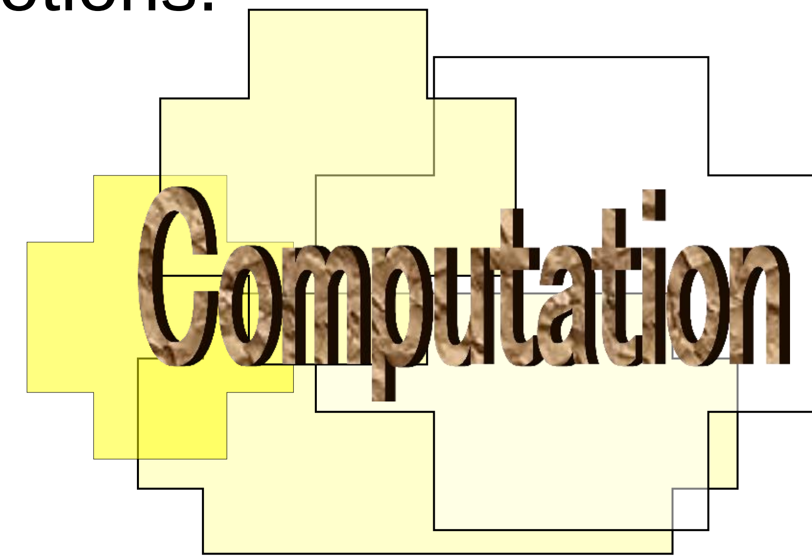
computation

But why ? !

1. Introduction to computation

1.1. Why do we study computation?

Merit #1:
We can get clear/better understanding on somewhat vague notions.



computerize = 計算化

Merit #2:
We can make best use of computational power.

supercomputer

1. Introduction to computation:
1.2. What is computation?

- （1＋4）×5 ＝
- Compute the Largest Common Divisor of 12 and 16
- Factorize $x^2 + 2xy + y^2$

:

- Analyze genome sequence
- Provide web info.

Activities of a brain
Growth of trees
Various chemical processes

What would be basic components of computation ?

1.2. What is computation?

data ＝ target of computation ＝ bin. seq (＝ number)

binary sequence : every data is expressed by bin. seq.

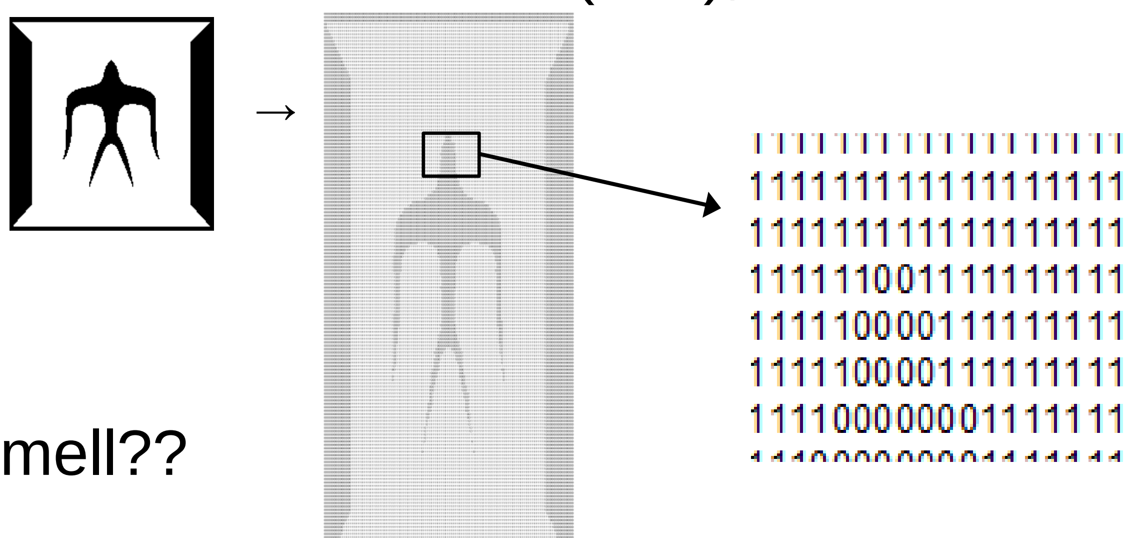
seq. of 0 and 1

Let's see..

- number 18, −5, 3.25, 1/3
- character a ← 01100001 (=97), b ← 01100010 (=98)
- image
- sound
- movie

....

