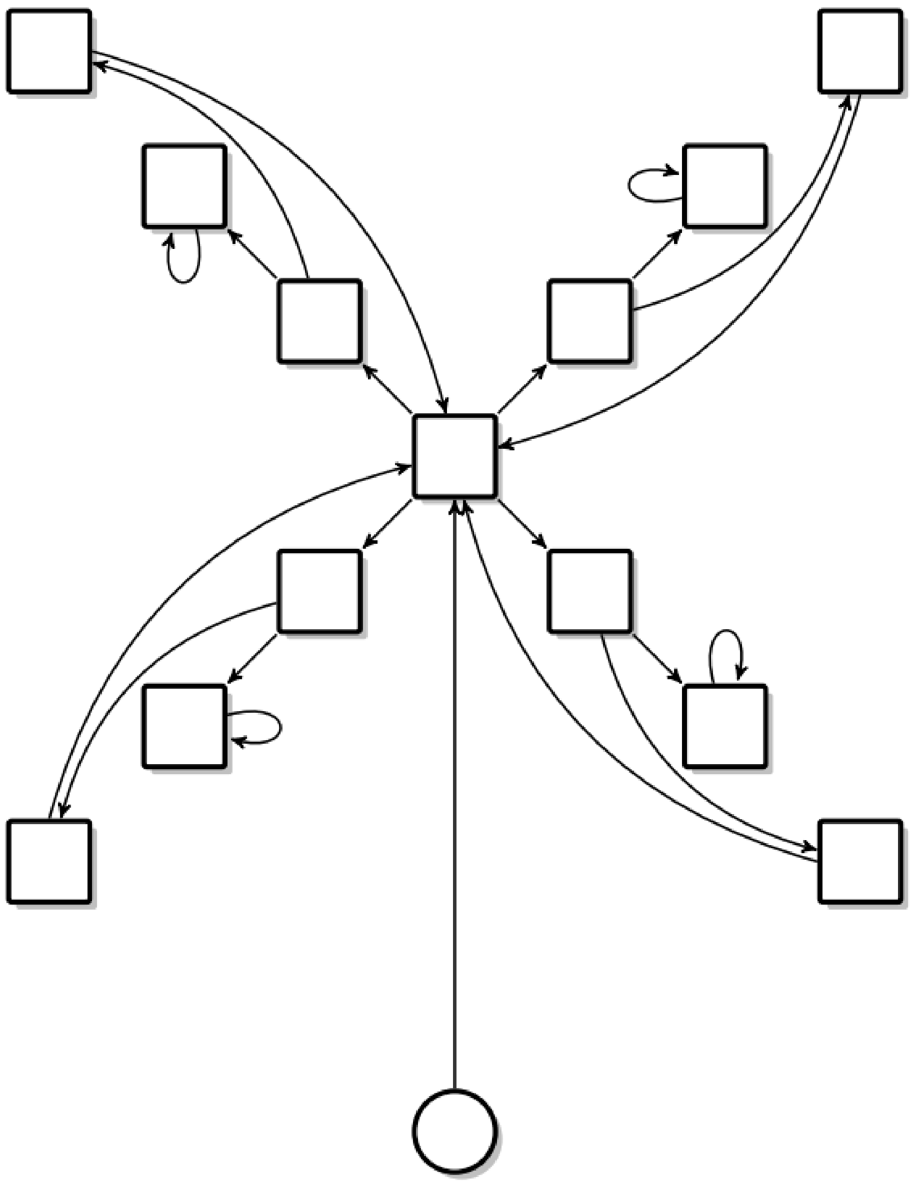


INFINITE GAMES

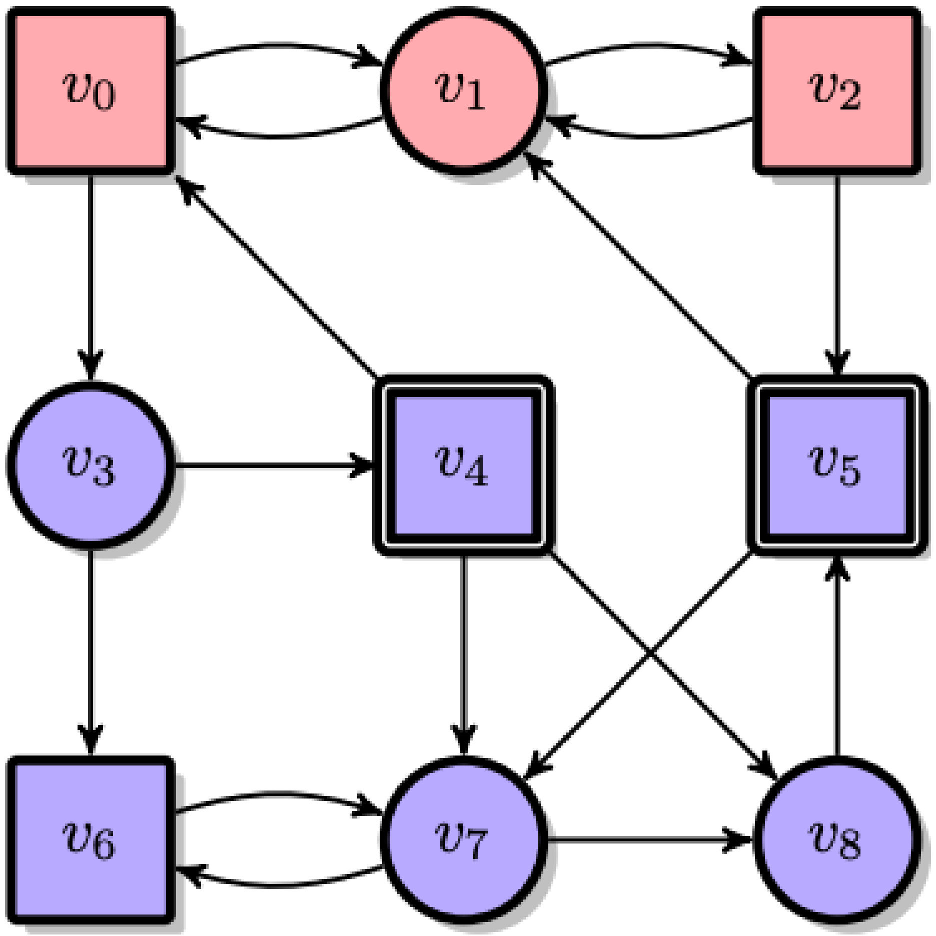
Aman Raj (170101006)
Mayank Wadhwani (170101038)

Motivation

- Programs for non-terminating systems
- e.g. controller regulating the anti-lock brake in a car, controller to regulate temperature of a system
- Lead to emergence of so-called reactive systems
- Interaction between the system and its environment
- We model them by a finite graph and find winning strategies

