CS420

Introduction to the Theory of Computation

Lecture 18: Countable and uncountable sets

Tiago Cogumbreiro

Today we will learn...

- Emptiness tests
- Equality tests
- Hilbert Hotel
- Countable and uncountable sets

> Section 4.1, 4.2
> Supplementary material: "Hospitality at the Hilbert Hotel" by Ana Pires

E_X: Emptiness tests

Decidable algorithms on emptiness

(Is the language of X empty?)

E_{DFA}: DFA Emptiness

The set of DFAs that recognize an empty language.

$$E_{\mathsf{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$$

Theorem 4.4. E_{DFA} is a decidable language.

Proof.

E_{DFA}: DFA Emptiness

The set of DFAs that recognize an empty language.

$$E_{\mathsf{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$$

Theorem 4.4. E_{DFA} is a decidable language.

Proof. (Note: we do not argue the correctness, although technically we should.)

1. Mark the initial state of DFA A as to-visit and visited.
2. While there are states to visit: Unmark one to-visit and mark all transitions that have not been visited as to-visit and as visited
3. Accept when zero visited states are accepted, otherwise reject.

Totality argument: The loop terminates because at each step the set of potential states to visit is bounded by the total number of states and that number decreases by at least one at each iteration step.

E_{DFA}: DFA Emptiness

Python implementation

```
def is_empty(dfa):
  to_visit = [dfa.start_state]
  visited = set(to_visit)
  while len(to_visit) > 0:
    node = to_visit.pop()
    for i in dfa.alphabet:
      st = dfa.state_transition(st,i)
      if st not in visited:
        to_visit.append(st)
        visited.add(st)
  for st in visited:
    if st in dfa.accepted_states:
      return True
  return False
```

E_{CFG}: CFG Emptiness

The set of CFGs that recognize an empty language.

$$E_{\mathsf{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}$$

Theorem 4.8. E_{CFG} is a decidable language.

Proof.

$E_{\textsf{CFG}}$: CFG Emptiness

The set of CFGs that recognize an empty language.

$$E_{\textsf{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}$$

Theorem 4.8. $E_{\textsf{CFG}}$ is a decidable language.

Proof.

1. Mark all terminal symbols of G
2. Until no new variables get marked
   - Mark any variable where G has a rule $G \to A_1 \ldots A_n$ and each A_i has been marked.
3. If the start variable is marked reject, otherwise accept.

Totality argument: The set of unmarked variables is bounded by the set of all variables and terminals. At each iteration the set of unmarked variables increases until it terminates.

EQ_X: Equality tests

Decidable algorithms on equality

(Can we **always** test if two elements of X are equal?)

EQ_{DFA}: DFA Equality

> The set of DFAs that are equal to one another.

$$EQ_{\mathsf{DFA}} = \{\langle A, B\rangle \mid A, B \text{ are DFAs and } L(A) = L(B)\}$$

Theorem 4.5. EQ_{DFA} is a decidable language.

Proof.

Theorem 4.5. EQ_{DFA} is a decidable language.

Proof.

Let the simmetric difference be defined as:

$$A \triangle B = (A - B) \cup (B - A)$$

1. It is easy to see that $L(A) = L(B)$ if, and only if, $L(A) \triangle L(B) = \emptyset$.
2. $A \triangle B$ is closed under the set of regular languages[†]
3. The algorithm then becomes testing the emptiness of the automaton $A \triangle B$, which we know to be decidable.[⋆]

[†]: The set difference and the union are closed under the set of regular languages.

[⋆]: The automaton $A \triangle B$ can be trivially defined as automaton-operations.

EQ_{CFG}: CFG Equality

Let us think about

$$\infty$$

David Hilbert on infinite sets

David Hilbert

- David Hilbert is one of the most influential mathematicians of the 19th/20th centuries

The difference between finite and infinite sets

- In a **full** hotel with a **finite** number of rooms, there is no room to accommodate a new guest arrives.
- In a **full** hotel with **infinite** rooms, there is always room to accommodate new guests!

Hilber Hotel

The Hilber Hotel

- A room with infinite rooms
- Every room is occupied by one person
- Every person is uniquely identified by a number
- Person i is in room number i

Room 1	**Room 2**	**Room 3**	**...**	**Room i**	**...**
Person 1	Person 2	Person 3	...	Person i	...