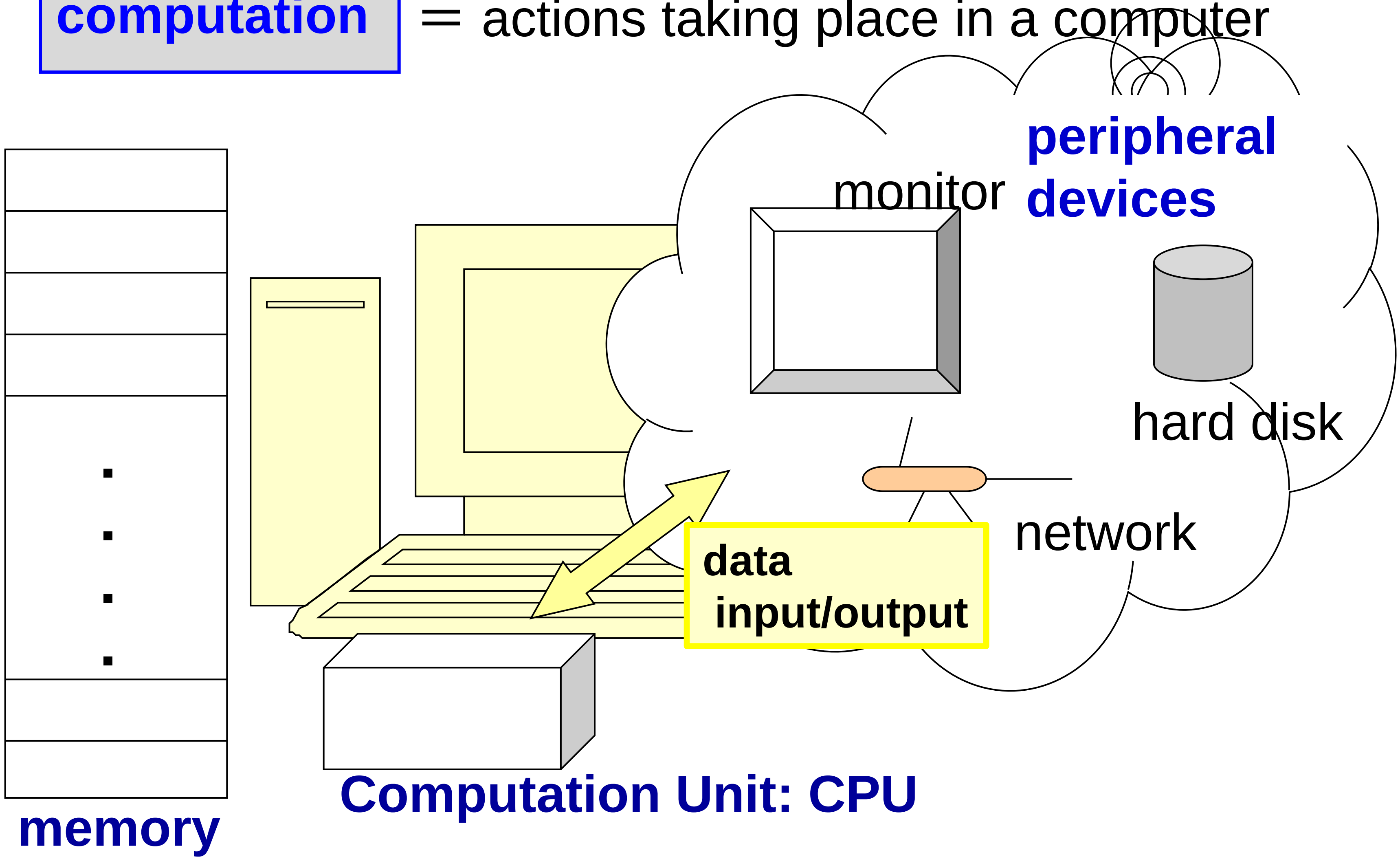
- taste, smell??



1.2. What is computation?

data = target of computation = bin. seq (= number)

