Cook’s Theorem

The Foundation of NP-Completeness
An alphabet \( \Sigma = \{s_1, s_2, \ldots, s_k\} \) is a set of symbols.

The set \( \Sigma^* \) is the set of all strings created using the symbols of \( \Sigma \).

A problem is a subset \( X \subseteq \Sigma^* \).

An algorithm A solves X, if given any string \( x \in \Sigma^* \) A accepts x if and only if \( x \in X \).
Decision Problems & Algorithms

- **Decision Problems are Set-Membership Problems**
- The sets are all assumed to be sets of strings
- Deterministic Algorithms **Accept** or **Reject** strings
- Non-Deterministic Algorithms **Accept** strings, but generally do not **Reject** strings
The Elements of NP

- A problem $X$ is an element of NP if there is a non-deterministic polynomial-time algorithm that solves $X$.
- All algorithms can be represented as Turing Machines
- Therefore, each $X \in \text{NP}$ has a Non-Deterministic Turing Machine $M_X$ that accepts the elements of $X$ in polynomial time
What The Theorem Must Do

✗ Given a problem $X$, and a string $x \in \Sigma^*$, create a Boolean expression in polynomial time
✗ The Boolean expression must be in CNF form
✗ The Boolean expression must be satisfiable if and only if $x \in X$
✗ We can make use of $M_X$ and the polynomial time bound $p(n)$ of $M_X$. 
The Structure of $M_X I$

- Any Turing Machine has a State Diagram, and a tape
- The tape is assumed to be a linear set of squares, each containing a symbol
- There is a blank symbol $b$, that occupies any un-accessed squares of the tape
- The tape extends indefinitely in both directions
The Structure of $M_X$ II

- The strings to be processed by $M_X$ are formed from the symbols from the set $\Sigma$.
- The symbols of $\Sigma$ may appear on the tape.
- There may be additional tape-only symbols, taken from the set $\Gamma$.
- The blank $b \notin \Sigma$ and $b \notin \Gamma$. 
The Structure of $M_X$ III

- The set of states is $Q$
- There are three distinguished states of $Q$, $q_0$, $q_y$ and $q_n$. ($q_n$ is not strictly required.)
- When $M_X$ is in state $q_y$ or $q_n$, it is halted and no more transitions are possible.
- $q_0$ is the start state
The Structure of $M_X$ IV

- The set $\delta$ determines the legal transitions
- The set of tape symbols $T = \Sigma \cup \Gamma \cup \{b\} \times \cup$
- $\delta \subseteq Q \times T \times Q \times T \times \{L,R\}$
- If $(q_i, s_j, q_k, s_l, L) \in \delta$, then when $M_X$ is in state $q_i$ and scanning symbol $s_j$ on its tape, then it is legal for $M$ to write $s_l$ into the current tape square, move one square to the left, and enter state $q_k$
The Structure of $M_X V$

- More than one element of $\delta$ might apply in a given configuration.
- The machine starts in state $q_0$ with the string $x$ to be recognized on the tape.
- The read-write head is scanning the leftmost symbol of $x$ (square 1) and blanks appear to the left and to the right of $x$. 
The Structure of $M_X$ VI

✗ Each application of an element of $\delta$ counts as one “unit of time” and is called a transition

✗ To accept $x$, it must be possible for $M_X$ to enter state $q_y$ in no more than $p(n)$ transitions, where $n$ is the length of $x$.

✗ It is not necessary for $M_X$ to enter $q_n$ to reject $x$. 
Starting Position

Read-Write Head

Square Number
Input String

... b b b b b b b b b x_1 x_2 x_3 ... x_n b b b ...
The Transform

- The accepting sequence of the machine $M_X$ must be modeled in CNF clauses.
- First it is necessary to define the a number of variables representing conditions in $M_X$
- Then it is necessary to define clauses to represent the legal accepting sequences of $M_X$
The Variables