Adding a new guest

> One new guest arrives. How can we accommodate them?

Tips

- We can broadcast messages to every occupant
- We can ask occupants to move

Room 1	***Room 2***	***Room 3***	***Room 4***	***Room 5***	**...**
Person 1	Person 2	Person 3	Person 4	Person 5	...

Adding a new guest

One new guest arrives. How can we accommodate them?

Tips

- We can broadcast messages to every occupant
- We can ask occupants to move

Room 1	***Room 2***	***Room 3***	***Room 4***	***Room 5***	***...***
Person 1	Person 2	Person 3	Person 4	Person 5	...

Solution

- We can ask every occupant to move one room to the right: $r(n) = n + 1$
- New guest goes to room 0

Do we have more guests than before?

Do we have more guests than before?

No. We still an infinite number of guests.

Adding k guests

A bus with k guests arrives. How can we accommodate them?

Room 1	Room 2	Room 3	Room 4	Room 5	...
Person 1	Person 2	Person 3	Person 4	Person 5	...

Adding k guests

A bus with k guests arrives. How can we accommodate them?

Room 1	Room 2	Room 3	Room 4	Room 5	...
Person 1	Person 2	Person 3	Person 4	Person 5	...

- We can ask every occupant to move k rooms to the right:
  $r(n) = n + k$
- New guests go to room $0, \ldots, k$

Adding infinite guests

- A bus with ∞ guests arrives. How can we accommodate them?

Adding infinite guests

A bus with ∞ guests arrives. How can we accommodate them?

Solution

- We can ask every occupant to move to even room numbers:
  $r(n) = 2 \times n$
- New guests go to odd room numbers:
  $r(n) = 2 \times n + 1$

Adding k busses full of infinite guests

> k bus arrive, each filled with ∞ guests. How can we accommodate them?

Adding k busses full of infinite guests

k bus arrive, each filled with ∞ guests. How can we accommodate them?

Solution

- Guests in the hotel go to room $r_{hotel}(n) = (k + 1) \times n$
- Bus $r_{bus}(b, n) = (k + 1) \times n + b$

Example

Let $k = 1$ (one bus), then $r_{hotel}(n) = 2 \times n$ and $r_{bus}(b, 1) = 2 \times n + 1$, as before.

Adding infinite busses with infinite guests

- ∞ busses with ∞ guests arrive. How can we accommodate them?

Adding infinite busses with infinite guests

> ∞ busses with ∞ guests arrive. How can we accommodate them?

Idea: arrange them in a matrix!

$$
\begin{array}{cccccc}
\text{hotel} & g_{11} & g_{12} & g_{13} & g_{14} & \cdots \\
b_1 & g_{21} & g_{22} & g_{23} & g_{24} & \cdots \\
b_2 & g_{31} & g_{32} & g_{33} & g_{34} & \cdots \\
b_3 & g_{41} & g_{42} & g_{43} & g_{44} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
$$

Adding infinite busses with infinite guests

> ∞ busses with ∞ guests arrive. How can we accommodate them?

Use diagonals to order the elements

$$\begin{array}{cccccc}
\text{hotel} & g_{11} & g_{12} & g_{13} & g_{14} & \cdots \\
b_1 & g_{21} & g_{22} & g_{23} & g_{24} & \cdots \\
b_2 & g_{31} & g_{32} & g_{33} & g_{34} & \cdots \\
b_3 & g_{41} & g_{42} & g_{43} & g_{44} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

Adding infinite busses with infinite guests

Now every element has a unique number

> ∞ busses with ∞ guests arrive. How can we accommodate them?

$$\begin{array}{cccccc}
\text{hotel} & r_1 & r_3 & r_6 & r_{10} & \cdots \\
b_1 & r_2 & r_5 & r_9 & r_{14} & \cdots \\
b_2 & r_4 & r_8 & r_{13} & r_{19} & \cdots \\
b_3 & r_7 & r_{12} & r_{18} & r_{25} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

Adding infinite busses with infinite guests

Now every element has a unique number

> ∞ busses with ∞ guests arrive. How can we accommodate them?

$$
\begin{array}{cccccc}
\text{hotel} & r_1 & r_3 & r_6 & r_{10} & \cdots \\
b_1 & r_2 & r_5 & r_9 & r_{14} & \cdots \\
b_2 & r_4 & r_8 & r_{13} & r_{19} & \cdots \\
b_3 & r_7 & r_{12} & r_{18} & r_{25} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
$$