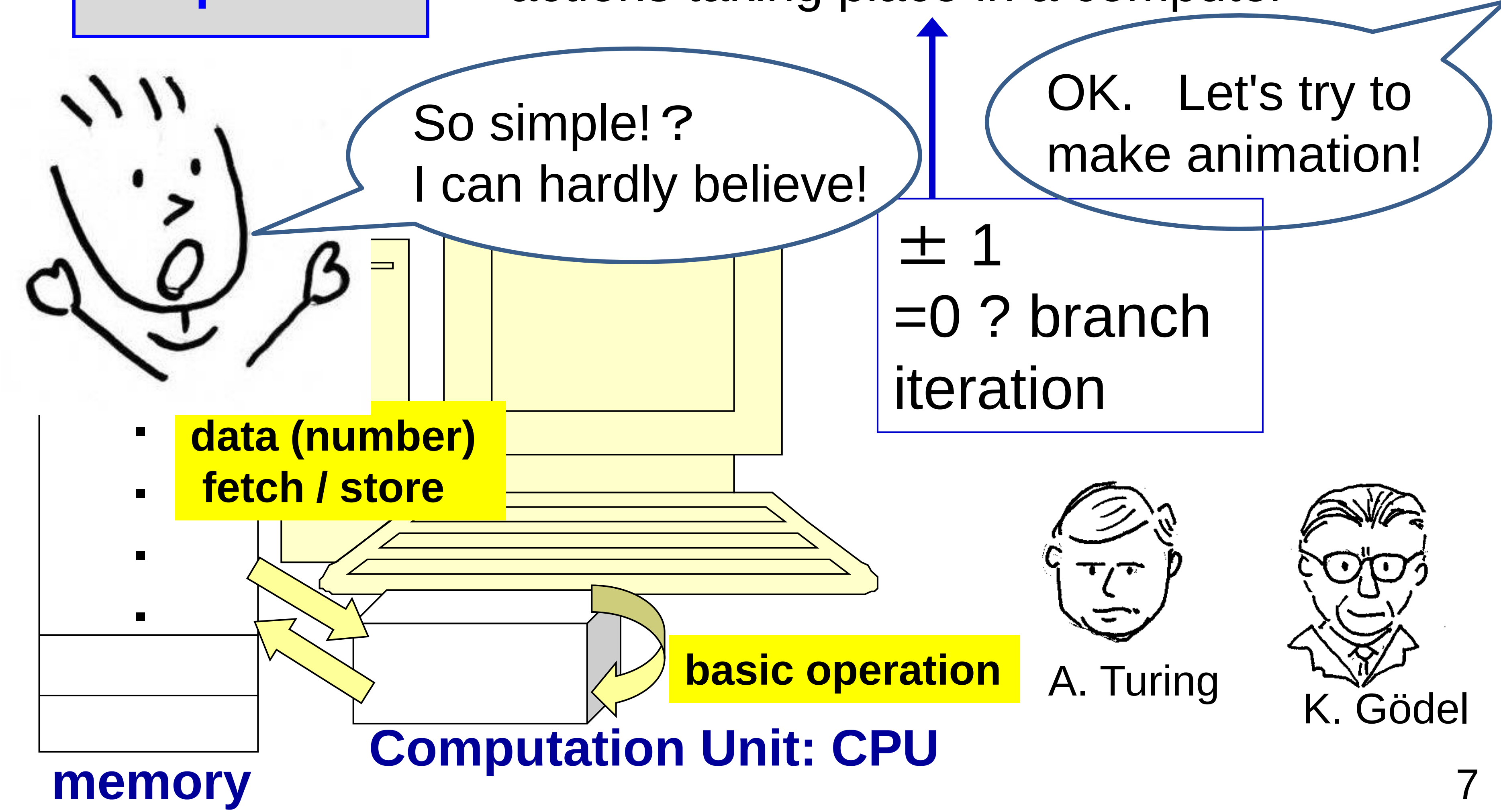
computation = actions taking place in a computer



1.2. What is computation?

data ＝ target of computation ＝ bin. seq (＝ number)

computation ＝ actions taking place in a computer



1.2. What is computation?

animation by using only ±1 & iteration

I can move this sheep.

00000000000000000000000000000000000000000000000000000000000

100000000000000000000000000000000000000000000000000000000000

100000000000000000000000001111111100000000000000000000000000

100000000000000000000000111000000000110000000000000000000000

100000001111111111111001100001100011111100000000000000000000

100000111000000000000000011001110001100001100000000000000000

100000110000000000000000001111100110001000110000000000000000

100000110000000000000000000000000000000001111000000000000000

100000110000000000000000000000000000000111000000000000000000

100000110000000000000000000000000000111000000000000000000000

100000110000000000000000000000000000110000000000000000000000

100000110000000000000000000000000001110000000000000000000000

100000000110001100011111100011000111000000000000000000000000

100000000011100111000000001110011100000000000000000000000000

100000000000000000000000000000000000000000000000000000000000

twenty 60 digit numbers

1.2. What is computation?

data = target of computation = bin. seq

computation = actions taking place inside a computer, ...
but what is their basic components

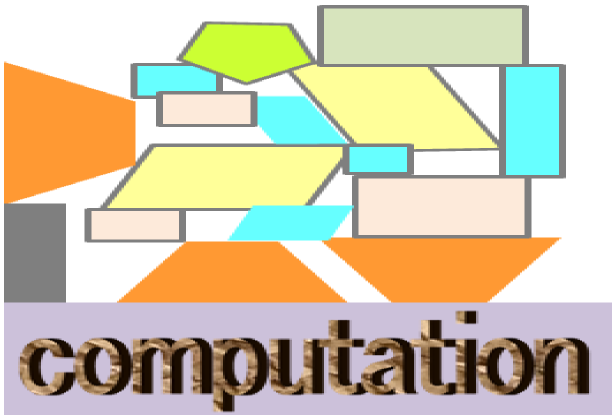
Ans:
± 1
=0 ? branch
iteration

How to specify comp.
for achieving
a target task?

program

How to design comp.
for achieving
a target task?

algorithm
- combinatrics
- theory of comp.