- $Q[i,q_k]$ where $i$ runs from 0 to $p(n)$ and $q_k$ runs through all states of $M_X$
- $H[i,j]$ where $I$ runs from 0 to $p(n)$ and $j$ runs from $-p(n)$ through $p(n)+1$
- $S[i,j,s_k]$ where $i$ runs from 0 to $p(n)$, $j$ runs from $-p(n)$ through $p(n)+1$, and $s_k$ runs through all symbols of $T$ (tape symbols)
The Meaning of the Variables

- $Q[i,q_j]$ means that at time $i$, $M_x$ is in state $q_j$
- $H[i,j]$ means that at time $i$, $M_x$ is scanning tape square $j$. Note that in $p(n)$ transitions, the read-write head can move at most distance $p(n)$ from its starting point.
- $S[i,j,s_k]$ means that at time $I$, the contents of tape square $j$ is $s_k$. 
Clause Groups

- **G1** - Guarantee that at each time $i$, $M_x$ is in one and only one state
- **G2** - Guarantee that at each time $i$, $M_x$ is scanning one and only one tape square
- **G3** - Guarantee that at each time $i$, there is one and only one symbol in each tape square of the used tape
Clause Groups II

✗ G4 - Guarantee that the machine starts in q0 with x properly positioned on the tape and the read-write head properly positioned.

✗ G5 - Guarantee that by time p(n) $M_X$ has entered $q_y$.

✗ G6 - Guarantee that the transitions are applied properly.
Group G1

✗ For each time i, add the clause
\{Q[i,q_1],Q[i,q_2], \ldots , Q[i,q_t]\} where t is the number of states in Q.

✗ For each time i, add the set of clauses
\{Q[i,q_k],Q[i,q_j]\} where k and j, taken together run through all pairs of states of Q. If Q has t states then t(t+1)/2 clauses are required for each time i.
G1 Clause Meanings

× The first part guarantees that at each time \( i \), \( M_X \) is in at least one state.

× The second part (with the paired barred variables) guarantees that \( M_X \) is not in more than one state at time \( i \).

× The time \( i \) runs from 0 through \( p(n) \)
Group G2

✗ For each time i, add the clause:
   \{H[i,-p(n)], H[i,-p(n)+1],\ldots,H[i,p(n)+1]\}

✗ For each time i, let j and k run through all possible pairs of tape squares from -p(n) to p(n)+1. For each pair (j,k), and each time i, add the clause \{H[i,j], H[i,k]\}. 
G2 Clause Meanings

✗ The first clause says that $M_X$ must be scanning at least one tape square at every time $i$.

✗ The second set of clauses says that $M_X$ cannot be scanning more than one tape square at any given time $i$. 
Group G3

- Let i run through all times from 0 to p(n) and j run through all tape squares from \(-p(n)\) through \(p(n)+1\). (There are \(p(n) \times 2(p(n)+1)\) combinations.

- For each (i,j) add \(\{S[i,j,s_0], S[i,j,s_1], \ldots, S[i,j,s_k]\}\), where \(s_0, s_1, \ldots, s_k\) run through all tape symbols in T.
Let \( l \) and \( m \) run through all pairs of tape symbols. If there are \( k \) tape symbols, then there are \( k(k+1)/2 \) pairs.

For each combination \((i,j)\) and each pair \((l,m)\), add the following clause \[ \{ S[i,j,l], S[i,j,m] \} \]
G3 Clause Meanings

✖ G3 Clauses model the behavior of the tape
✖ The first set of clauses guarantees that at any time i, each tape square contains at least one tape symbol. We are concerned only about squares numbered from \(-p(n)\) through \(p(n)+1\).
✖ The second set of clauses guarantees that at any time i, no tape square contains more than one tape symbol.
Group G4

- Add \(\{Q[0,q_0]\}\)
- Add \(\{H[0,1]\}\)
- Add \(\{S[0,1,x_1]\}, \{S[0,1,x_2]\}, \ldots ,\{S[0,n,x_n]\}\)
- Add \(\{S[0,0,b]\}\)
- Add \(\{S[0,n+1,b]\}, \{S[0,n+2,b]\}, \ldots ,\{S[0,p(n)+1,b]\}\)
- Add \(\{S[0,-1,b]\}, \{S[0,-2,b]\}, \ldots ,\{S[0,-p(n),b]\}\)
G4 Clause Meanings

✗ The first clause guarantees we start in state 0.
✗ The second clause guarantees the read-write head starts with square 1.
✗ The next set of clauses guarantees that the input string is on the tape in the correct position at time 0.
✗ The final sets of clauses guarantee that at time 0, the rest of the tape is blank.
Group G5

✗ Add \{Q[p(n),q_y]\}

✗ Once we enter state \(q_y\), no further transitions are allowed.