Solution

$$r(i,j) = \sum (i + j - 2) + j$$

Recap

Recap: injective function

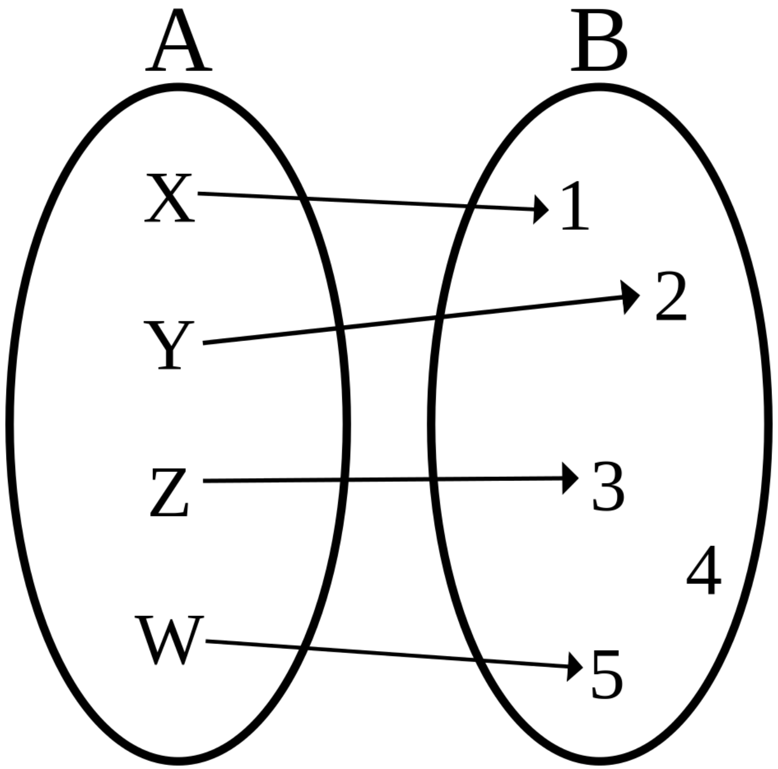
One-to-one relation

$f : A \to B$

A is the domain

B is the domain

- **Injective** functions $x \neq y$ implies $f(x) \neq f(y)$



Recap: surjective function

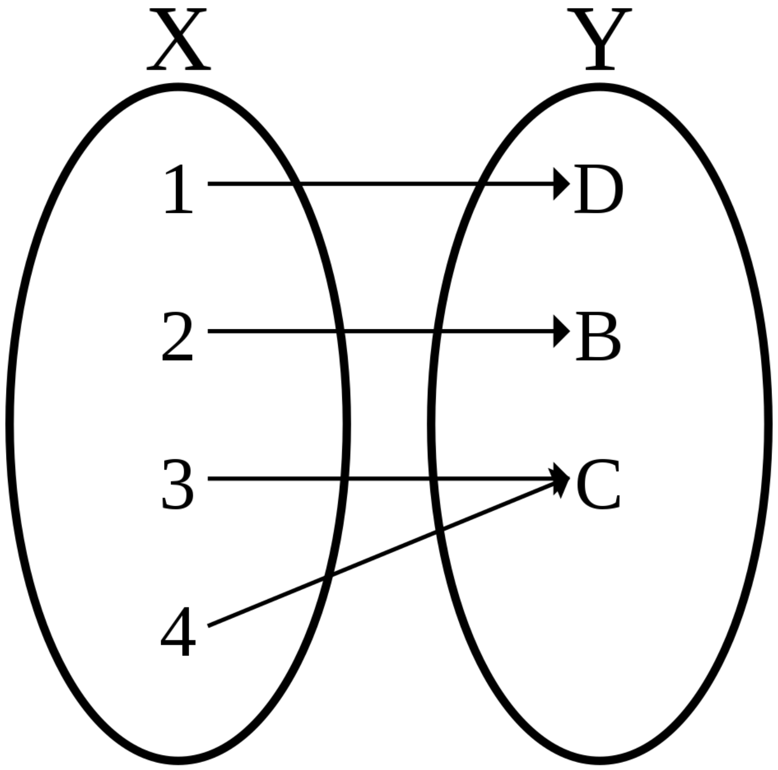
Onto relation

$f : A \to B$

A is the domain

B is the domain

- **Surjective** all elements in the ***domain*** $b \in B$, then $\exists a, f(a) = b$



