Lecture 8: Co-Classes

We will cover:

Lecture 8: Co-Classes

We will cover:

- What are complementary classes.

Lecture 8: Co-Classes

We will cover:

- What are complementary classes.
- co-NP Turing machines.

Lecture 8: Co-Classes

We will cover:

- What are complementary classes.
- co-NP Turing machines.
- The relations between the complexity classes we've seen thus far.

Lecture 8: Co-Classes

We will cover:

- What are complementary classes.
- co-NP Turing machines.
- The relations between the complexity classes we've seen thus far.
- Implications of what it would mean if $SAT \in co{-}NP$.

Complementation

For any decision problem A there is a complementary problem $\overline{A}$. E.g.,

"Is n a prime number?"

is complementary to

"Is n a composite number?"

The yes-instances of A are the no-instance of $\overline{A}$, and vice versa.
Exercise: Prove that if $A \in \mathbf{P}$ then $\overline{A} \in \mathbf{P}$.

co-Classes

Given a class of languages $\mathcal{L}$ (e.g., $\mathcal{L} = P$ or $\mathcal{L} = NP$), we define its complement as

$$co\text{-}\mathcal{L} = \{\overline{L} \mid L \in \mathcal{L}\} = \{\Sigma^* \setminus L \mid L \in \mathcal{L}\}.$$

co-Classes

Given a class of languages $\mathcal{L}$ (e.g., $\mathcal{L} = P$ or $\mathcal{L} = NP$), we define its complement as

$$co\text{-}\mathcal{L} = \{\overline{L} \mid L \in \mathcal{L}\} = \{\Sigma^* \setminus L \mid L \in \mathcal{L}\}.$$

Observe that $P = co\text{-}P$. (why?)

co-Classes

Given a class of languages $\mathcal{L}$ (e.g., $\mathcal{L} = P$ or $\mathcal{L} = NP$), we define its complement as

$$co\text{-}\mathcal{L} = \{\overline{L} \mid L \in \mathcal{L}\} = \{\Sigma^* \setminus L \mid L \in \mathcal{L}\}.$$

Observe that $P = co\text{-}P$. (why?)
What about NP?

co-NP machines

It is often useful to think of "co-NP non-deterministic Turing machines".

co-NP machines

It is often useful to think of "co-NP non-deterministic Turing machines".

In a co-NP NDTM, an input is accepted only if *all* computation paths halt in Y and is rejected if *there exists* a path halting in N.

co-NP machines

It is often useful to think of "co-NP non-deterministic Turing machines".

In a co-NP NDTM, an input is accepted only if *all* computation paths halt in Y and is rejected if *there exists* a path halting in N.

Having a co-NP NDTM machine for L is equivalent to having a (standard) NDTM machine for $\overline{L}$. (Exercise!)

co-NP

Let us look at some examples:

co-NP

Let us look at some examples:

- $\overline{SAT} \in co\text{-}NP$.

co-NP

Let us look at some examples:

- $\overline{SAT} \in co\text{-}NP$.
- $BEQ = \{\psi_1, \psi_2 \mid \text{For every assignment } v,\ v(\psi_1) = v(\psi_2)\}$.

co-NP

Let us look at some examples:

- $\overline{SAT} \in \textit{co-NP}$.
- $BEQ = \{\psi_1, \psi_2 \mid \text{For every assignment } v,\ v(\psi_1) = v(\psi_2)\}$.
- All problems in P.

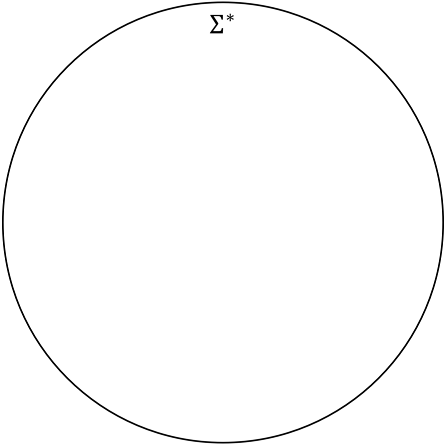
co-NP

Let us look at some examples:

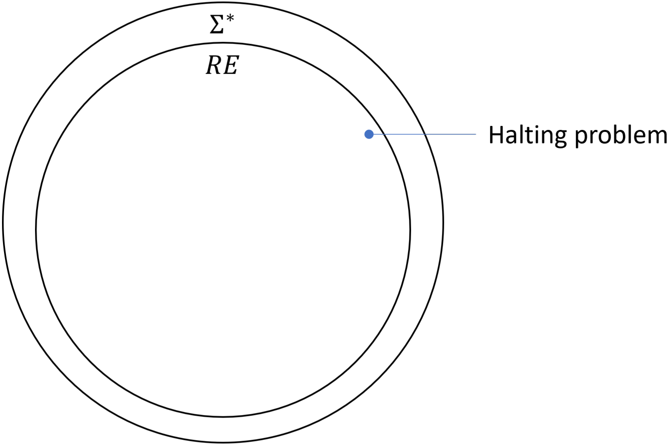
- $\overline{SAT} \in co\text{-}NP$.
- $BEQ = \{\psi_1, \psi_2 \mid \text{For every assignment } v,\ v(\psi_1) = v(\psi_2)\}$.
- All problems in P.