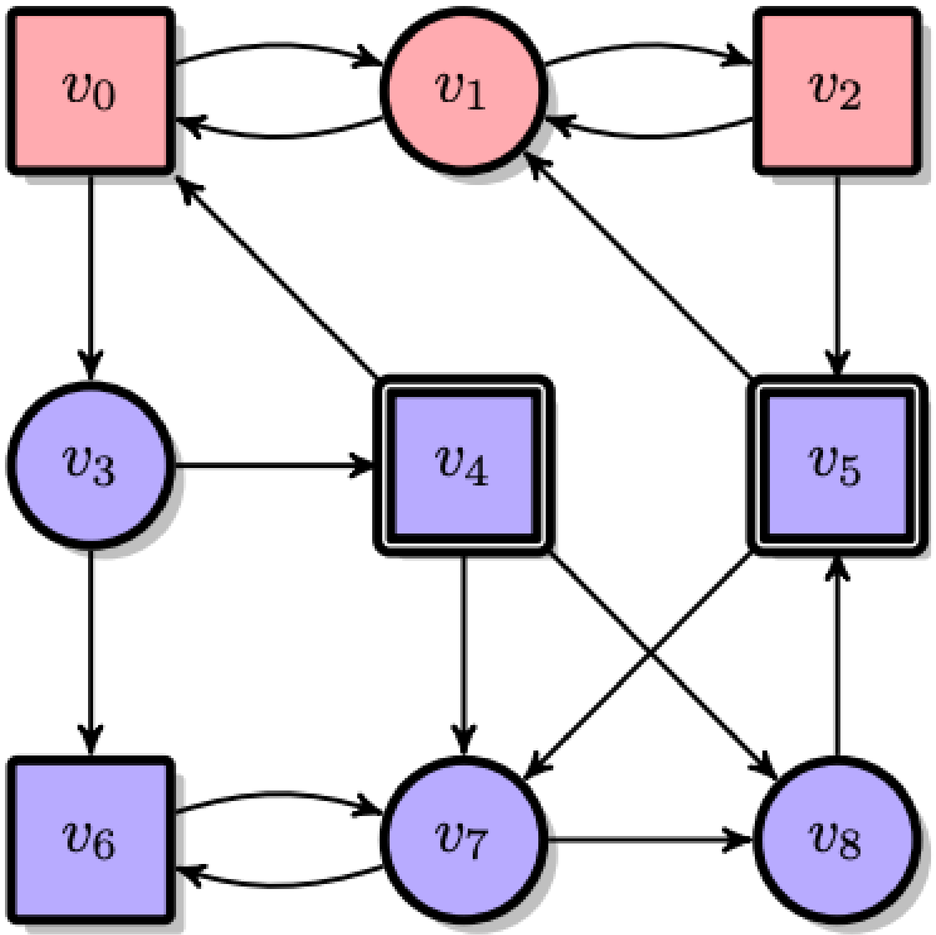
Essential Terminologies

- Arena
- Play
- Strategy
- Positional Strategy
- Determinacy



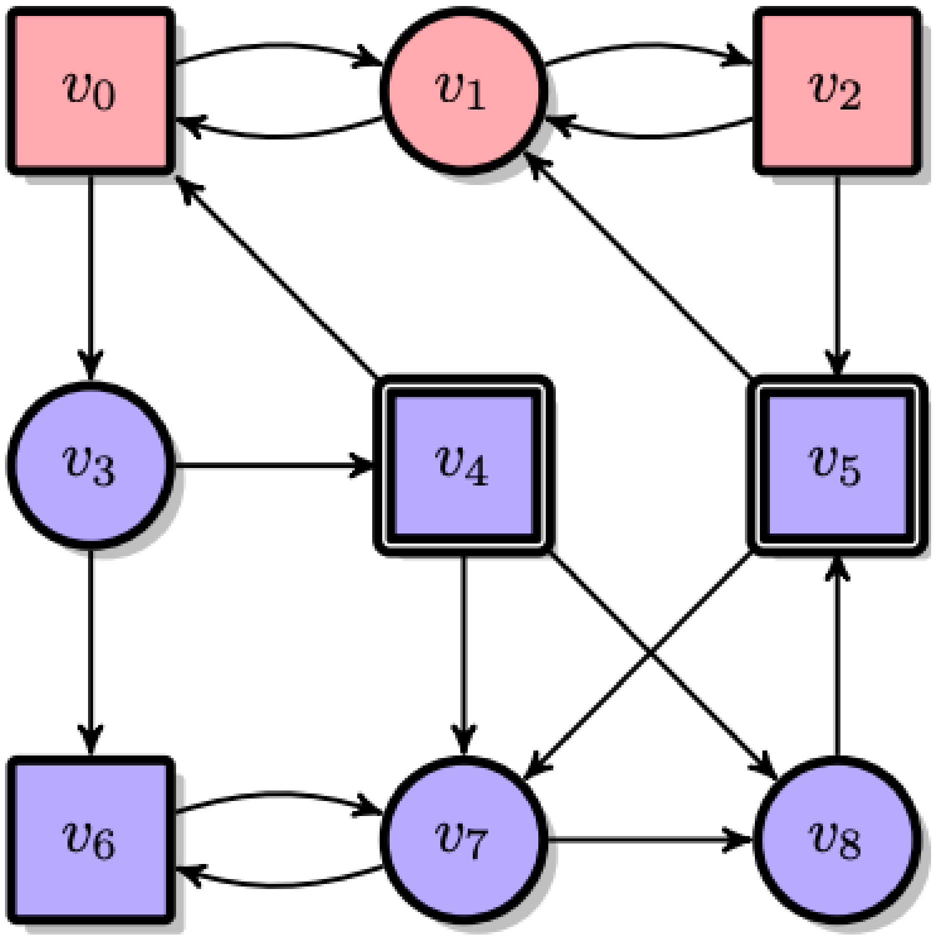
Essential Terminologies

- **Arena: A = (V , V0 , V1 , E)**
- Play
- Strategy
- Positional Strategy
- Determinacy