Recap: bijective function

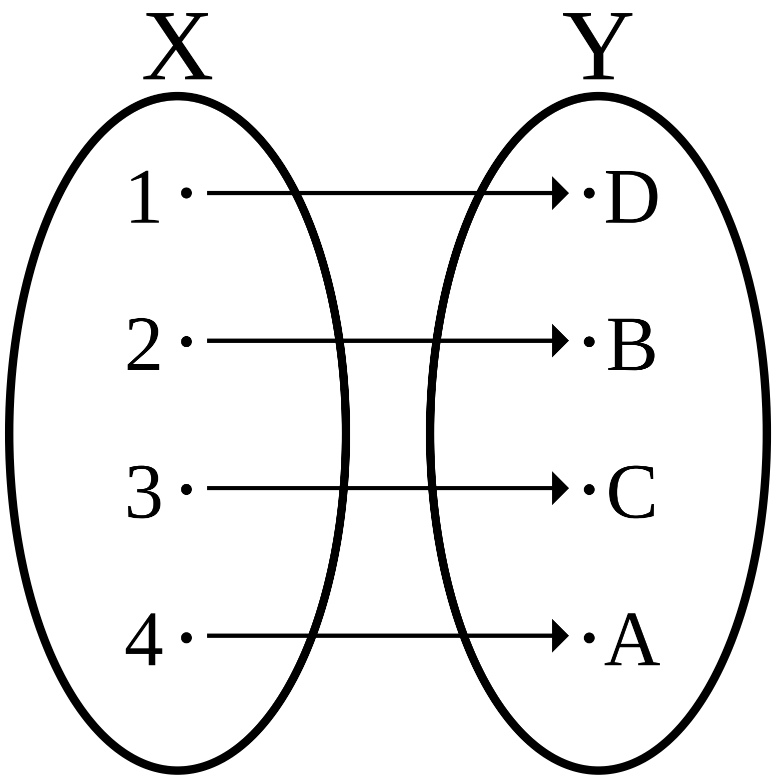
Correspondence relation

$f : A \to B$

A is the domain

B is the domain

- **Bijective** injective and surjective



Set size

- Two sets have the **same size** if, and only if, there exists a **bijection** between them.

Set size

- Two sets have the **same size** if, and only if, there exists a **bijection** between them.

Applicable to infinite sets!

Countable sets

Infinite sets

Definition 4.14: Countable

- Any finite set is **countable**.
- An infinite set is **countable** if, and only if, it has the same size of $\mathcal{N}$.

$$\mathcal{N} = \{1, 2, 3, \dots\}$$

Odd numbers

$$O = \{1, 3, 5, 7, \dots\}$$

We know that $O \subset \mathcal{N}$

Any even number in $\mathcal{N}$ is not in O.

Is the size of the odd numbers smaller than $\mathcal{N}$?

Odd numbers

$$O = \{1, 3, 5, 7, \dots\}$$

We know that $O \subset \mathcal{N}$

Any even number in $\mathcal{N}$ is not in O.

Is the size of the odd numbers smaller than $\mathcal{N}$?

No. The size of odd numbers is the **same** as $\mathcal{N}$. The odd numbers are **countable**!

The bijection is: $o(n) = 2 \times n - 1$

O	$\mathcal{N}$
1	1
3	2
5	3
7	4
...	...

Positive rational numbers

$$\mathcal{Q}^+ = \{n \div m \mid n, m \in \mathcal{N}\}$$

We know that $\mathcal{N} \subset \mathcal{Q}^+$

Any fraction is not in $\mathcal{N}$.

Is the size of $\mathcal{Q}^+$ larger than $\mathcal{N}$?

Positive rational numbers

$$\mathcal{Q}^+ = \{n \div m \mid n, m \in \mathcal{N}\}$$

We know that $\mathcal{N} \subset \mathcal{Q}^+$

Any fraction is not in $\mathcal{N}$.

Is the size of $\mathcal{Q}^+$ larger than $\mathcal{N}$?

They have the **same** size!

Proving that Q^+ is countable