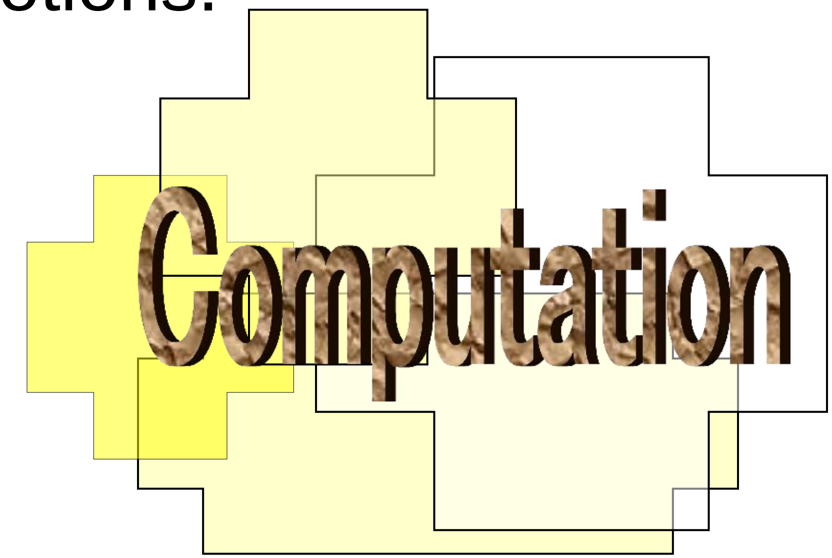


fundamental
programming
techniques

1. Introduction to computation

1.1. Why do we study computation?

Merit #1:
We can get clear/better understanding on somewhat vague notions.

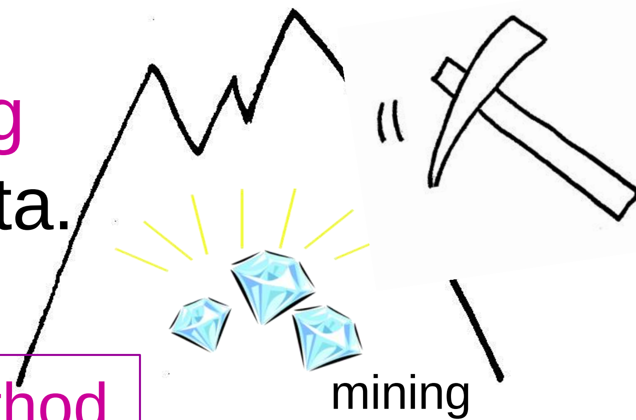


Typical Examples

1. Data Mining
2. Simulation

2. Introduction to data mining
also, machine learning (ML), and AI

Data Mining is to discover something important from a huge amount of data.

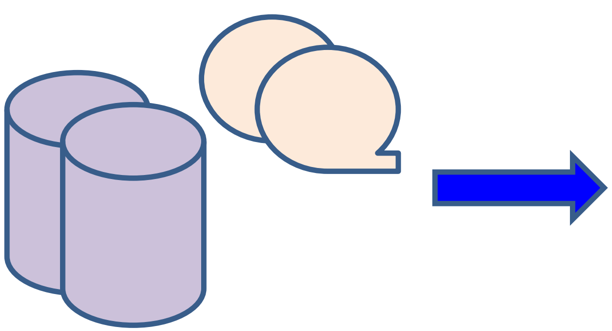


examples

genome analysis ⇒ new health care method
protein analysis ⇒ drug design
business data analysis ⇒ profit / productivity