✗ This clause guarantees that we have entered state \(q_y\) either at some time prior to \(p(n)\) or at time \(p(n)\).

✗ Entering \(q_y\) causes \(M_X\) to accept its input.
Let \((q_a, s_b, q_c, s_d, e)\) be an element of \(\delta\), where \(e\) is an element of \(\{L, R\}\).

We need to model the following logical statement in CNF form:
If the current time is \(i\) and \(M_X\) is in state \(q_a\) and \(M_X\) is scanning tape square \(j\) and tape square \(j\) contains symbol \(s_b\), then at time \(i+1\), \(M_X\) will be in state \(q_b\), tape square \(j\) will contain \(s_d\) and \(M_X\) will be scanning either square \(j+1\) or \(j-1\) depending on \(e\).
Group G6 Continued

✗ If P then Q is logically equivalent to
~P OR Q.

✗ Assume e=L, then using the variables we get:
~(Q[i,q_a] AND H[i,j] AND S[i,j,s_b]) OR
(Q[i+1,q_b] AND H[i+1,j+1] AND S[i+1,j,s_d])

✗ For e=R,
~(Q[i,q_a] AND H[i,j] AND S[i,j,s_b]) OR
(Q[i+1,q_b] AND H[i+1,j-1] AND S[i+1,j,s_d])
Deriving CNF Form

\[ \neg(Q[i,q_a] \land H[i,j] \land S[i,j,s_b]) \lor (Q[i+1,q_b] \land H[i+1,j+1] \land S[i+1,j,s_d]) \]

\[ \text{DeMorgan’s Law:} \]
\[ (Q[i,q_a] \lor \overline{H[i,j]} \lor \overline{S[i,j,s_b]}) \lor (Q[i+1,q_b] \land H[i+1,j+1] \land S[i+1,j,s_d]) \]

\[ \text{Apply Distributive Law to obtain Three Clauses} \]
Final Group

\( e=L \)
\[
\{ \overline{Q[i,q_a]}, \overline{H[i,j]}, \overline{S[i,j,s_b]}, Q[i+1,q_b] \}
\{ \overline{Q[i,q_a]}, \overline{H[i,j]}, \overline{S[i,j,s_b]}, H[i+1,j+1] \}
\{ \overline{Q[i,q_a]}, \overline{H[i,j]}, \overline{S[i,j,s_b]}, S[i+1,j,s_d] \}
\]

\( e=R \)
\[
\{ \overline{Q[i,q_a]}, \overline{H[i,j]}, \overline{S[i,j,s_b]}, Q[i+1,q_b] \}
\{ \overline{Q[i,q_a]}, \overline{H[i,j]}, \overline{S[i,j,s_b]}, H[i+1,j-1] \}
\{ \overline{Q[i,q_a]}, \overline{H[i,j]}, \overline{S[i,j,s_b]}, S[i+1,j,s_d] \}
\]
G6 The Final Step

✗ For each element of $\delta$, add one three-clause group for each combination of time $i$, and tape square $j$.

✗ For each element of $\delta$, we generate $3p(n)2(p(n)+1)$ clauses.
The final Boolean Expression is

\[ E = G_1 \cup G_2 \cup G_3 \cup G_4 \cup G_5 \cup G_6 \]

E is satisfiable if and only if \( M_X \) accepts

\[ x = x_1 x_2 x_3 \ldots x_n \]

E can be constructed from \( M_X \) and \( x \) in polynomial time.
Cook’s Theorem