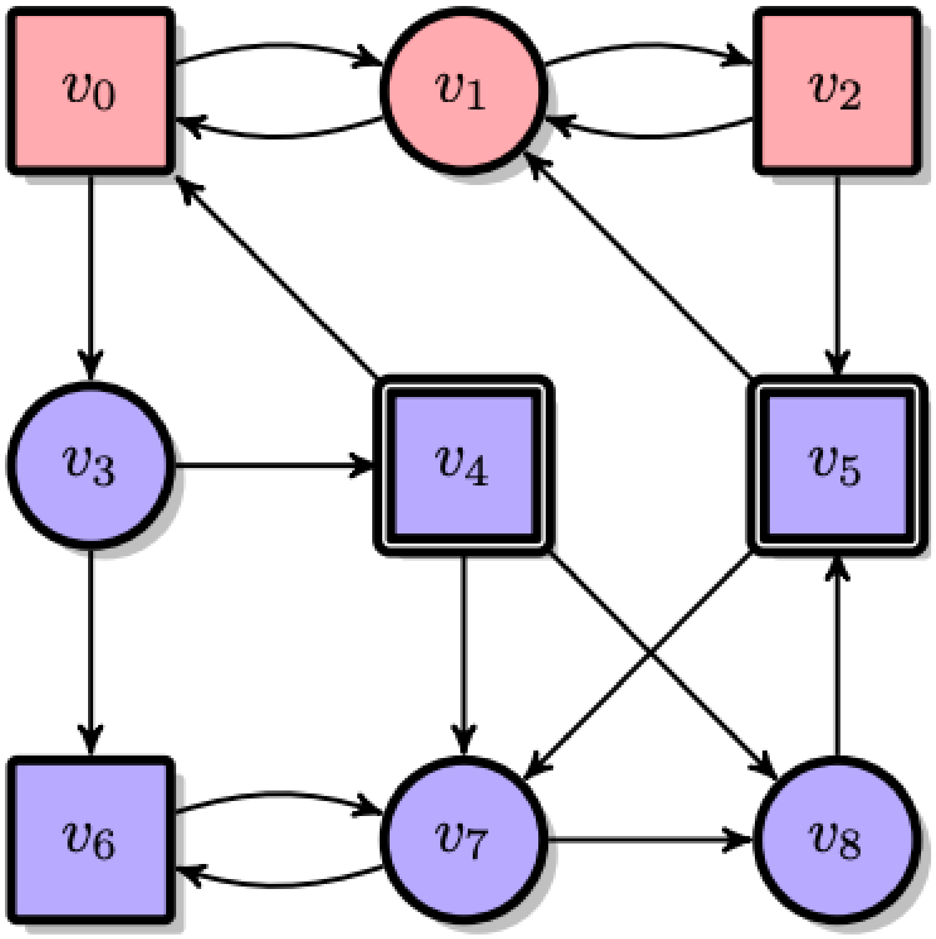
Essential Terminologies

- Arena
- **Play = infinite sequence**
- **Ex. v0, v1, v2, v1 , v0 , v1 , …….**
- Strategy
- Positional Strategy
- Determinacy



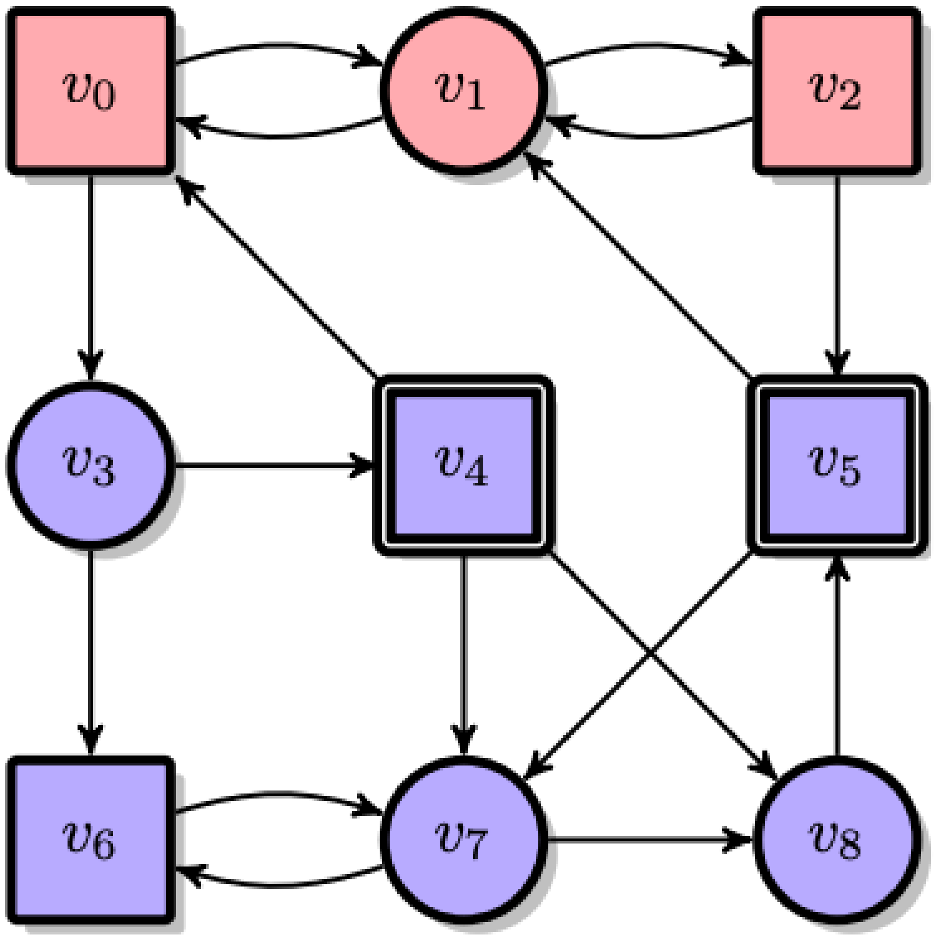
Essential Terminologies

- Arena
- Play
- **Strategy: function V*Vi -> V**
- Positional Strategy
- Determinacy