example

supermarket costumer analysis



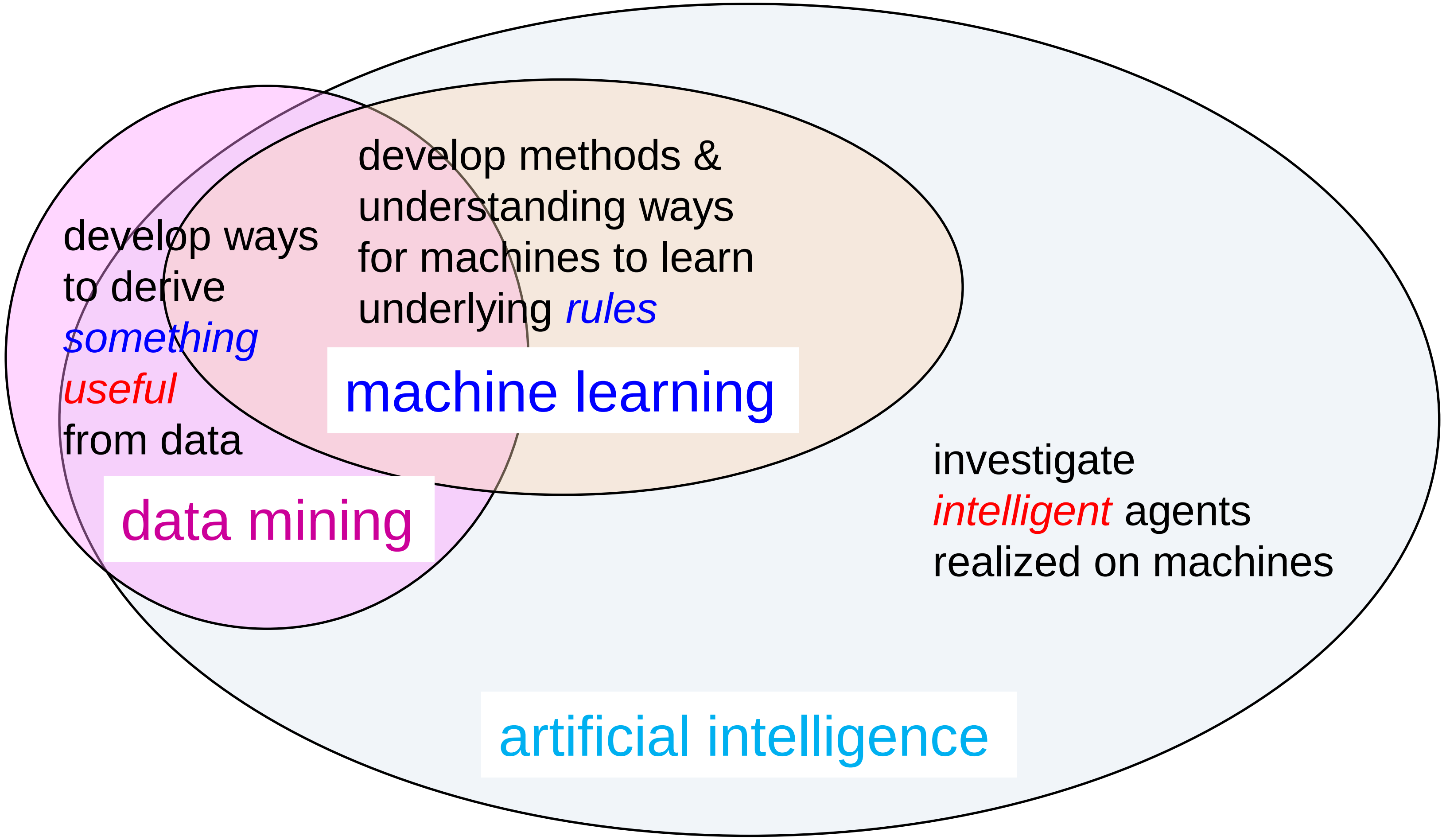
diapers + ?

milk

baby food, ...

