Seminar

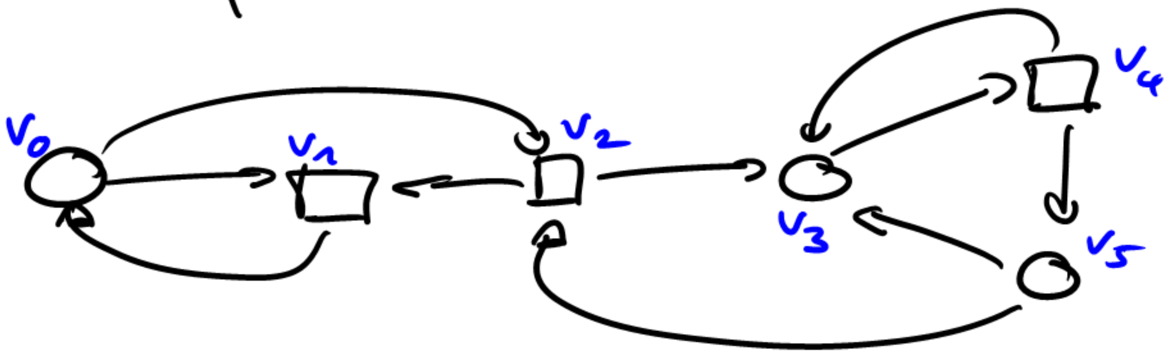
„Never-ending games"

Games in Verification and Logics

Games in Verification & Logics

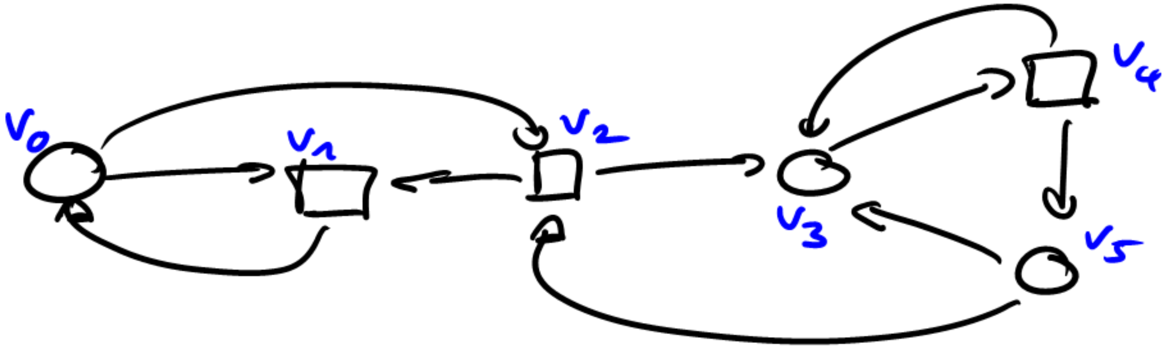
- Graph $G = (V, E)$

Standard assumption: No dead ends

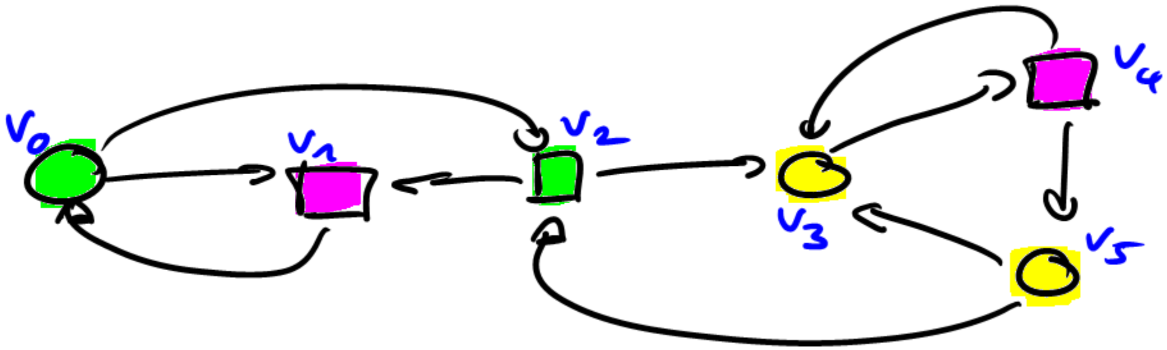


- Two players: Player ○ aka Even aka 0 aka...
  Player □ aka Odd aka 1 aka...
- Play: Infinite path e.g. $\pi = (v_0 v_2 v_3 v_4 v_3 v_4 v_5 v_2 v_1)^\omega$
- Winning condition for Player ○: $Win \subseteq V^\omega$
  Player ○ wins a play π iff $\pi \in Win$
  otherwise Player □ wins π
  "Zero-sum game"