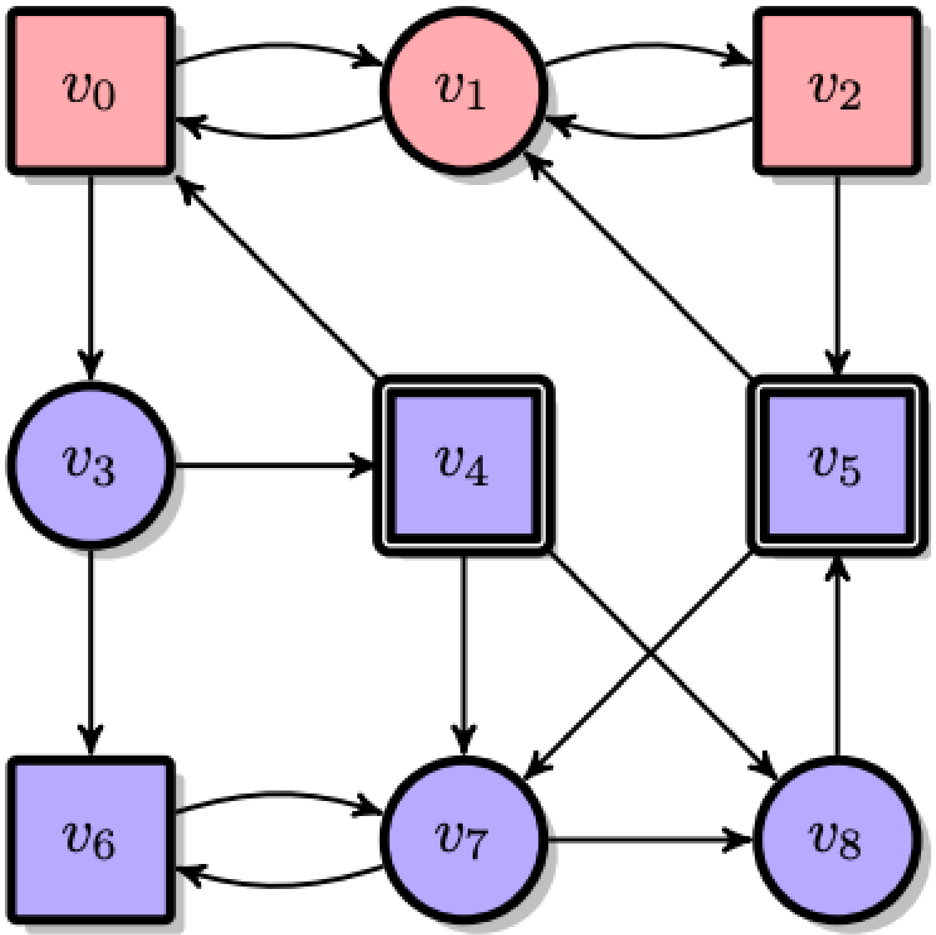
Essential Terminologies

- Arena
- Play
- Strategy
- **Positional Strategy: strategy depends only on current state (no history component)**
- **For red: f(vo) = v1, f(v2) = v1**
- Determinacy



Essential Terminologies

- Arena
- Play
- Strategy
- Positional Strategy
- **Determinacy**
- **W0(V) union W1(V) = V**



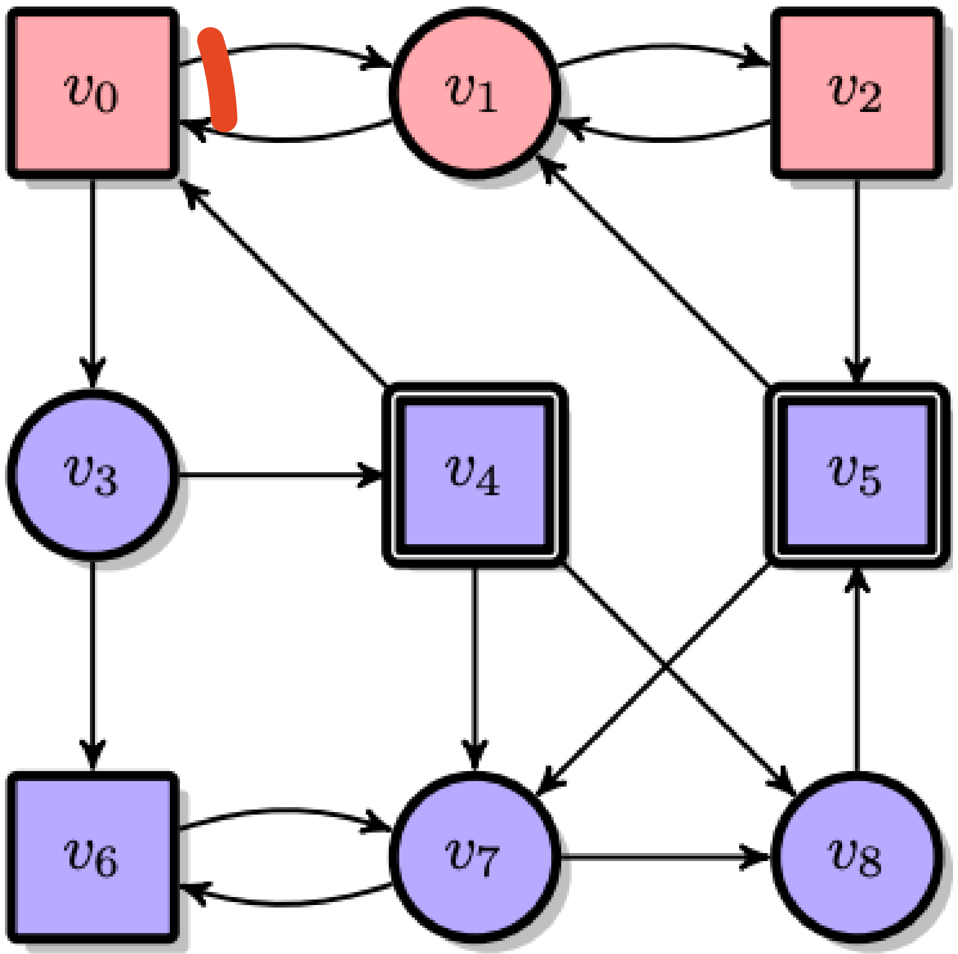
Examples of Games with Uniform Positional Strategy