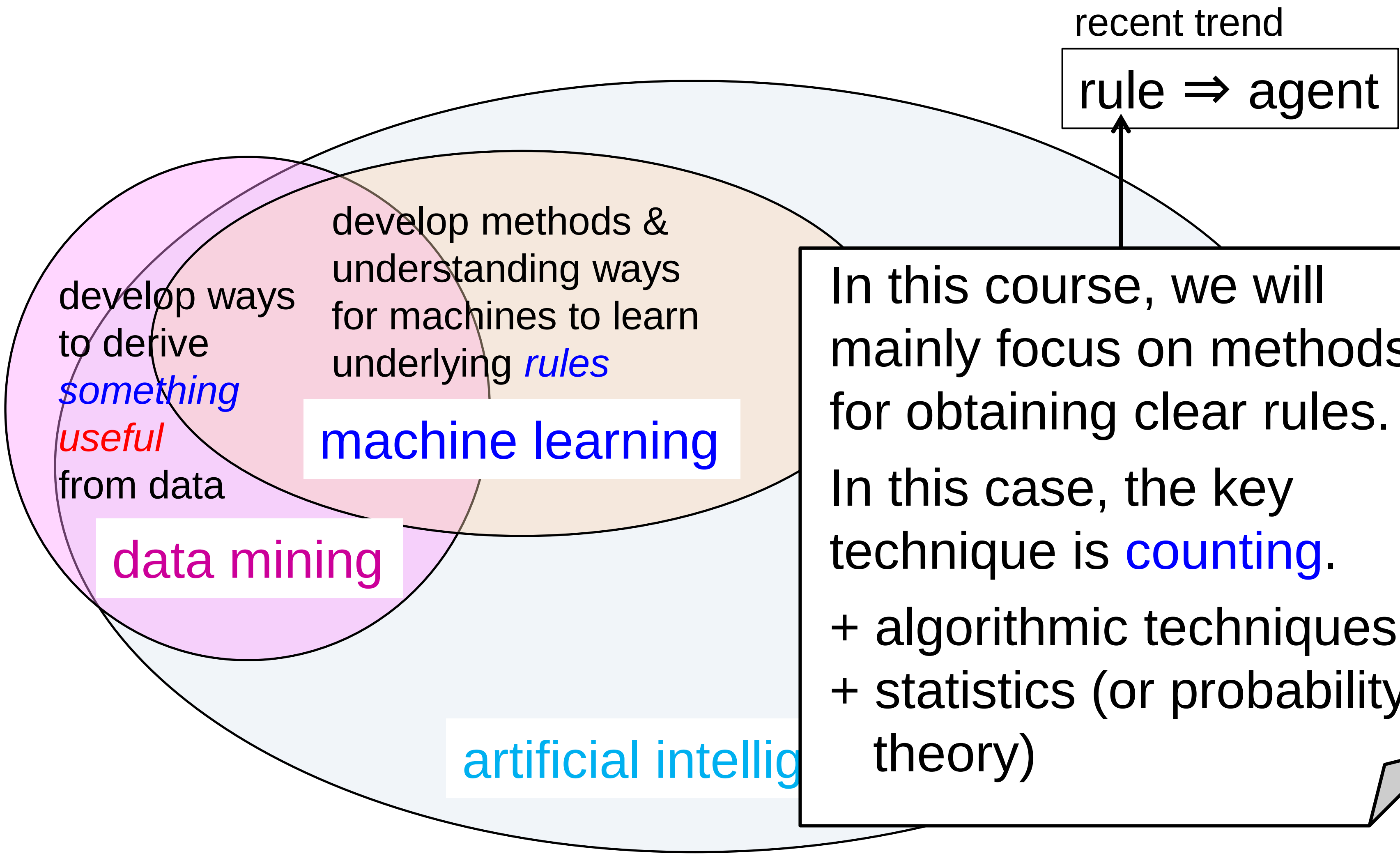
standard staff, what else?

2. Introduction to data mining
2.1. data mining, ML, and AI



Three Areas as Research Fields (My Personal View)

2. Introduction to data mining
2.1. data mining, ML, and AI



In this course, we will mainly focus on methods for obtaining clear rules.

In this case, the key technique is counting.

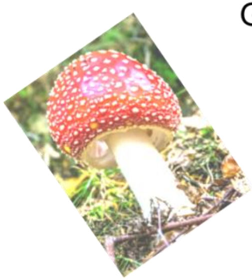
+ algorithmic techniques
+ statistics (or probability theory)

Three Areas as Research Fields (My Personal View)

3. On Exercise #1

As a typical example of data mining

Classification rule discovery of poisonous mushrooms.



Cap surface

Size

Cap shape

About 20 attributes

Smell

General shape

Spots

Cap color

General color

Stem color

Gills

In this course, we will mainly focus on methods for obtaining clear rules.

In this case, the key technique is counting.

+ algorithmic techniques
+ statistics (or probability theory)

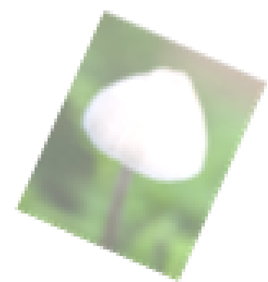
3. On Exercise #1

Now from handout w1ex.pdf for Ex#1.

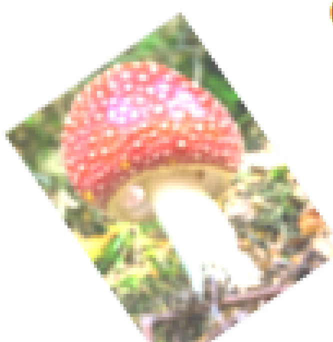
1. Data description and our goal

Discover the rule for classifying poisonous mushrooms

From the mushroom characteristics (attributes), discover a rule (binary decision rule) for determining whether the mushroom is poisonous or not.



