9.5
Turing complete
Equivalent to a program

Definition

A system is **Turing complete** if one can simulate a Turing machine using it.

- Programming languages (yey!).
- C++ templates system (boo).
- John Conway’s game of life.
- Many games (Minesweeper).
- Post’s correspondence problem.
Equivalent to a program

Definition

A system is **Turing complete** if one can simulate a Turing machine using it.

1. Programming languages (yey!).
2. C++ templates system (boo).
3. John Conway’s game of life.
5. Post’s correspondence problem.
Post’s correspondence problem

$S$: set of **domino tiles**.

$\begin{array}{c}
\text{abb} \\
\text{bc}
\end{array}$: domino piece a string at the top and a string at the bottom.

Example:

$$S = \left\{ \begin{array}{c}
\begin{array}{c}
\text{b} \\
\text{ca}
\end{array},
\begin{array}{c}
\text{a} \\
\text{ab}
\end{array},
\begin{array}{c}
\text{ca} \\
\text{a}
\end{array},
\begin{array}{c}
\text{abc} \\
\text{c}
\end{array}\end{array} \right\}.$$
Matching dominoes

\[ S = \{ \begin{array}{c|c|c|c} b & a & ca \\ \hline ca & ab & a \\ \hline abc & c & \end{array} \} . \]

**Match** for \( S \): ordered list of dominoes from \( S \), such that top strings make same string as bottom strings. Example:

\[
\begin{array}{cccccc}
a & b & ca & a & abc \\
ab & ca & a & ab & c \\
\end{array}
\]

(1) Can use same domino more than once.
(2) Do not have to use all pieces of \( S \).
Matching dominoes

\[ S = \{ \begin{array}{ccc} b & a & \text{abc} \\ ca & ab & c \end{array} \} \].

**match** for \( S \): ordered list of dominoes from \( S \), such that top strings make same string as bottom strings. Example:

\[
\begin{array}{cccc}
  a & b & ca & a \\
  ab & ca & a & ab \\
\end{array}
\].

(1) Can use same domino more than once.
(2) Do not have to use all pieces of \( S \).
Matching dominoes

\[ S = \{ \begin{array}{c} b \\ ca \end{array}, \begin{array}{c} a \\ ab \end{array}, \begin{array}{c} ca \\ a \end{array}, \begin{array}{c} abc \\ c \end{array} \} . \]

**match** for \( S \): ordered list of dominoes from \( S \), such that top strings make same string as bottom strings. Example:

\[
\begin{array}{cccc}
  a & b & ca & abc \\
  ab & ca & a & c \\
\end{array}
\]

(1) Can use same domino more than once.
(2) Do not have to use all pieces of \( S \).
Post's Correspondence Problem

Post’s Correspondence Problem (PCP) is deciding whether a set of dominos has a match or not.

modified Post’s Correspondence Problem (MPCP): PCP + a special tile.
Matches for MPCP have to start with the special tile.

Theorem

The MPCP problem is undecidable.
THE END

... (for now)