- Reachability games
- Büchi games
- Parity games

Reachability Games

Intro

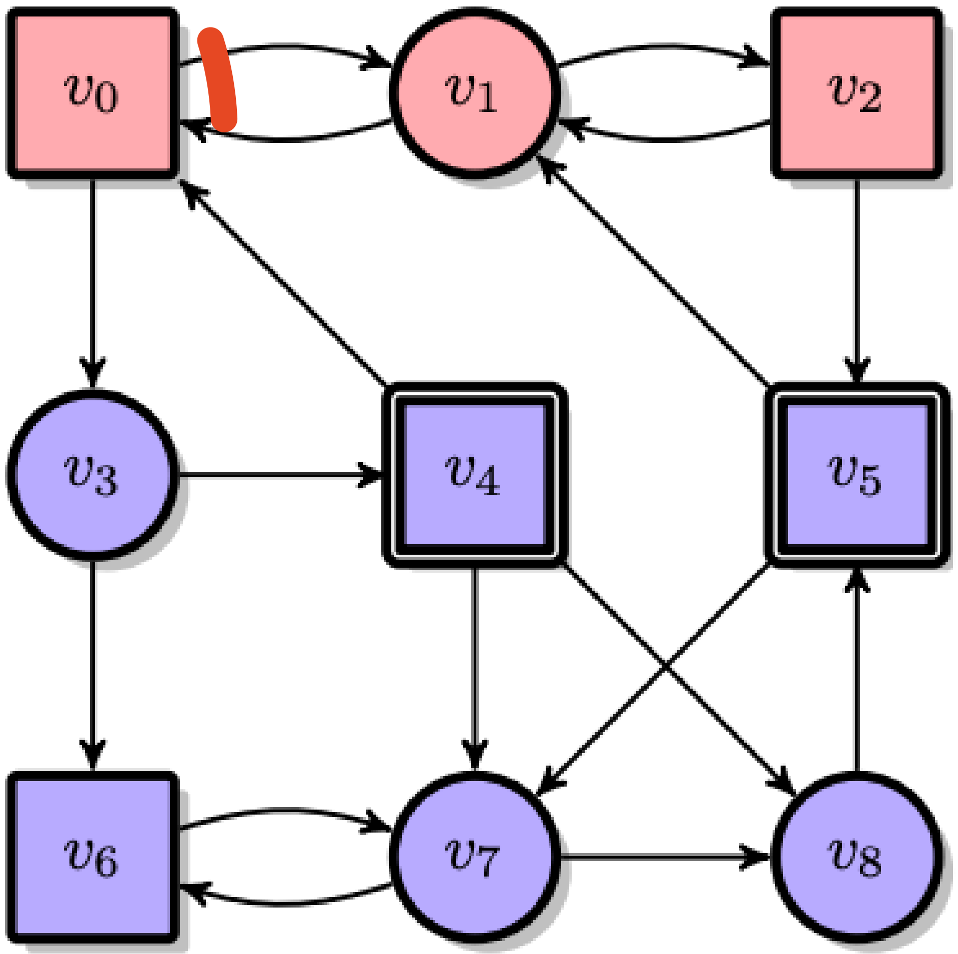
- Given, $R \subseteq V$ be a subset of A's vertices, here $R = \{v_4, v_5\}$
- Player 0's goal is to reach R at least once
- Player 1 tries to avoid reaching R
- $\mathrm{Reach}(R) = \{v_0, v_1, \ldots, \in V^w \mid \exists\, i: v_i \in R\}$



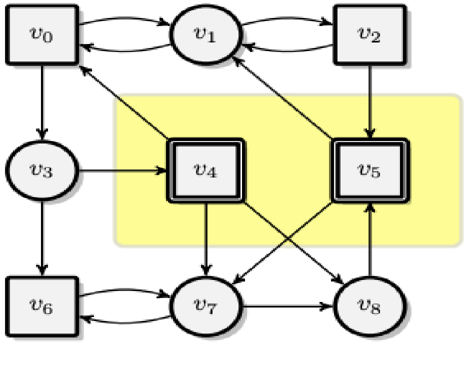
Reachability Games (Cont.)