Cap surface
Size
Cap shape
About 20 attributes
Smell
Spots
General shape
Cap color
General color
Stem color
Gills



2

From Ex#1 explanation doc.

data or dataset
=
sample = collection of examples (or instances)
=
a tuple of attribute values + class value

class value

- p: poisonous
- e: edible

22 attribute values

cap color

- y: yellow
- p: pink
- b: brown
- …

8000+ mushroom instances

```
p f s n f s f c n b t s s p w p w o e w v d
p k s e f f f c n b t k k w p p w o e w v p
e f f w f n f w b n t f f w w p w o e k s g
e x s w t l f c b n e s s w w p w o p n n g
e x y u f n f c n p e s f w w p w o f h v d
e x y g t n f c b n t s s p g p w o p k y d
e f f e t n f c b u t s s p w p w o p n v d
p f y y f n f w n y e y y y y p y o e w c l
e b s y t l f c b n e s s w w p w o p k s m
p x s w f c f w n g e s s w w p w o p k v d
e s f g f n f c n k e s s w w p w o p k y u
p x s n f y f c n b t k k p p p w o e w v d
e x f g t n f c b p t s s w g p w o p k v d
:
```

Use only these data!

UC Irvine ML Repository, 1987

From Ex#1 explanation doc.

goal = to find a *good* rule for detecting the class value of a given mushroom instance

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
p	f	s	n	f	s	f	c	n	b	t	s	s	p	w	p	w	o	e	w	v	d
p	k	s	e	f	f	f	c	n	b	t	k	k	w	p	p	w	o	e	w	v	p
e	f	f	w	f	n	f	w	b	n	t	f	f	w	w	p	w	o	e	k	s	g
e	x	s	w	t	l	f	c	b	n	e	s	s	w	w	p	w	o	p	n	n	g
e	x	y	u	f	n	f	c	n	p	e	s	f	w	w	p	w	o	f	h	v	d
e	x	y	g	t	n	f	c	b	n	t	s	s	p	g	p	w	o	p	k	y	d
e	f	f	e	t	n	f	c	b	u	t	s	s	p	w	p	w	o	p	n	v	d
p	f	y	y	f	n	f	w	n	y	e	y	y	y	y	p	y	o	e	w	c	l
e	b	s	y	t	l	f	c	b	n	e	s	s	w	w	p	w	o	p	k	s	m
p	x	s	w	f	c	f	w	n	g	e	s	s	w	w	p	w	o	p	k	v	d
e	s	f	g	f	n	f	c	n	k	e	s	s	w	w	p	w	o	p	k	y	u
p	x	s	n	f	y	f	c	n	b	t	k	k	p	p	p	w	o	e	w	v	d
e	x	f	g	t	n	f	c	b	p	t	s	s	w	g	p	w	o	p	k	v	d

⋮

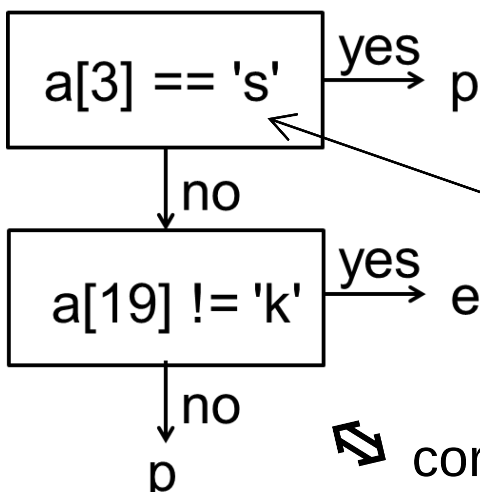
UC Irvine ML Repository, 1987

rule = a Boolean expression that determines a given mushroom's toxicity

more specifically

rule = a decision list

e.g.,



each base predicate is to ask whether a[*k*] == *val* or not

use some appropriate # of base predicates

⇔ corresponding Boolean expression

a[3] == 's' ∨ (a[3] != 's' ∧ a[19] == 'k')

From Ex#1 explanation doc.

Good rule = a decision list with low error rate

$$\text{Accuracy (or, success rate)} = \frac{\text{Number of correctly classified instances}}{\text{Total number of instances}}$$

$$\text{Error rate} = \frac{\text{Number of } \textit{in}\text{correctly classified instances}}{\text{Total number of instances}}$$

False positive = Incorrectly classified as positive

False negative = Incorrectly classified as negative

in our mushroom data, let us call

p: poisonous = +1, positive
e: edible = -1, negative

Better to avoid false negative!

for our mushroom classification task

Homework assignment #1: Task

training set

Your task:

(a) Obtain a decision list using 2000 instances of the mushroom data (m8124org.txt) with accuracy > 90% on the whole dataset.

page 7

demo. in the ex. session

* Use only the provided python programs.
* You may modify these programs *as you like!!*

(b) Understand the mechanism of the provided python programs.

