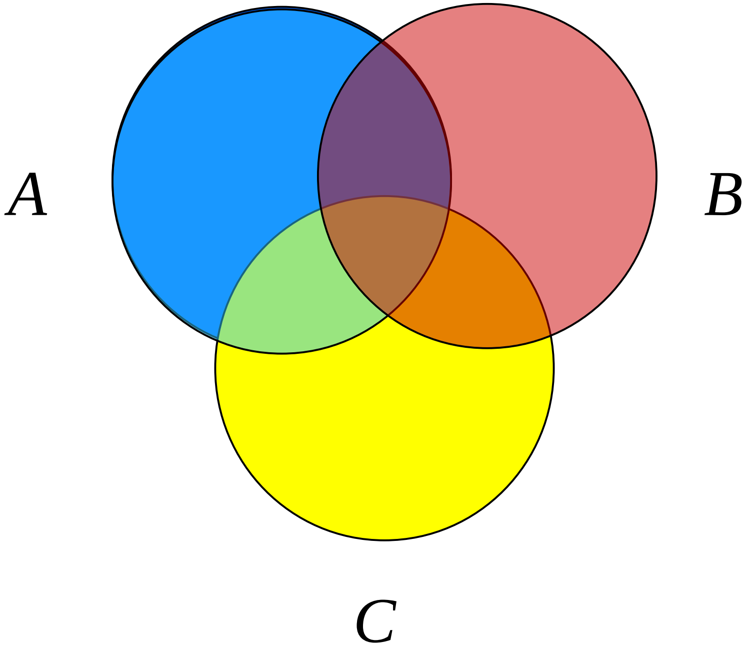
Set Theory



This Lecture

We will first introduce set theory before we do counting.

- Basic Definitions
- Operations on Sets
- Set Identities
- Russell's Paradox

Defining Sets

Definition: A **set** is an unordered collection of objects.

The objects in a set are called the **elements** or **members** of the set S, and we say S **contains** its elements.

We can define a set by directly listing all its elements.

e.g. S = {2, 3, 5, 7, 11, 13, 17, 19},

S = {CSC1130, CSC2110, ERG2020, MAT2510}

After we define a set, the set is a single mathematical object, and it can be an element of another set.

e.g. S = {{1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}}

Defining Sets by Properties

It is inconvenient, and sometimes impossible, to define a set by listing all its elements.

Alternatively, we can define by a set by describing the *properties* that its elements should satisfy.

We use the notation $\{x \in A \mid P(x)\}$

to define the set as the set of elements, *x*, in *A* such that *x* satisfies property *P*.

e.g. $\{x \mid x \text{ is a prime number and } x < 1000\}$

$\{x \mid x \text{ is a real number and } -2 < x < 5\}$

Examples of Sets

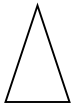
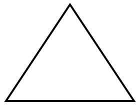
Well known sets:

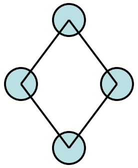
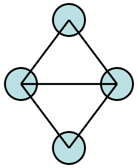
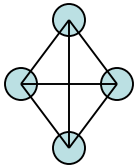
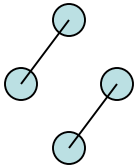
- the set of all real numbers, $\mathbb{R}$
- the set of all complex numbers, $\mathbb{C}$
- the set of all integers, $\mathbb{Z}$
- the set of all positive integers $\mathbb{Z}^+$
- **empty set**, $\emptyset = \{\}$, the set with no elements.

Other examples:

The set of all polynomials with degree at most three: $\{1, x, x^2, x^3, 2x+3x^2, \ldots\}$.

The set of all n-bit strings: {000…0, 000…1, …, 111…1}

The set of all triangles without an obtuse angle: {  ,  ,… }

The set of all graphs with four nodes: {  ,  ,  ,  ,…}

Membership

Order, number of occurence are not important.

e.g. {a,b,c} = {c,b,a} = {a,a,b,c,b}

The most basic question in set theory is whether an element is in a set.

$x \in A$ x is an element of A

x is in A

$x \notin A$ x is not an element of A

x is not in A

e.g. Recall that Z is the set of all integers. So $7 \in \mathrm{Z}$ and $2/3 \notin \mathrm{Z}$.

Let P be the set of all prime numbers. Then $97 \in P$ and $321 \notin P$

Let Q be the set of all rational numbers. Then $0.5 \in Q$ and $\sqrt{2} \notin Q$

↗

(will prove later)

Size of a Set

In this course we mostly focus on finite sets.

Definition: The **size** of a set S, denoted by **|S|**, is defined as the number of elements contained in S.

e.g. if S = {2, 3, 5, 7, 11, 13, 17, 19}, then |S|=8.

if S = {CSC1130, CSC2110, ERG2020, MAT2510}, then |S|=4.

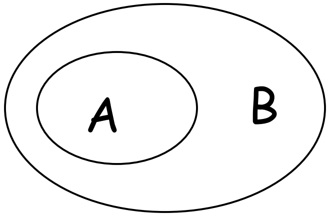
if S = {{1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}}, then |S|=6.

Later we will study how to determine the size of the following sets:

- the set of poker hands which are “full house”.
- the set of n-bit strings without three consecutive ones.
- the set of valid ways to add n pairs of parentheses

Subset

Definition: Given two sets A and B, we say A is a **subset** of B, denoted by $A \subseteq B$, if every element of A is also an element of B.



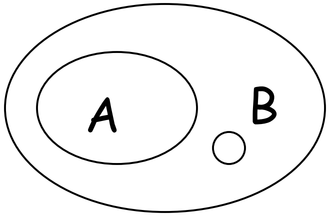
not a subset

- If A={4, 8, 12, 16} and B={2, 4, 6, 8, 10, 12, 14, 16}, then $A \subseteq B$ but $B \not\subseteq A$
- $A \subseteq A$ because every element in A is an element of A.
- $\emptyset \subseteq A$ for any A because the empty set has no elements.
- If A is the set of prime numbers and B is the set of odd numbers, then $A \not\subseteq B$

Fact: If $A \subseteq B$, then |A| <= |B|.