It is not known if $NP = co-NP$, or if $P = NP \cap co-NP$.

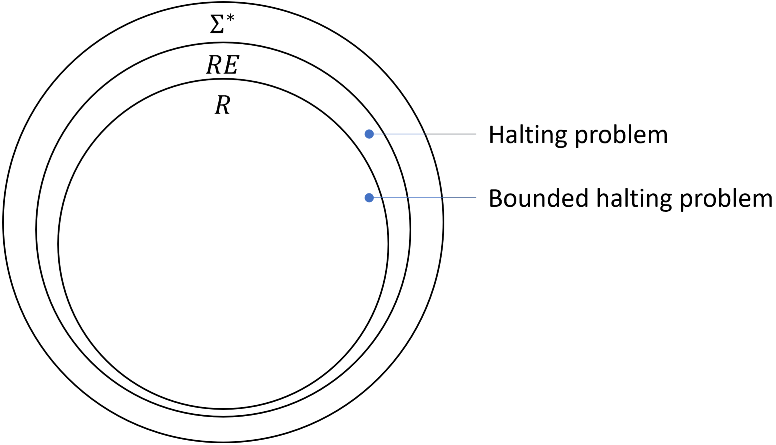
The world, as we know it



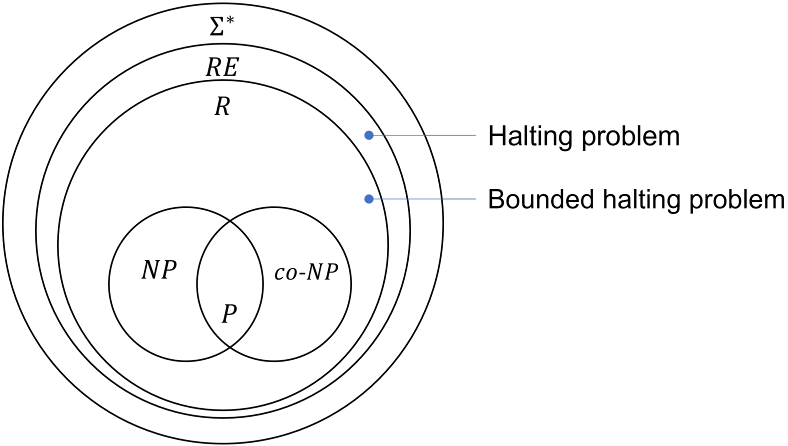
The world, as we know it



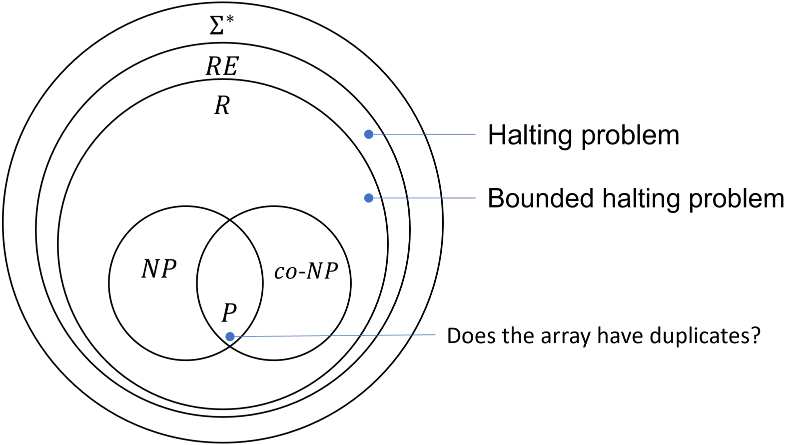
The world, as we know it



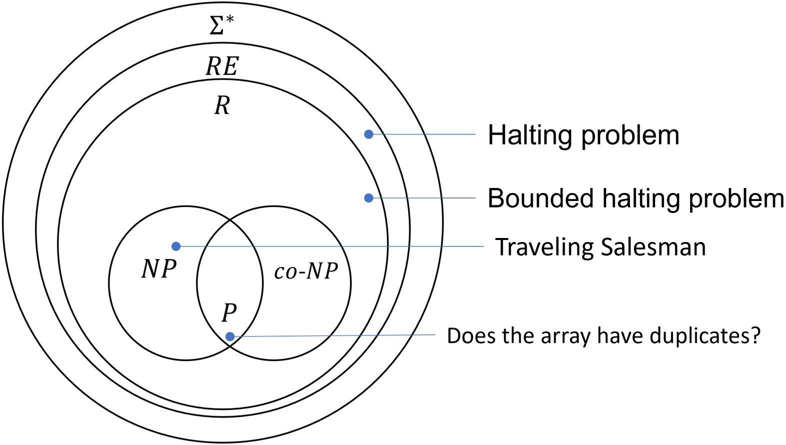
The world, as we know it



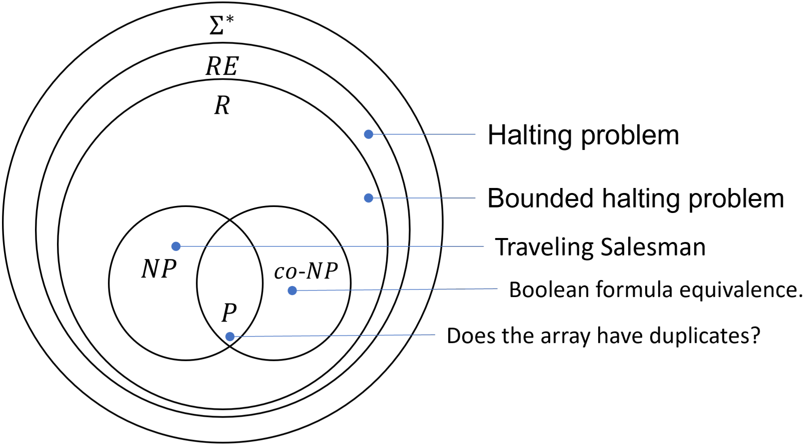
The world, as we know it



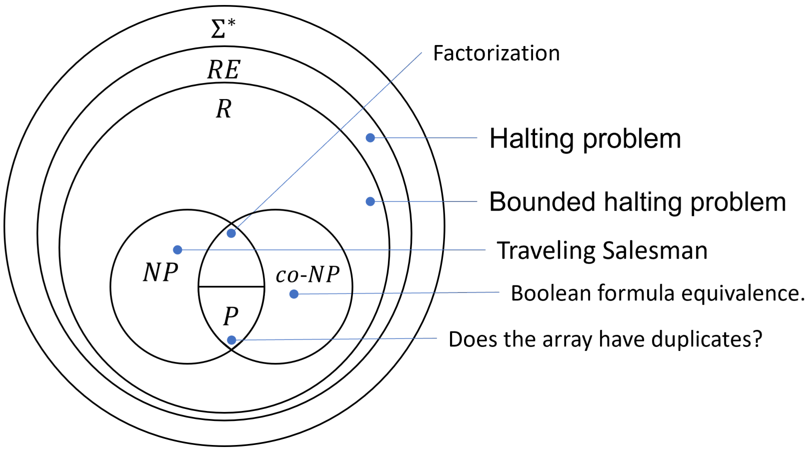
The world, as we know it



The world, as we know it



The world, as we know it



Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co-NP$ then $NP = co-NP$.

Proof.

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co{-}NP$ then $NP = co{-}NP$.

Proof.

Let $L' \in NP$.

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co-NP$ then $NP = co-NP$.

Proof.

Let $L' \in NP$.

Since $L \in co-NP$, it has a co-NP NDTM M.

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co-NP$ then $NP = co-NP$.

Proof.

Let $L' \in NP$.
Since $L \in co-NP$, it has a co-NP NDTM M.
We have that $L' \leq_p L$ and and thus there exists a DTM M' that reduces L' to L.

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co-NP$ then $NP = co-NP$.

Proof.

Let $L' \in NP$.
Since $L \in co-NP$, it has a co-NP NDTM M.
We have that $L' \leq_p L$ and and thus there exists a DTM M' that reduces L' to L.
Therefore, we can design a co-NP NDTM for L': We first run M', rewind the head, and then run M.