Homework assignment #1: Report

From Ex#1 explanation doc.

submit through OCW *before* week2lect

Required items that you need to explain:

Japanese is OK!!
About 1 page for each item, please!

(1) + a decision list that you obtained,
+ its corresponding Boolean expression (used in test.py), and
+ its statistical data, that is,
+ accuracy, true positive rate, true negative rate
both on the training set and on the whole data set.

$$\text{Ture positive rate} = \frac{\text{Number of correctly classified positive instances}}{\text{Total number of positive instances}}$$

$$\text{Ture negative rate} = \frac{\text{Number of correctly detected negative instances}}{\text{Total number of negative instances}}$$

Homework assignment #1: Report

From Ex#1 explanation doc.

Required items that you need to explain: (Cont.)

(2) Explain a way to obtain your decision list at.
  * The outline of what you did (or what your program did) for obtaining your decision list.

(3) Explain a key program (e.g., count.py) that you used.
  + explanation of the program outline, and
  + the source code of the program with explanation on what is computed at each key statement.
  ↑ hand written comments are enough!!

3. On Exercise #1

Additional explanation for those who have no experience on reading/writing programs.

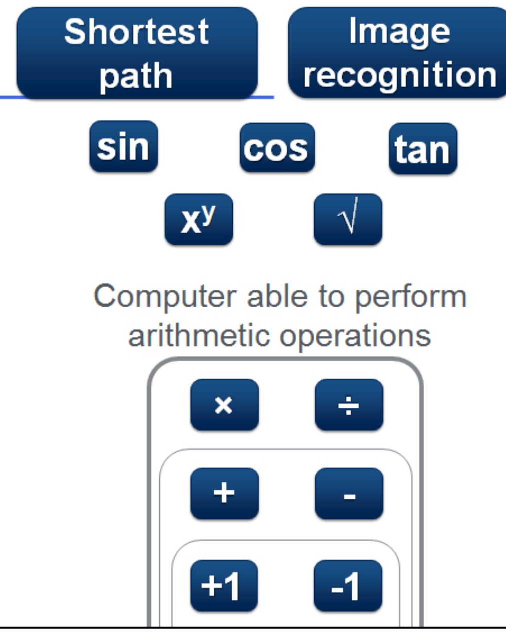
Warning

The following explanation materials are partially from CS2 (100 undergrad. course), in which we use Ruby for programming language. So some examples are Ruby programs.

In the following, we explain some very basic points on programming for understanding those used for the Ex#1, e.g., test.py, count.py, etc.

Summary of Previous Lecture:

- **Data**
  - Data = number
  - Basic elements: 0 and 1
- **Computation**
  - Basic operations: **±1** and **loop**



Though theoretically correct, it is too much to write a program by using only these basic operations.

Various important programming techniques have been invented.

Program

- **Assignment** (variables)
- **Loop statement** (while statement)
- **Conditional branching** (if statement)

```
a = 8
b = 3
wa = a
wa = wa + 1
b = b - 1
```

```
if x >= 0
  puts( x )
else
  puts( x*(-1) )
end
```

```
while b > 0
  wa = wa + 1
  b = b - 1
end
```

One of such important techniques:

Array A variable that can store multiple data values.

E.g.,

	0	1	2	3	4	5
a	2	4	0	11	8	12

referred as a[3]

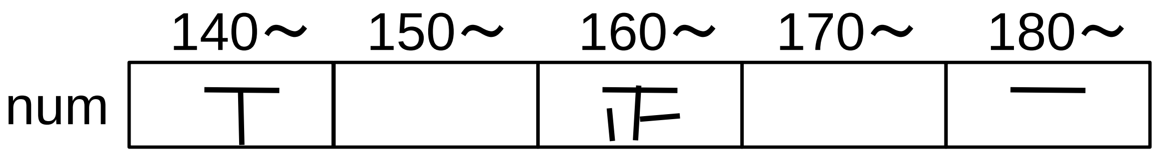
computation of summation

```
a = [2, 4, 6, 8, 10, 12 ]
s = 0
k = 0
while k < 6:
   s = s + a[ k ]
   k = k + 1
print(s)
```

$$s = \sum_{k=0}^{5} a_k$$

```
a = [2, 4, 6, 8, 10, 12 ]
s = 0
for k in range(6):
   s = s + a[ k ]
print(s)
```

Array can be used for counting!




```
num = [0, 0, 0, 0, 0 ]

n = 35

for  k  in rage(n):
   d = int( input() )
   p = int( (d - 140)/10 )
   num[p] = num[p] + 1

print(num)
```

determine the index to increment