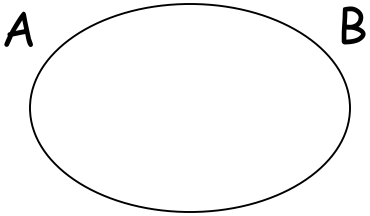
Proper Subset, Equality

Definition: Given two sets A and B, we say A is a **proper subset** of B, denoted by $A \subset B$, if every element of A is an element of B, But there is an element in B that is not contained in A.



Fact: If $A \subset B$, then |A| < |B|.

Definition: Given two sets A and B, we say A = B if $A \subseteq B$ and $B \subseteq A$.



Fact: If A = B, then |A| = |B|.

Exercises

1. $\mathbb{Z} \subset \mathbb{R}$?
2. $\{3\} \in \{5,7,3\}$?
3. $\varnothing \in$ every set?
4. $\{1,2\} \subseteq \{\{1,2\}, \{2,3\}, \{3,1\}\}$?
5. $\{a\} = \{\{a\}\}$?
6. If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$?
7. If $A \subset B$ and $B \subseteq C$, then $A \subset C$?

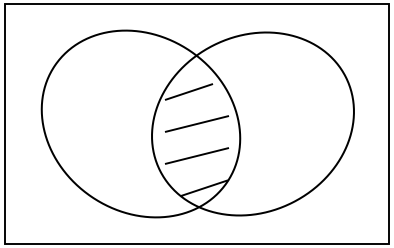
This Lecture

- Basic Definitions
- Operations on Sets
- Set Identities
- Russell's Paradox

Basic Operations on Sets

Let A, B be two subsets of a *universal* set U
(depending on the context U could be R, Z, or other sets).

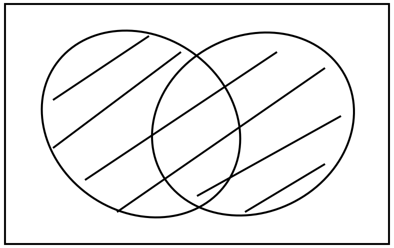
intersection: $A \cap B = \{x \in U \mid x \in A \text{ and } x \in B\}$



Defintion: Two sets are said to be **disjoint** if their intersection is an empty set.

e.g. Let A be the set of odd numbers, and B be the set of even numbers.
Then A and B are disjoint.

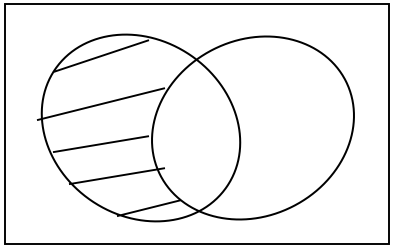
union: $A \cup B = \{x \in U \mid x \in A \text{ or } x \in B\}$



Fact: $|A \cup B| = |A| + |B| - |A \cap B|$

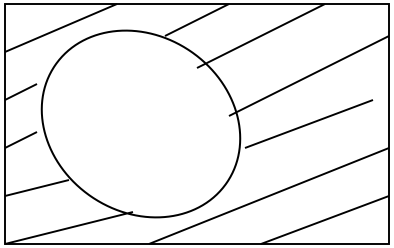
Basic Operations on Sets

difference: $A - B = \{x \in U \mid x \in A \text{ and } x \notin B\}$



Fact: $|A - B| = |A| - |A \cap B|$

complement: $\overline{A} = A^c = \{x \in U \mid x \notin A\}$



e.g. Let U = Z and A be the set of odd numbers.

Then $\overline{A}$ is the set of even numbers.

Fact: If $A \subseteq B$, then $\overline{B} \subseteq \overline{A}$

Examples

$A = \{1, 3, 6, 8, 10\}$ $B = \{2, 4, 6, 7, 10\}$

$A \cap B = \{6, 10\}$, $A \cup B = \{1, 2, 3, 4, 6, 7, 8, 10\}$ $A - B = \{1, 3, 8\}$

Let $U = \{ x \in Z \mid 1 <= x <= 100\}$.

$A = \{ x \in U \mid x$ is divisible by 3$\}$, $B = \{ x \in U \mid x$ is divisible by 5$\}$

$A \cap B = \{ x \in U \mid x$ is divisible by 15$\}$

$A \cup B = \{ x \in U \mid x$ is divisible by 3 or is divisible by 5 (or both)$\}$

$A - B = \{ x \in U \mid x$ is divisible by 3 but is not divisible by 5 $\}$

Exercise: compute $|A|$, $|B|$, $|A \cap B|$, $|A \cup B|$, $|A - B|$.

Partitions of Sets

Two sets are disjoint if their intersection is empty.

A collection of nonempty sets $\{A_1, A_2, \ldots, A_n\}$ is a partition of a set A if and only if

$$A = A_1 \cup A_2 \cup \cdots \cup A_n$$

$A_1, A_2, \ldots, A_n$ are mutually disjoint (or pairwise disjoint).

e.g. Let A be the set of integers.

Let A_1 be the set of negative integers.

Let A_2 be the set of positive integers.

Then $\{A_1, A_2\}$ is not a partition of A, because $A \neq A_1 \cup A_2$

as 0 is contained in A but not contained in $A_1 \cup A_2$

Partitions of Sets

e.g. Let A be the set of integers divisible by 6.

A_1 be the set of integers divisible by 2.

A_2 be the set of integers divisible by 3.

Then $\{A_1, A_2\}$ is not a partition of A, because A_1 and A_2 are not disjoint, and also $A \subset A_1 \cup A_2$ (so both conditions are not satisfied).

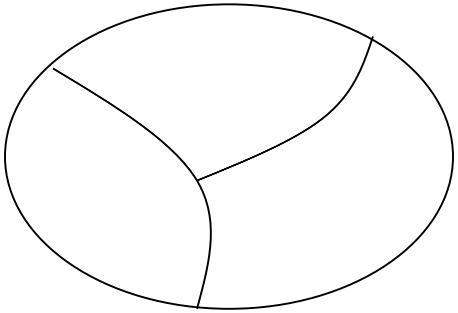
e.g. Let A be the set of integers.