The Foundation of NP-Completeness
Theoretical Foundations

※ An alphabet $\Sigma = \{s_1, s_2, \ldots, s_k\}$ is a set of symbols.
※ The set $\Sigma^*$ is the set of all strings created using the symbols of $\Sigma$.
※ A problem is a subset $X \subseteq \Sigma^*$.
※ An algorithm $A$ solves $X$, if given any string $x \in \Sigma^*$, $A$ accepts $x$ if and only if $x \in X$. 
Decision Problems & Algorithms

Decision Problems are Set-Membership Problems

The sets are all assumed to be sets of strings

Deterministic Algorithms Accept or Reject strings

Non-Deterministic Algorithms Accept strings, but generally do not Reject strings
The Elements of NP

✗ A problem X is an element of NP if there is a non-deterministic polynomial-time algorithm that solves X.

✗ All algorithms can be represented as Turing Machines

✗ Therefore, each $X \in \text{NP}$ has a Non-Deterministic Turing Machine $M_X$ that accepts the elements of X in polynomial time
What The Theorem Must Do

✗ Given a problem $X$, and a string $x \in \Sigma^*$, create a Boolean expression in polynomial time

✗ The Boolean expression must be in CNF form

✗ The Boolean expression must be satisfiable if and only if $x \in X$

✗ We can make use of $M_X$ and the polynomial time bound $p(n)$ of $M_X$. 
The Structure of $M_X$ I

✗ Any Turing Machine has a State Diagram, and a tape
✗ The tape is assumed to be a linear set of squares, each containing a symbol
✗ There is a blank symbol b, that occupies any un-accessed squares of the tape
✗ The tape extends indefinitely in both directions
The Structure of $M_X$ II

$x$ The strings to be processed by $M_X$ are formed from the symbols from the set $\Sigma$

$x$ The symbols of $\Sigma$ may appear on the tape

$x$ There may be additional tape-only symbols, taken from the set $\Gamma$

$x$ The blank $b \notin \Sigma$ and $b \notin \Gamma$. 
The Structure of $M_X$ III

- The set of states is $Q$
- There are three distinguished states of $Q$, $q_0$, $q_y$ and $q_n$. ($q_n$ is not strictly required.)
- When $M_X$ is in state $q_y$ or $q_n$, it is halted and no more transitions are possible.
- $q_0$ is the start state
The Structure of $M_X$ IV

- The set $\delta$ determines the legal transitions
- The set of tape symbols $T = \Sigma \cup \Gamma \cup \{b\} \times \cup$
- $\delta \subseteq Q \times T \times Q \times T \times \{L,R\}$
- If $(q_i, s_j, q_k, s_l, L) \in \delta$, then when $M_X$ is in state $q_i$ and scanning symbol $s_j$ on its tape, then it is legal for $M$ to write $s_l$ into the current tape square, move one square to the left, and enter state $q_k$
The Structure of $M_X$ V

✗ More than one element of $\delta$ might apply in a given configuration

✗ The machine starts in state $q_0$ with the string $x$ to be recognized on the tape.

✗ The read-write head is scanning the leftmost symbol of $x$ (square 1) and blanks appear to the left and to the right of $x$. 
The Structure of $M_X$ VI

$x$ Each application of an element of $\delta$ counts as one “unit of time” and is called a transition.

$x$ To accept $x$, it must be possible for $M_X$ to enter state $q_y$ in no more than $p(n)$ transitions, where $n$ is the length of $x$.

$x$ It is not necessary for $M_X$ to enter $q_n$ to reject $x$. 
Starting Position

Read-Write

Head

\[
\begin{array}{ccccccc}
  \ldots & b & b & b & b & b & b & b & b & b & x_1 & x_2 & x_3 & \ldots & x_n & b & b & b & \ldots \\
  \ldots & -1 & 0 & 1 & 2 & 3 & \ldots & n & \ldots \\
\end{array}
\]

