Goal  Introduce principles of algorithm construction
Vehicle  Fun problems (games, puzzles)
Chocolate-bar Problem

How many cuts are needed to cut a chocolate bar into all its individual pieces?
Assignment and Invariants

Let $p$ be the number of pieces, and $c$ be the number of cuts.

The process of cutting the bar is modelled by:

$$p, c := p+1, c+1.$$

We observe that $(p-c)$ is an invariant. That is,

$$(p-c)[p, c := p+1, c+1] = (p+1) - (c+1) = p-c$$

Initially, $p-c$ is 1. So, number of cuts is always one less than the number of pieces.
Eg. Jealous couples:

- Three couples $Aa$, $Bb$ and $Cc$.
- One boat which can carry at most two people.
- Wives ($a$, $b$ and $c$) may not be with a man ($A$, $B$ and $C$) unless their husband is present.

Construct a sequence of actions $S_0$ satisfying

$$\{ AaBbCc | \} \quad S_0 \quad \{ |AaBbCc \} .$$
Problem Decomposition

- Exploit symmetry!

Decompose into

\[
\begin{align*}
&\{ AaBbCc \mid \} \\
&S_1 \\
; &\{ ABC \mid abc \} \\
&S_2 \\
; &\{ abc \mid ABC \} \\
&S_3 \\
&\{ \mid AaBbCc \}
\end{align*}
\]
(Impartial, Two-Person) Games

- Assume number of positions is finite.
- Assume game is guaranteed to terminate no matter how the players choose their moves.
- Game is lost when a player cannot move.

- A position is *losing* if *every* move is to a winning position.
- A position is *winning* if *there is* a move to a losing position.

Winning strategy is to maintain the invariant that one’s opponent is always left in a losing position.
Winning Strategy

Maintain the invariant that one’s opponent is always left in a losing position.

\{ \text{losing position, and not an end position} \}

make an arbitrary (legal) move

; \{ \text{winning position, i.e. not a losing position} \}

apply winning strategy

\{ \text{losing position} \}
Example Winning Strategy

One pile of matches.
Move: remove one or two matches.

Winning strategy is to maintain the invariant that one’s opponent is always left in a position where the number of matches is a multiple of 3.

\[
\begin{align*}
&\{ n \text{ is a multiple of 3, and } n \neq 0 \} \\
&\text{if } 1 \leq n \rightarrow n := n-1 \quad \square \quad 2 \leq n \rightarrow n := n-2 \fi \\
&; \quad \{ n \text{ is not a multiple of 3 } \} \\
&\quad n := n - (n \mod 3) \\
&\{ n \text{ is a multiple of 3 } \}
\end{align*}
\]
Sum Games

Given two games, each with its own rules for making a move, the sum of the games is the game described as follows.

For clarity, we call the two games the left and the right game.

A position in the sum game is the combination of a position in the left game, and a position in the right game.

A move in the sum game is a move in one of the games.
Define two functions $L$ and $R$, say, on left and right positions, respectively, in such a way that a position $(l, r)$ is a losing position exactly when $L.l = R.r$.

How do we specify the functions $L$ and $R$?
Sum Games (Cont)

First: \( L \) and \( R \) have equal values on end positions.

Second:

\[
\{ \ L.l = R.r \land (l \text{ is not an end position } \lor \ r \text{ is not an end position}) \ \}
\]

if \( l \) is not an end position \( \rightarrow \) change \( l \)

\( \Box \ r \) is not an end position \( \rightarrow \) change \( r \)

fi

\[
\{ \ L.l \neq R.r \ \}
\]

Third,

\[
\{ \ L.l \neq R.r \ \}
\]

apply winning strategy

\[
\{ \ L.l = R.r \ \}
\]
Satisfying the first two requirements:

- For end positions $l$ and $r$ of the respective games, $L.l = 0 = R.r$.
- For every $l'$ such that there is a move from $l$ to $l'$ in the left game, $L.l \neq L.l'$. Similarly, for every $r'$ such that there is a move from $r$ to $r'$ in the right game, $R.r \neq R.r'$. 
Winning strategy (third requirement):

\[ \{ \ L.l \neq R.r \ \} \]

if \( L.l < R.r \) → change \( r \)

\( \square \ R.r < L.l \) → change \( l \)

fi

\( \{ \ L.l = R.r \ \} \).
Winning strategy (third requirement):

\[
\{ \ L.l \neq R.r \ \} \\
\text{if} \ L.l < R.r \rightarrow \text{change } r \\
\square R.r < L.l \rightarrow \text{change } l \\
\text{fi} \\
\{ \ L.l = R.r \ \} .
\]

- For any number \( m \) less than \( R.r \), it is possible to move from \( r \) to a position \( r' \) such that \( R.r' = m \). (Similarly, for left game.)
Satisfying the first two requirements:

- For end positions $l$ and $r$ of the respective games, $L.l = 0 = R.r$.

- For every $l'$ such that there is a move from $l$ to $l'$ in the left game, $L.l \neq L.l'$. Similarly, for every $r'$ such that there is a move from $r$ to $r'$ in the right game, $R.r \neq R.r'$.

- For any number $m$ less than $R.r$, it is possible to move from $r$ to a position $r'$ such that $R.r' = m$. (Similarly, for left game.)
Let $p$ be a position in a game $G$. The *mex* value of $p$, denoted $\text{mex}_G.p$, is defined to be the smallest natural number, $n$, such that

- There is no legal move in the game $G$ from $p$ to a position $q$ satisfying $\text{mex}_G.q = n$.

- For every natural number $m$ less than $n$, there is a legal move in the game $G$ from $p$ to a position $q$ satisfying $\text{mex}_G.q = m$. 
Characterising Features

- Non-mathematical, easily explained problems (requiring mathematical solution)
- Minimal notation.
- Challenging problems.
- Simultaneous introduction of programming constructs and principles of program construction.