$A_1 = \{x \in A \mid x = 3k+1 \text{ for some integer } k\}$

$A_2 = \{x \in A \mid x = 3k+2 \text{ for some integer } k\}$

$A_3 = \{x \in A \mid x = 3k \text{ for some integer } k\}$

Then $\{A_1, A_2, A_3\}$ is a partition of A



Power Sets

power set: $$\mathrm{pow}(A) ::= \{S \mid S \subseteq A\}$$

In words, the power set pow(*A*) of a set *A* contains all the subsets of *A* as members.

pow({a,b}) = {∅, {a}, {b}, {a,b}}

pow({a,b,c}) = {∅, {a}, {b}, {c}, {a,b}, {a,c}, {b,c}, {a,b,c}}

pow({a,b,c,d}) = {∅, {a}, {b}, {c}, {d},
{a,b}, {a,c}, {b,c}, {a,d}, {b,d}, {c,d},
{a,b,c}, {a,b,d}, {a,c,d}, {b,c,d}, {a,b,c,d}}

Fact (to be explained later): If *A* has n elements, then pow(*A*) has 2^n elements.

Cartesian Products

Definition: Given two sets A and B, the **Cartesian product** A x B is the set of all **ordered** pairs (a,b), where a is in A and b is in B. Formally,

$$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$$

Ordered pairs means the ordering is important, e.g. (1,2) ≠ (2,1)

e.g. Let A be the set of letters, i.e. {a,b,c,…,x,y,z}.

Let B be the set of digits, i.e. {0,1,…,9}.

AxA is just the set of strings with two letters.

BxB is just the set of strings with two digits.

AxB is the set of strings where the first character is a letter and the second character is a digit.

Cartesian Products

The definition can be generalized to any number of sets, e.g.

$$A \times B \times C = \{(a,b,c) \mid a \in A \text{ and } b \in B \text{ and } c \in C\}$$

Using the above examples, AxAxA is the set of strings with three letters.

An ID card number has one letter and then six digits,
so the set of ID card numbers is the set AxBxBxBxBxBxB.

Fact: If |A| = n and |B| = m, then |AxB| = mn.

Fact: If |A| = n and |B| = m and |C| = l, then |AxBxC| = mnl.

Fact: $|A_1 x A_2 x \ldots x A_k| = |A_1| x |A_2| x \ldots x |A_k|$.

Exercises

1. Let A be the set of prime numbers, and let B be the set of even numbers. What is $A \cap B$ and $|A \cap B|$?

2. Is $|A \cup B| > |A| > |A \cap B|$ always true?

3. Let A be the set of all n-bit binary strings, A_i be the set of all n-bit binary strings with i ones. Is $(A_1, A_2, \ldots, A_i, \ldots, A_n)$ a partition of A?

4. Let $A = \{x,y\}$. What is pow(A)xpow(A) and |pow(A)xpow(A)|?

This Lecture

- Basic Definitions
- Operations on Sets
- Set Identities
- Russell’s Paradox

Set Identities

Some basic properties of sets, which are true for all sets.

$$A \cap B \subseteq A$$

$$A \subseteq A \cup B$$

if $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$

$$A \cap \overline{A} = \emptyset$$

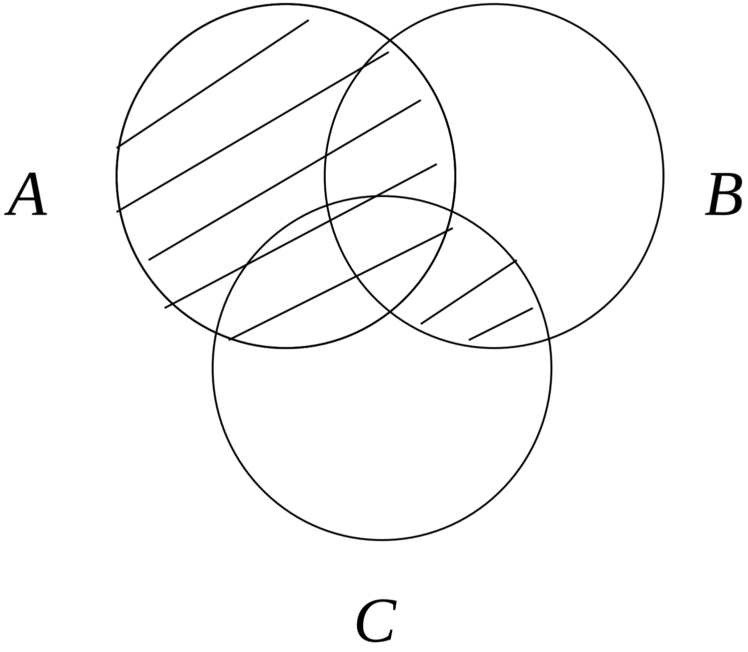
$$\overline{\overline{A}} = A$$

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

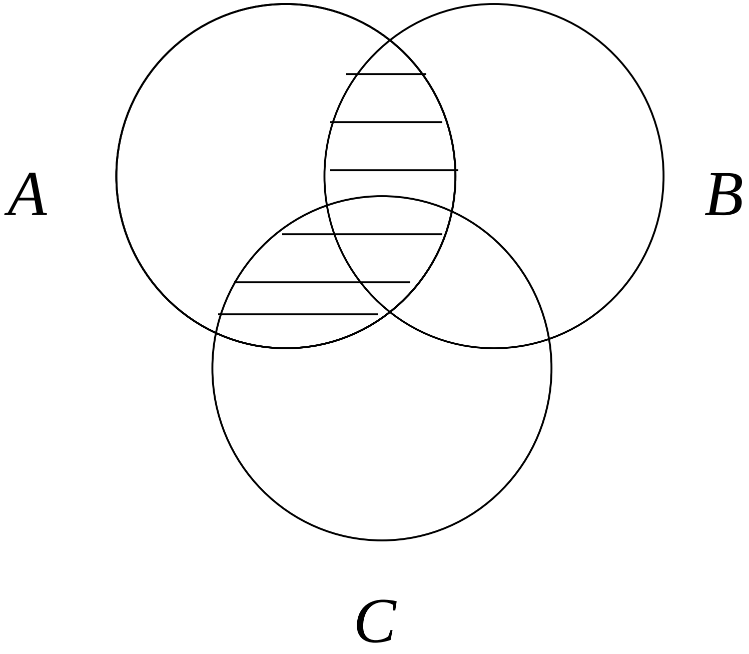
Set Identities

Distributive Law: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (1)

