The language of a decision problem is the set of strings encoding valid problem instances with the answer "Yes".

An algorithm for a decision problem $L$ takes in a string $x$ and outputs:

- Yes if $x \in L$
- No if $x \notin L$

$L \in \mathcal{P} \iff \exists$ a polynomial-time algorithm (Turing machine) $M$ such that for all strings $x$, $x \in L \iff M(x) = \text{Yes}$.
$L \in NP \iff \exists$ a polynomial-time algorithm $M$ an a polynomial $p$ such that for all strings $x$, $x \in L \iff \exists w$ with $|w| \leq p(|x|)$ and $M(x,w) = \text{Yes}$.

What would the "language" of a search problem be?

$(x,y)$

Instead of languages, we represent search problems as binary relations.
A binary relation \( R \) is a set of \((x, y)\) pairs.

An algorithm for a search problem \( R \) takes a string \( x \), and outputs a string \( y \) such that \((x, y) \in R\), if any such \( y \) exists.

What if no such \( y \) exists?

\((x, \text{Fail})\)
\( R \in \text{FP} \iff \exists \text{ a polynomial-time algorithm } M \text{ such that} \\
\forall \text{ strings } x \quad (x, M(x)) \in R \)

\( \text{TFNP} \subseteq \text{FNP} \quad \forall x \quad \exists y \text{ s.t. } M(x, y) = \text{Yes} \)

\( 2\text{P Zero Sum Nash} \in \text{FP} \)

\( \text{Nash} \in \text{TFNP} \)
PPAD is the subset of TFNP where totality is established by a polynomial parity argument on a directed graph.

Theorem: Brouwer is PPAD-complete.

\[ \forall Q \in \text{PPAD} \quad Q \leq_p \text{Brouwer} \]
[+ \text{Brouwer} \leq \text{PPAD} \]
Theorem: Nash is PPAD-complete.