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co{-}NP$ then $NP = co{-}NP$.

Proof.

Let $L' \in NP$.
Since $L \in co{-}NP$, it has a co-NP NDTM M.
We have that $L' \leq_p L$ and and thus there exists a DTM M' that reduces L' to L.
Therefore, we can design a co-NP NDTM for L': We first run M', rewind the head, and then run M.
This means that $NP \subseteq co{-}NP$.

Can SAT be in co-NP?

We saw that SAT is NP-complete. What does that mean about co-NP?

Theorem

If $L \in NPC$ and $L \in co{-}NP$ then $NP = co{-}NP$.

Proof.

Let $L' \in NP$.
Since $L \in co{-}NP$, it has a co-NP NDTM M.
We have that $L' \leq_p L$ and and thus there exists a DTM M' that reduces L' to L.
Therefore, we can design a co-NP NDTM for L': We first run M', rewind the head, and then run M.
This means that $NP \subseteq co{-}NP$.
By complementing each language in both sides, we get that $co{-}NP \subseteq co{-}co{-}NP$, which gives $co{-}NP \subseteq NP$. $\square$

Lecture 8: Recap

Today we saw:

Lecture 8: Recap

Today we saw:

- The co-NP complexity class.

Lecture 8: Recap

Today we saw:

- The co-NP complexity class.
- We can prove hardness of other problems by reductions from NP-hard problems like SAT.

Lecture 8: Recap

Today we saw:

- The co-NP complexity class.
- We can prove hardness of other problems by reductions from NP-hard problems like SAT.

Next lecture: Space Complexity.