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (2)



(1)

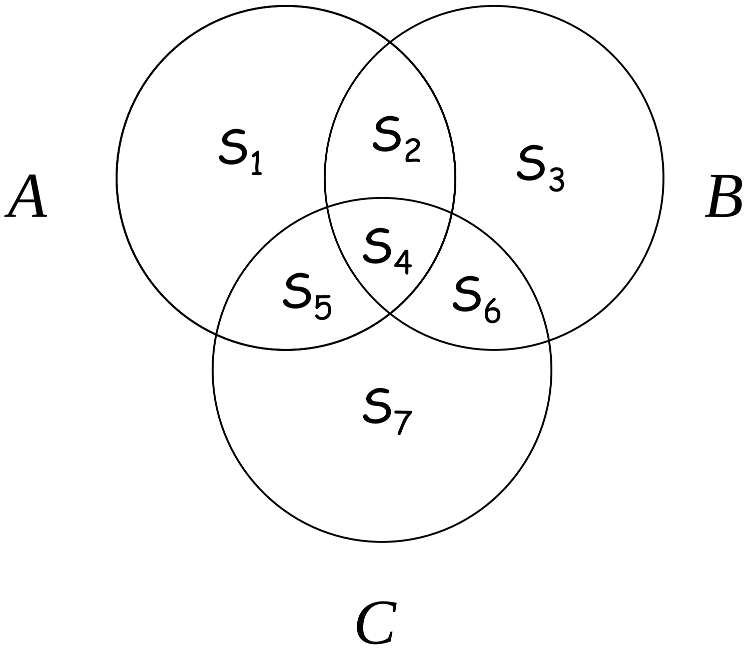


(2)

Set Identities

Distributive Law: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

We can also verify this law more carefully



L.H.S

$A = S_1 \cup S_2 \cup S_4 \cup S_5$

$B \cap C = S_4 \cup S_6$

$A \cup (B \cap C) = S_1 \cup S_2 \cup S_4 \cup S_5 \cup S_6$

R.H.S.

$(A \cup B) = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6$

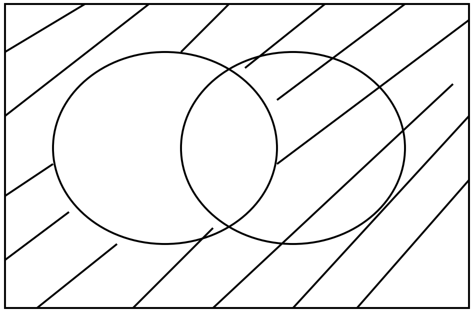
$(A \cup C) = S_1 \cup S_2 \cup S_4 \cup S_5 \cup S_6 \cup S_7$

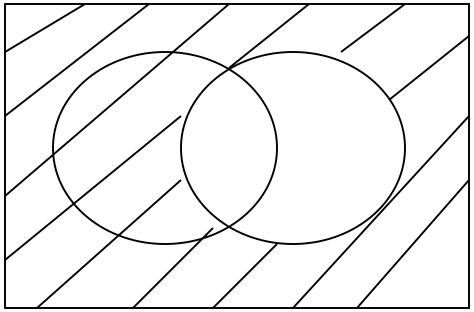
$(A \cup B) \cap (A \cup C) = S_1 \cup S_2 \cup S_4 \cup S_5 \cup S_6$

Set Identities

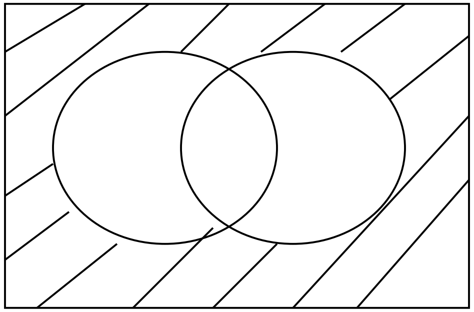
De Morgan's Law: $\overline{A \cup B} = \overline{A} \cap \overline{B}$