Attractor

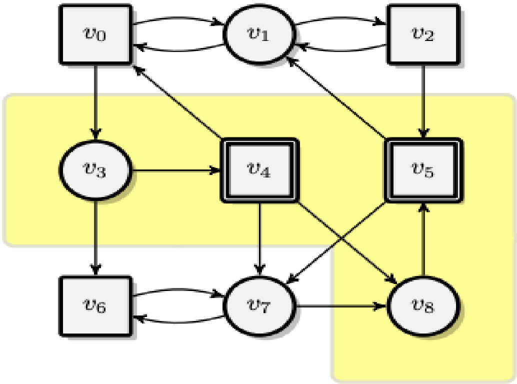
- $Attr_0(R)$ or 0-attractor of R :
  All vertices from where Player 0 can attract the token to R
- Construction of the attractor is hierarchical
- Vertex v is added to attractor if:
  $\{v \in V_i \mid \text{some successor of } v \in R\}$
  or
  $\{v \in V_{1-i} \mid \text{all successors of } v \in R\}$



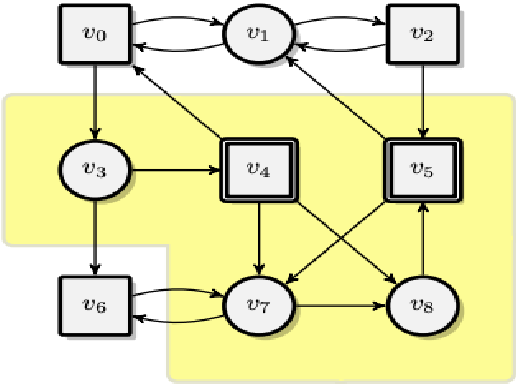
Reachability Games (Cont.)



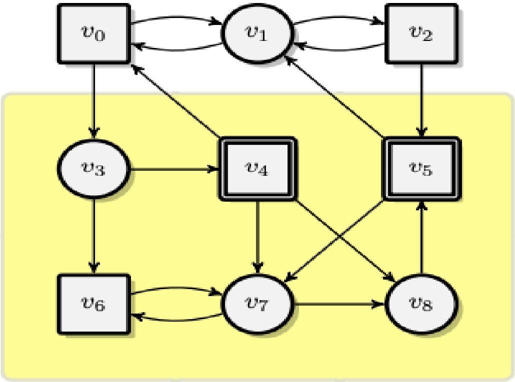
$\mathrm{Attr}_0^0(\{v_4, v_5\}) =$
$\{v_4, v_5\}$



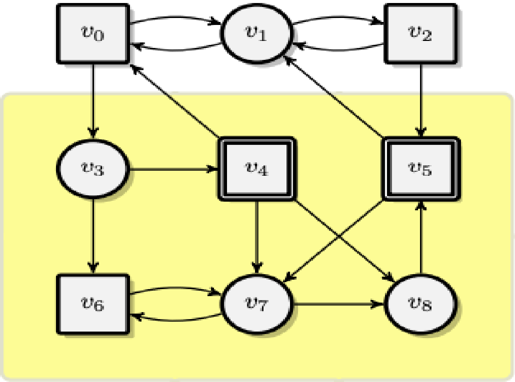
$\mathrm{Attr}_0^1(\{v_4, v_5\}) =$
$\{v_4, v_5\} \cup \{v_3, v_8\}$



$\mathrm{Attr}_0^2(\{v_4, v_5\}) =$
$\{v_3, v_4, v_5, v_8\} \cup \{v_3, v_7, v_8\}$



$\mathrm{Attr}_0^3(\{v_4, v_5\}) =$
$\{v_3, v_4, v_5, v_7, v_8\} \cup \{v_3, v_6, v_7, v_8\}$

