Definitions and Constraints

- two players
  - first player A(lice)
  - second player B(ob)
- turn based
  - A, B, A, B
- both players have complete information
- no randomness
- positions (states)
  - finite set of positions with one or more starting positions
  - repeating moves (infinite loops) are considered draws
- moves (transition from one position to the next)
  - each position has a set of possible moves/next positions
    - potentially no legal move
  - normal play
    - first player who cannot move loses
  - every game ends after a finite number of moves
    - e.g. chess prevents the same move 3 times in a row
    - some exceptions exist
- might be asymmetric
  - e.g. Fuchs und Henne
  - [[Examples of Combinatorial Games]]

First-Player and Second-Player Win games

- some games favor the first (starting) or second player
- one player may have a major advantage due to the starting position or being able to move first or second
- therefore this player always wins
  - assuming both players play optimally

Levels of Game Solutions

- ultra-weakly solved
  - known who wins but not how
- weakly solved
  - strategy is known
  - must be followed from the very start on
- strongly solved

    - known from any valid state
- ultra-strongly solved
    - know for any move during any game state whether it wins/loses/draws
    - also know in how many half-moves

Game-Tree vs State-Space Complexity

-
-

Storing Game States

- needs to be efficient and complete
- move generator
    - creates successors of game states
- identify final states
    - win
    - lose
    - draw
- equivalent game states
    - allow transitions to the same successor state
    - must not be perfectly identical
        - reflections
        - rotations
        - inversion
        - color-change
    - fingerprint/canonical state
        - store only one of the equivalent states
        - ???

Processing Game States

-
- state code
    - non-negative integer
    -
    - draw does not contain the number of half moves
        - due to circles/infinite loops
        - exceptions exist such as Connect 4
    - determine action based on code
        -

- compute codes
    *