$\overline{A}$





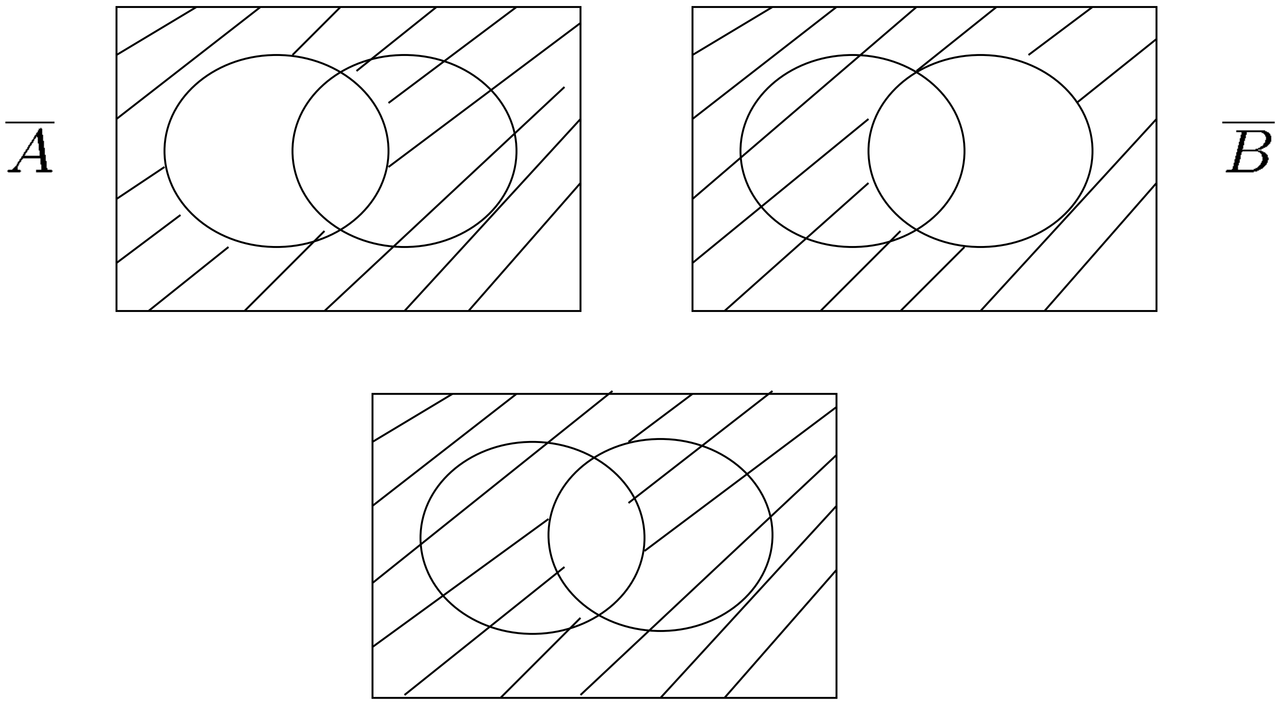
$\overline{B}$



$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

Set Identities

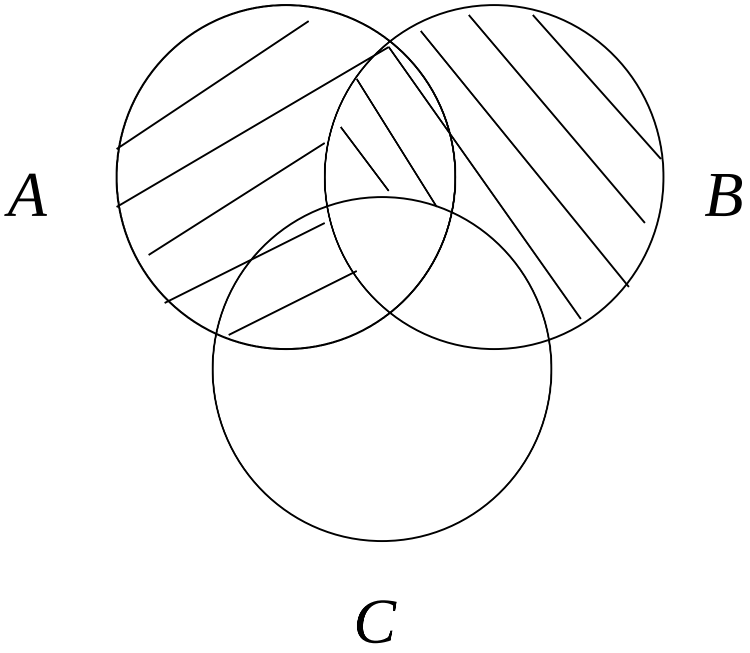
De Morgan's Law: $\overline{A \cap B} = \overline{A} \cup \overline{B}$



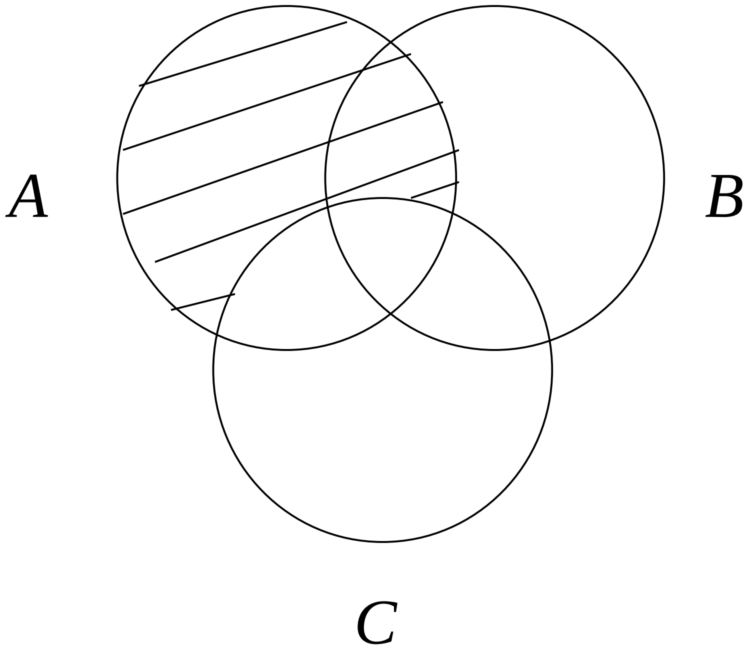
$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Disproof