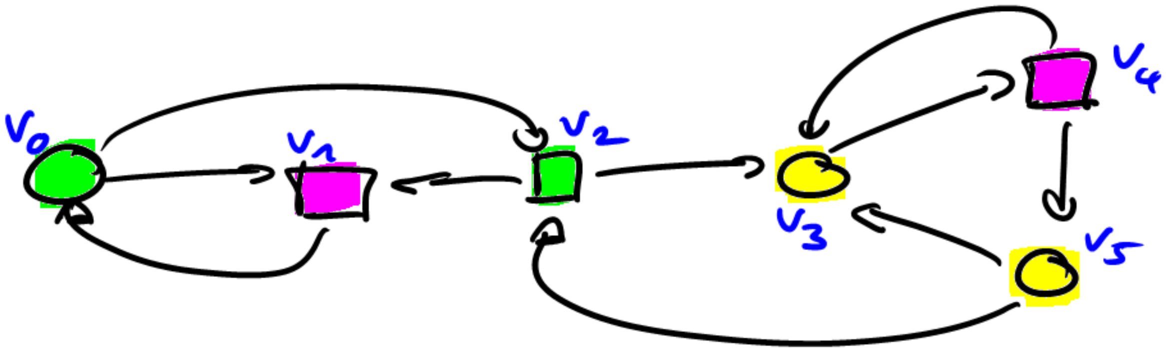
Games in Verification & Logics



Typical winning conditions:
defined w.r.t. a node coloring



Typical winning conditions

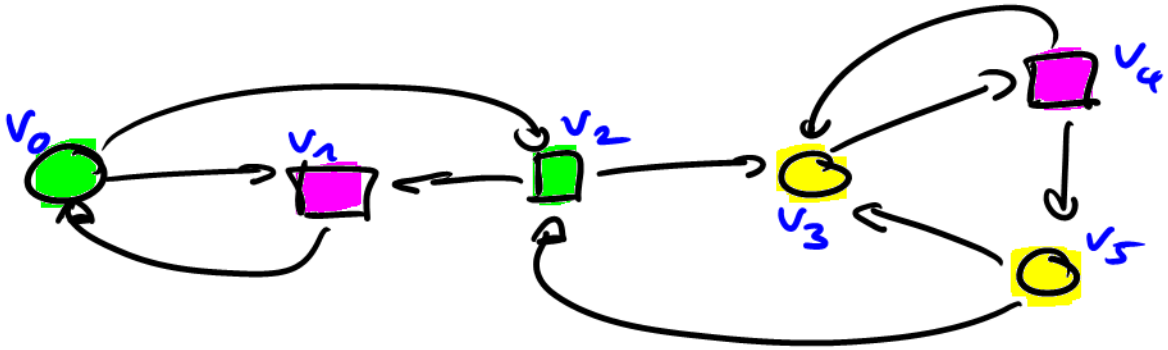


- Reachability condition:
  Player ○ wins a play if it visits a  node at least once.

- Safety condition:
  Player ○ wins a play if it never visits a  node

- Büchi condition
  Player ○ wins a play if it visits a node infinitely often

- More involved: Muller, Rabin, Streett, Parity, ...



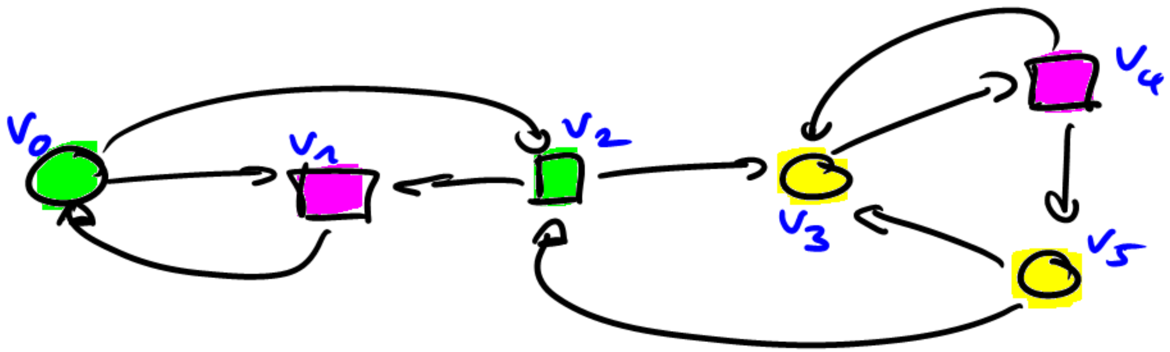
- <u>In general</u>:

Not interesting "Does Player ○ win π?"

Interested in: "Has Player ○ a strategy to win the node v?"

Strategy: "A program" which reads the moves by the opponent and tells Player ○ how to move.

Strategy wins node v (for player ○): All resulting plays are won by Player ○.