Square Number

Input String
The Transform

- The accepting sequence of the machine \( M_X \) must be modeled in CNF clauses.
- First it is necessary to define the a number of variables representing conditions in \( M_X \).
- Then it is necessary to define clauses to represent the legal accepting sequences of \( M_X \).
The Variables

\[ Q[i,q_k] \] where \( i \) runs from 0 to \( p(n) \) and \( q_k \) runs through all states of \( M_x \)

\[ H[i,j] \] where \( I \) runs from 0 to \( p(n) \) and \( j \) runs from \(-p(n)\) through \( p(n)+1 \)

\[ S[i,j,s_k] \] where \( i \) runs from 0 to \( p(n) \), \( j \) runs from \(-p(n)\) through \( p(n)+1 \), and \( s_k \) runs through all symbols of \( T \) (tape symbols)
The Meaning of the Variables

- $Q[i,q_j]$ means that at time $i$, $M_X$ is in state $q_j$.
- $H[i,j]$ means that at time $i$, $M_X$ is scanning tape square $j$. Note that in $p(n)$ transitions, the read-write head can move at most distance $p(n)$ from its starting point.
- $S[i,j,s_k]$ means that at time $I$, the contents of tape square $j$ is $s_k$. 
Clause Groups

✗ G1 - Guarantee that at each time \( i \), \( M_X \) is in one and only one state

✗ G2 - Guarantee that at each time \( i \), \( M_X \) is scanning one and only one tape square

✗ G3 - Guarantee that at each time \( i \), there is one and only one symbol in each tape square of the used tape
Clause Groups II

\* G4 - Guarantee that the machine starts in q0 with x properly positioned on the tape and the read-write head properly positioned.

\* G5 - Guarantee that by time p(n) $M_x$ has entered $q_y$.

\* G6 - Guarantee that the transitions are applied properly.
Group G1

✗ For each time $i$, add the clause 
$\{Q[i,q_1], Q[i,q_2], \ldots, Q[i,q_t]\}$ where $t$ is the number of states in $Q$.

✗ For each time $i$, add the set of clauses 
$\{Q[i,q_k], Q[i,q_j]\}$ where $k$ and $j$, taken together run through all pairs of states of $Q$. If $Q$ has $t$ states then $t(t+1)/2$ clauses are required for each time $i$. 
G1 Clause Meanings

✗ The first part guarantees that at each time \( i \), \( M_X \) is in at least one state.

✗ The second part (with the paired barred variables) guarantees that \( M_X \) is not in more than one state at time \( i \).

✗ The time \( i \) runs from 0 through \( p(n) \)
Group G2

✗ For each time $i$, add the clause:
\[
\{H[i,-p(n)], H[i,-p(n)+1], \ldots, H[i,p(n)+1]\}
\]

✗ For each time $i$, let $j$ and $k$ run through all possible pairs of tape squares from $-p(n)$ to $p(n)+1$. For each pair $(j,k)$, and each time $i$, add the clause $\{H[i,j], H[i,k]\}$. 
G2 Clause Meanings

-The first clause says that $M_X$ must be scanning at least one tape square at every time $i$.

-The second set of clauses says that $M_X$ cannot be scanning more than one tape square at any given time $i$. 
Let \( i \) run through all times from 0 to \( p(n) \) and \( j \) run through all tape squares from \(-p(n)\) through \( p(n)+1\). (There are \( p(n) \times 2(p(n)+1) \) combinations.

For each \((i,j)\) add
\[
\{S[i,j,s_0], S[i,j,s_1], \ldots, S[i,j,s_k]\},
\]
where \( s_0, s_1, \ldots, s_k \) run through all tape symbols in \( T \).
Let $l$ and $m$ run through all pairs of tape symbols. If there are $k$ tape symbols, then there are $k(k+1)/2$ pairs.

For each combination $(i,j)$ and each pair $(l,m)$, add the following clause

$\{S[i,j,l], S[i,j,m]\}$
G3 Clause Meanings

✗ G3 Clauses model the behavior of the tape

✗ The first set of clauses guarantees that at any time \( i \), each tape square contains at least one tape symbol. We are concerned only about squares numbered from \(-p(n)\) through \(p(n)+1\).