$$(A - B) \cup (B - C) = A - C?$$



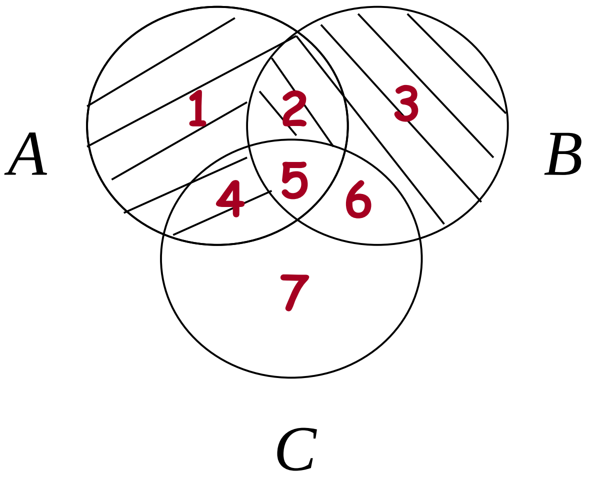
L.H.S

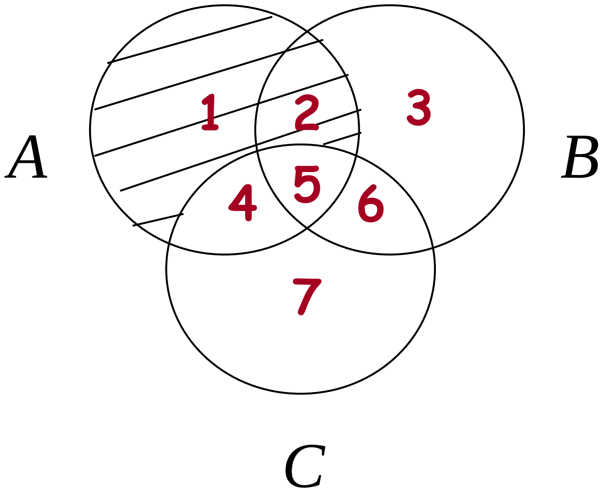


R.H.S

Disproof

$$(A - B) \cup (B - C) = A - C?$$





We can easily construct a **counterexample** to the equality, by putting a number in each region in the figure.

Let A = {1,2,4,5}, B = {2,3,5,6}, C = {4,5,6,7}.

Then we see that L.H.S = {1,2,3,4} and R.H.S = {1,2}.

Algebraic Proof

Sometimes when we know some rules, we can use them to prove new rules without drawing figures.

e.g. we can prove $\overline{(\overline{A} \cap \overline{B})} = A \cup B$ without drawing figures.

$$\overline{(\overline{A} \cap \overline{B})} = \overline{\overline{A}} \cup \overline{\overline{B}}$$ by using DeMorgan's rule on $\overline{A}$ and $\overline{B}$

$$= A \cup B$$

Algebraic Proof

$$\overline{((A \cup C) \cap (B \cup C))} = (\overline{A} \cup \overline{B}) \cup \overline{C}?$$

$$\overline{((A \cup C) \cap (B \cup C))}$$

$$= \overline{(A \cup C)} \cup \overline{(B \cup C)}$$ by DeMorgan's law on A U C and B U C

$$= (\overline{A} \cap \overline{C}) \cup \overline{(B \cup C)}$$ by DeMorgan's law on the first half

$$= (\overline{A} \cap \overline{C}) \cup (\overline{B} \cap \overline{C})$$ by DeMorgan's law on the second half

$$= (\overline{A} \cup \overline{B}) \cap \overline{C}$$ by distributive law

$$\neq (\overline{A} \cup \overline{B}) \cup \overline{C}$$

Exercises

$$A - (A \cap B) = A - B?$$

$$(A \cup B) - C = (A - C) \cup (B - C)?$$

$$\overline{(A \cup B \cup C)} = \overline{A} \cap \overline{B} \cap \overline{C}?$$

This Lecture

- Basic Definitions
- Operations on Sets
- Set Identities
- Russell's Paradox

Russell's Paradox (Optional)

$$\text{Let } W ::= \left\{ S \in \text{Sets} \mid S \notin S \right\}$$

In words, W is the set that contains all the sets that don't contain themselves.

Is W in W?

If W is in W, then W contains itself.

But W contains only those sets that don't contain themselves.

So W is not in W.

If W is not in W, then W does not contain itself.

But W contains those sets that don't contain themselves.

So W is in W.

What's wrong???

Barber's Paradox (Optional)

There is a male barber who shaves all those men,

and only those men, who do not shave themselves.

Does the barber shave himself?

Suppose the barber shaves himself.
But the barber only shaves those men who don't shave themselves.
Since the barber shaves himself, he does not shave himself.

Suppose the barber does not shave himself.
But the barber shaves those men who don't shave themselves.
Since the barber does not shave himself, he shaves himself.

What's wrong???

Solution to Russell's Paradox (Optional)

A man either shaves himself or does not shave himself.

A barber neither shaves himself nor not shaves himself.

Perhaps such a barber does not exist?

Actually that's the way out of this paradox.

Going back to the barber's paradox,

we conclude that W cannot be a set,

because every set either contains itself or does not contain itself,

but either case cannot happen for W.

This paradox tells us that not everything we define is a set.

Later on mathematicians define sets more carefully,

e.g. using sets that we already know.

Halting Problem

Now we study one of the most famous problems in computer science.

The halting problem: Can we write a program which detects an infinite loop?

We want a program H that given any program P and input I:

H(P,I) returns "halt" if P will terminate given input I;

H(P,I) returns "loop forever" if P will not terminate given input I.

The halting problem: Does such a program H exist?

Note that the program H can not just simulate the program P on input I;

- if P halts on I, then H can return halt successfully;
- but if P loops forever on I, then H will also loop forever.

Halting Problem

We want a program H that given any program P and input I:

H(P,I) returns "halt" if P will terminate given input I;

H(P,I) returns "loop forever" if P will not terminate given input I.

The halting problem: Does such a program H exist?