Example: Win = "Player 0 reach [green] "

- Strategy to win v_0: trivial
- Strategy to win v_1: trivial
- Strategy to win v_5: "Play at v_5 play to v_2"
- Strategy to win v_3: ?

Where do these games arise?

- Model checking (not Heodi !!!)

  <u>Given:</u> Program & formal specification

  LTL, CTL, μ-calculus

  <u>Goal:</u> Decide if program satisfies specification

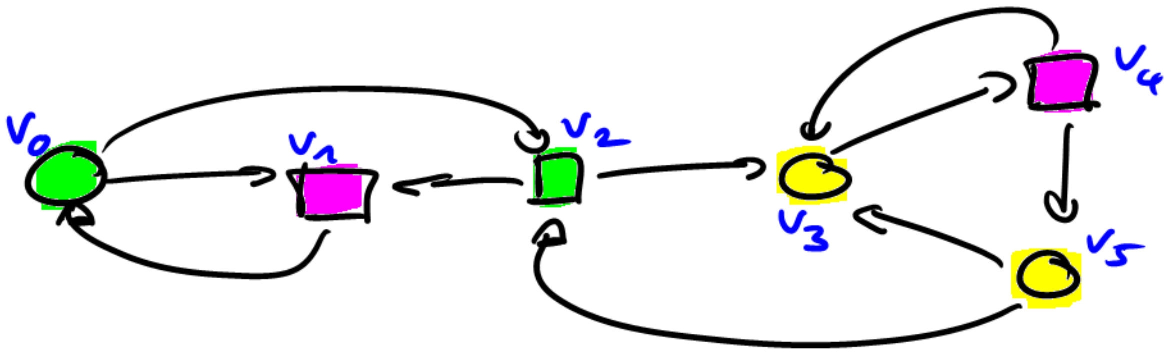
- <u>Synthesis:</u>

  <u>Given:</u> A controllable program & formal specification

  <u>Goal:</u> Construct a controller for the program so that it always satisfies the specification (even in case of an adversarial user.)

Controller synthesis: simplified

- Programs ≙ succinct description of possibly infinite graphs



- Player ○ : controller
- Player □ : possibly malevolent environment (user)

<u>Goal</u>: Find strategy (≙ "synthesize controller") s.t. every request is eventually processed

▷ If program starts in v_3 : ?
if program starts in v_0 : ?

Topics:

- Memoryless determinacy of parity games and Zielonka's algorithm.

  Literature:
  - Wieslaw Zielonka: Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees
  - Grädel, Thomas, Wilke: Automata, Logics, and Infinite Games (Chapter 6)
  - Perrin and Pin: Infinite Words (Chapter IV.4)

- Small-progress measures for solving parity games

  Literature:
  - Marcin Jurdzinski: Small Progress Measures for Solving Parity Games
  - Grädel, Thomas, Wilke: Automata, Logics, and Infinite Games (Chapter 7)

- Mean-payoff games: Relation to parity games; how to solve them using.

  Literature:
  - Henrik Björklund, Sven Sandberg, Sergei G. Vorobyov: A Combinatorial Strongly Subexponential Strategy Improvement Algorithm for Mean Payoff Games

- Concurrent reachability games

  Literature:
  - Luca de Alfaro, Thomas A. Henzinger, Orna Kupferman: Concurrent Reachability Games.

- Simple stochastic games and Markov decision processes: Applications and how to solve them.

  Literature:
  - Anne Condon: On Algorithms for Simple Stochastic Games
  - Anne Condon: The Complexity of Stochastic Games
  - PRISM

- Rabin games and LTL synthesis

  Literature:
  - Nir Piterman, Amir Pnueli: Faster Solutions of Rabin and Streett Games
  - Lily

- Muller games

  Literature:
  - John Fearnley, Martin Zimmermann: Playing Muller Games in a Hurry

- Strategy synthesis for parity games with imperfect information

  Literature:
  - Dietmar Berwanger, Krishnendu Chatterjee, Martin De Wulf, Laurent Doyen, Thomas A. Henzinger: Strategy construction for parity games with imperfect information

- Solving games of imperfect information

  Literatute:
  - Martin De Wulf, Lauren Doyen, Jean-Francois Raskin: A Lattice Theory for Solving Games of Imperfect Information

Preliminary organization:

- First meeting in the first or second week of the summer term.
  - Talk on infinite games in general: Definitions & Relations
    either Jan, Maximilian, or me
  - "How a seminar talk should (not) be."
- Student talks somewhen in second half of the semester
  - possibly 2 talks per date, several dates per week
- ~40 min per talk

Preliminary organization

Time schedule:

4 weeks before the talk:

- first meeting with the supervisor
- must have read the paper
- fix the content   content

2 weeks before the talk:

- preliminary version of the slides

1 week before the talk:

- final version of the slides
- preliminary version of the written summary (4–6 pages)

2 weeks after the talk:

- final version of the summary