✗ The second set of clauses guarantees that at any time \( i \), no tape square contains more than one tape symbol.
Group G4

× Add \{Q[0,q_0]\} 
× Add \{H[0,1]\} 
× Add \{S[0,1,x_1]\}, \{S[0,1,x_2]\}, \ldots ,\{S[0,n,x_n]\} 
× Add \{S[0,0,b]\} 
× Add \{S[0,n+1,b]\}, \{S[0,n+2,b]\}, \ldots , \{S[0,p(n)+1,b]\} 
× Add \{S[0,-1,b]\}, \{S[0,-2,b]\}, \ldots , \{S[0,-p(n),b]\}
The first clause guarantees we start in state 0.
The second clause guarantees the read-write head starts with square 1.
The next set of clauses guarantees that the input string is on the tape in the correct position at time 0.
The final sets of clauses guarantee that at time 0, the rest of the tape is blank.
Group G5

Add \{Q[p(n),q_y]\}

Once we enter state q_y, no further transitions are allowed.

This clause guarantees that we have entered state q_y either at some time prior to p(n) or at time p(n).

Entering q_y causes \(M_x\) to accept its input.
Group G6

Let \((q_a,s_b,q_c,s_d,e)\) be an element of \(\delta\), where \(e\) is an element of \(\{L,R\}\).

We need to model the following logical statement in CNF form:
If the current time is \(i\) and \(M_X\) is in state \(q_a\) and \(M_X\) is scanning tape square \(j\) and tape square \(j\) contains symbol \(s_b\), then at time \(I+1\), \(M_X\) will be in state \(q_b\), tape square \(j\) will contain \(s_d\) and \(M_X\) will be scanning either square \(j+1\) or \(j-1\) depending on \(e\).
If P then Q is logically equivalent to
\[ \sim P \text{ OR } Q. \]
Assume e=L, then using the variables we get:
\[ \sim (Q[i,q_a] \text{ AND } H[i,j] \text{ AND } S[i,j,s_b]) \text{ OR } (Q[i+1,q_b] \text{ AND } H[i+1,j+1] \text{ AND } S[i+1,j,s_d]) \]
For e=R,
\[ \sim (Q[i,q_a] \text{ AND } H[i,j] \text{ AND } S[i,j,s_b]) \text{ OR } (Q[i+1,q_b] \text{ AND } H[i+1,j-1] \text{ AND } S[i+1,j,s_d]) \]
Deriving CNF Form

\( \sim (Q[i,q_a] \text{ AND } H[i,j] \text{ AND } S[i,j,s_b]) \text{ OR } (Q[i+1,q_b] \text{ AND } H[i+1,j+1] \text{ AND } S[i+1,j,s_d]) \)

**DeMorgan’s Law:**

\( (Q[i,q_a] \text{ OR } H[i,j] \text{ OR } S[i,j,s_b]) \text{ OR } (Q[i+1,q_b] \text{ AND } H[i+1,j+1] \text{ AND } S[i+1,j,s_d]) \)

**Apply Distributive Law to obtain Three Clauses**
Final Group

\( e=L \)
\[
\{ Q[i,q_a], H[i,j], S[i,j,s_b], Q[i+1,q_b] \}
\{ Q[i,q_a], H[i,j], S[i,j,s_b], H[i+1,j+1] \}
\{ Q[i,q_a], H[i,j], S[i,j,s_b], S[i+1,j,s_d] \}
\]

\( e=R \)
\[
\{ Q[i,q_a], H[i,j], S[i,j,s_b], Q[i+1,q_b] \}
\{ Q[i,q_a], H[i,j], S[i,j,s_b], H[i+1,j-1] \}
\{ Q[i,q_a], H[i,j], S[i,j,s_b], S[i+1,j,s_d] \}
\]
G6 The Final Step

✗ For each element of $\delta$, add one three-clause group for each combination of time $i$, and tape square $j$.

✗ For each element of $\delta$, we generate $3*p(n)*2(p(n)+1)$ clauses.
Windup

✗ The final Boolean Expression is
   \[ E = G_1 \cup G_2 \cup G_3 \cup G_4 \cup G_5 \cup G_6 \]

✗ E is satisfiable if and only if \( M_x \) accepts
   \[ x = x_1 x_2 x_3 \ldots x_n \]

✗ E can be constructed from \( M_x \) and x in polynomial time.