Proof by contradiction:

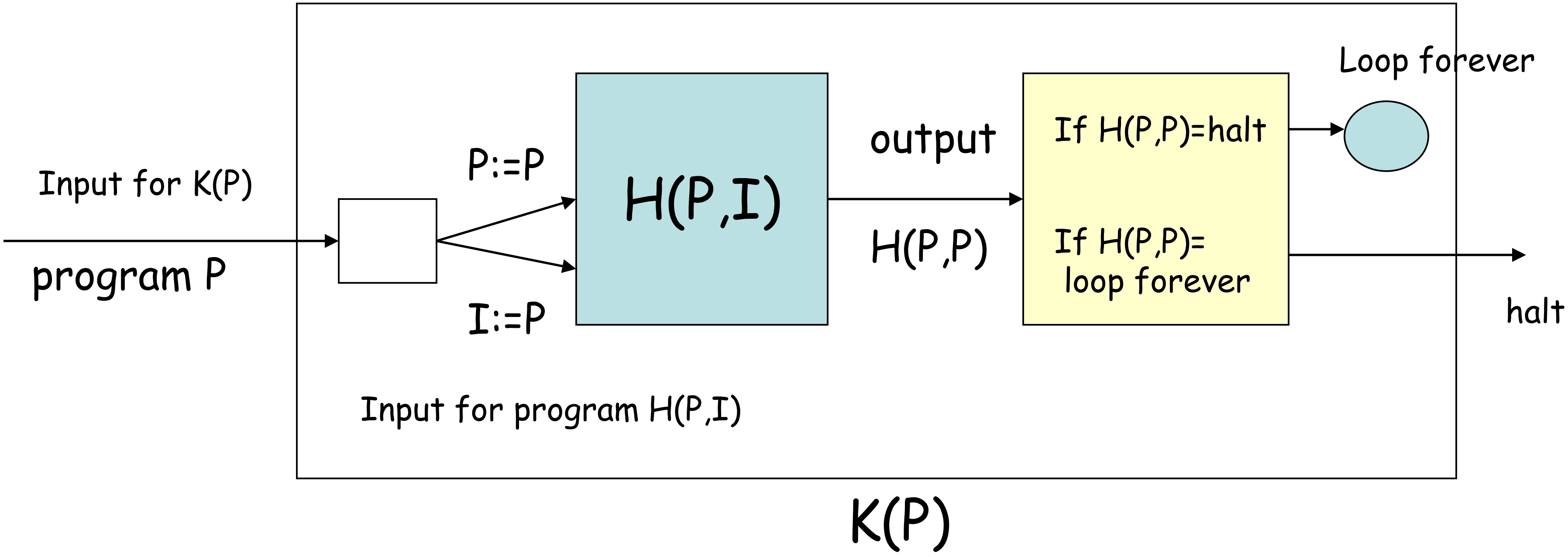
- Suppose, by way of contradiction, that H exists.
- Both P and I are binary strings.
- H should be able to determine if P will terminate given itself as the input.
- That is, H(P,P) will either return "halt" or "loop forever".

Halting Problem

Inverter: Do the opposite to what H says.

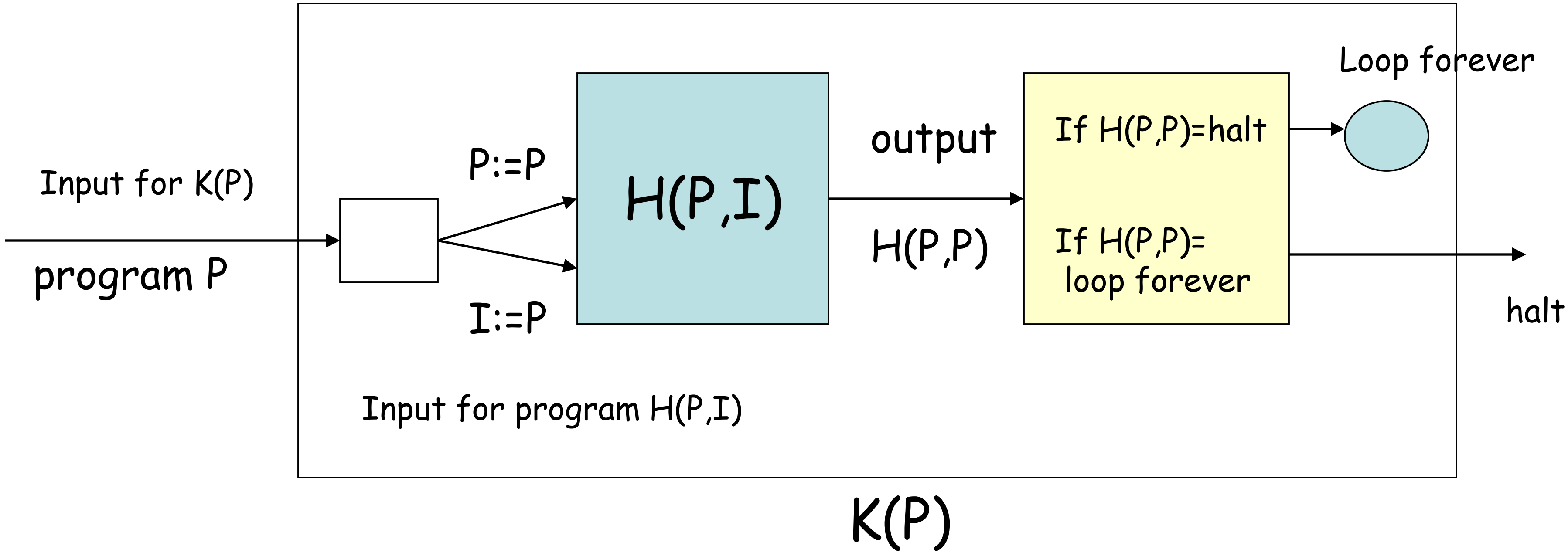
Construct an "inverter" program K which does the following given input P:

- if H(P,P) returns "halt", then K(P) will "loop forever" ;
- if H(P,P) returns "loop forever", then K(P) will "halt".



Halting Problem

What happen if K is the input to K? What is K(K)?

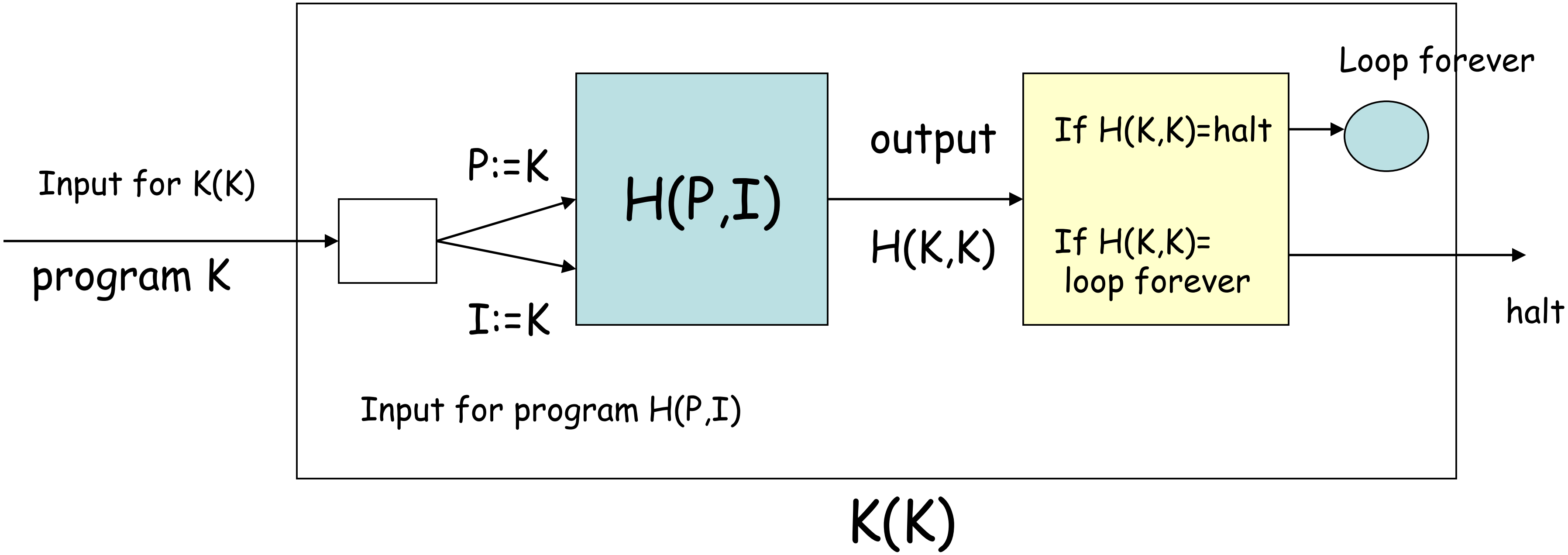


Halting Problem

What happen if K is the input to K? What is K(K)?

Case 1: Suppose H(K,K) says "halt", that is H determines K(K) will "halt".

But then K(K) will "loop forever", which is exactly the opposite to the answer.

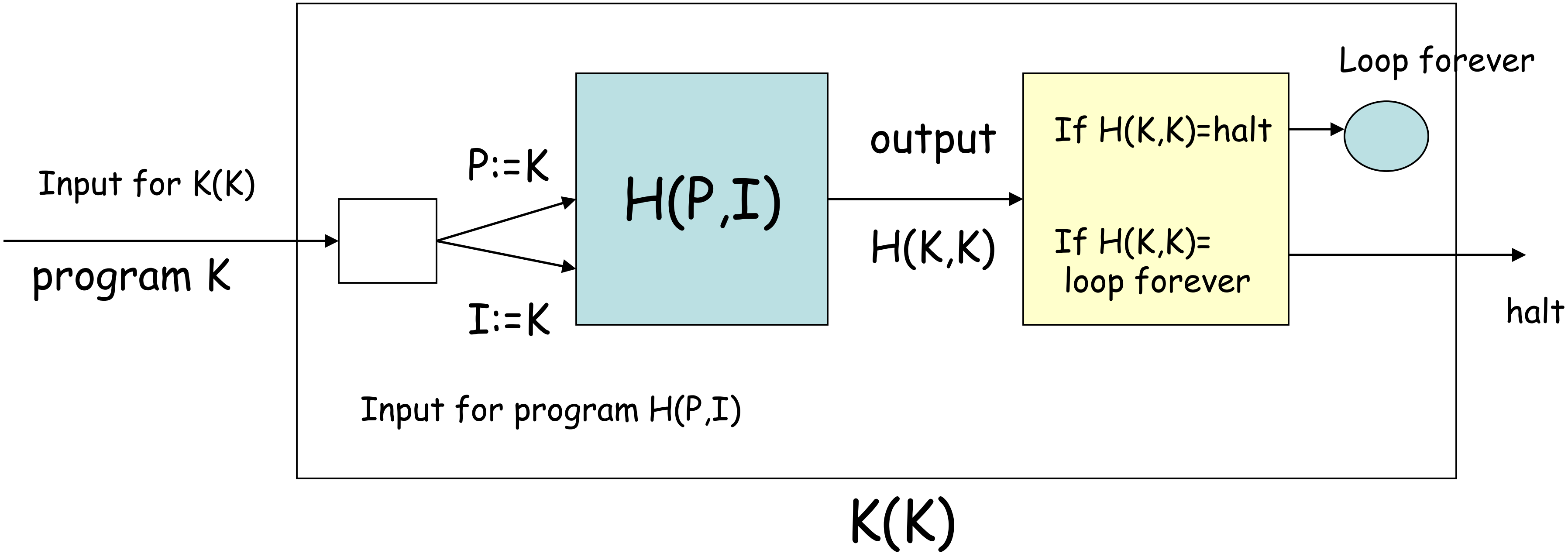


Halting Problem

What happen if K is the input to K? What is K(K)?

Case 2: Suppose H(K,K) says "loop forever", that is H determines K(K) will "loop",

But then K(K) will "halt", which is exactly the opposite to the answer.



Input for K(K)

program K

P:=K

I:=K

H(P,I)

Input for program H(P,I)

output

H(K,K)

If H(K,K)=halt

If H(K,K)=
loop forever

Loop forever

halt

K(K)

Halting Problem

In either case, H outputs a wrong answer to K(K), this contradicts that such a program exists.

Intuitively, no program can determine whether K halts when given input K, because the program K will do the opposite **"after"** you give an answer.

The proof is due to Alan Turing (1936); you will see more of such arguments in a later class.

Remarks and References

The argument used in the halting problem is called the “diagonalization” method. This was originally used by Cantor to prove that real numbers are “more” than Integers.

This method has many other applications in theoretical computer science.