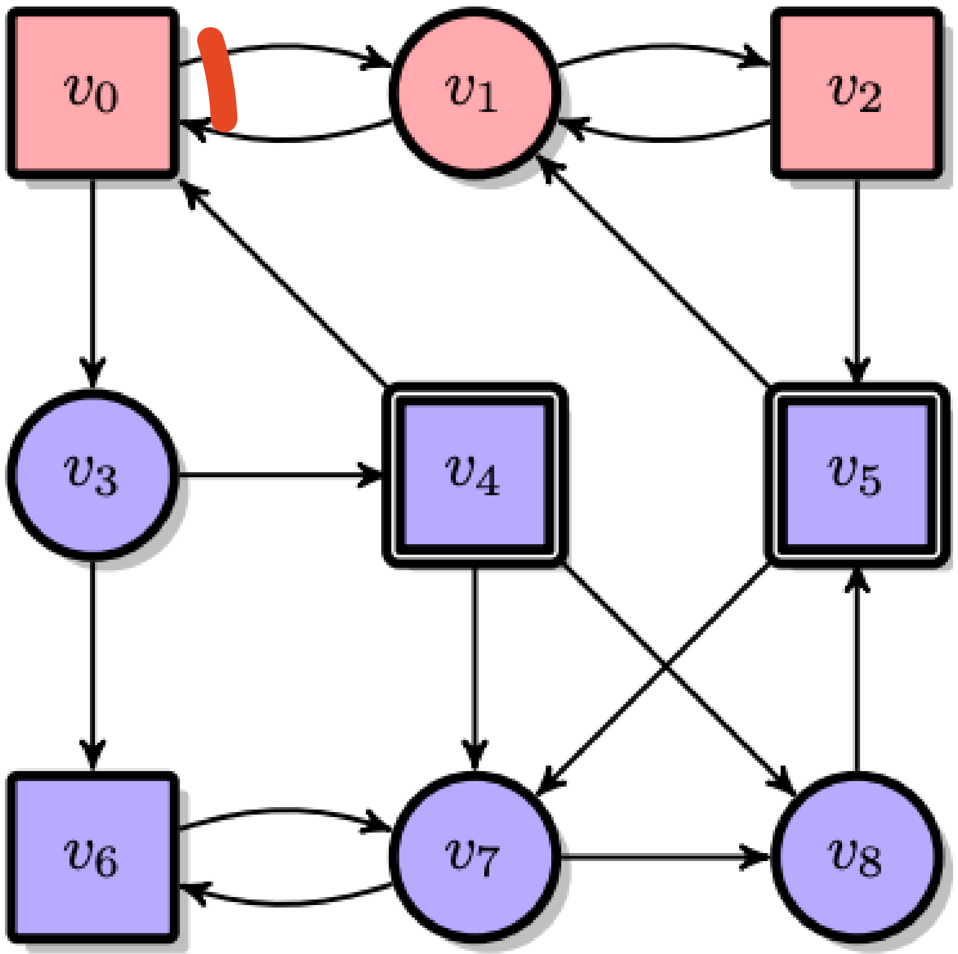


$\mathrm{Attr}_0^4(\{v_4, v_5\}) =$
$\{v_3, v_4, v_5, v_6, v_7, v_8\}$

Reachability Games (Cont.)

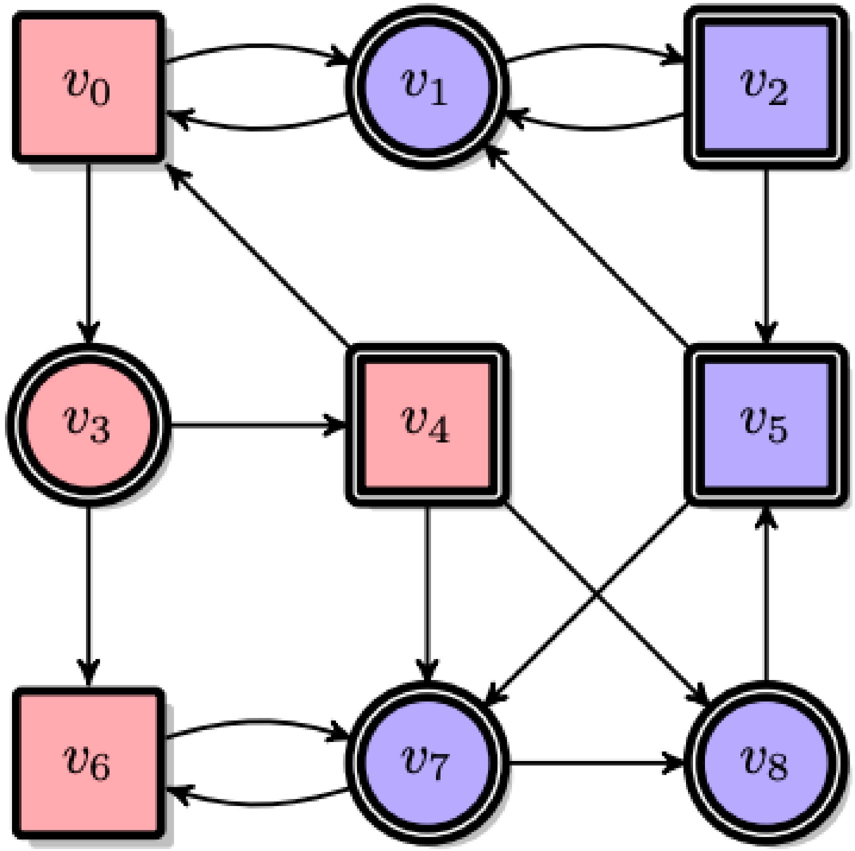
Solution

- $W_0(G) = Attr_0(R)$ and
- $W_1(G) = V \setminus Attr_0(R)$
- Uniform positional winning strategies for both players
- Player 0 can enforce a win from every vertex in $W_0(G)$
- Player 1 can enforce a win by keeping the token in $W_1(G)$



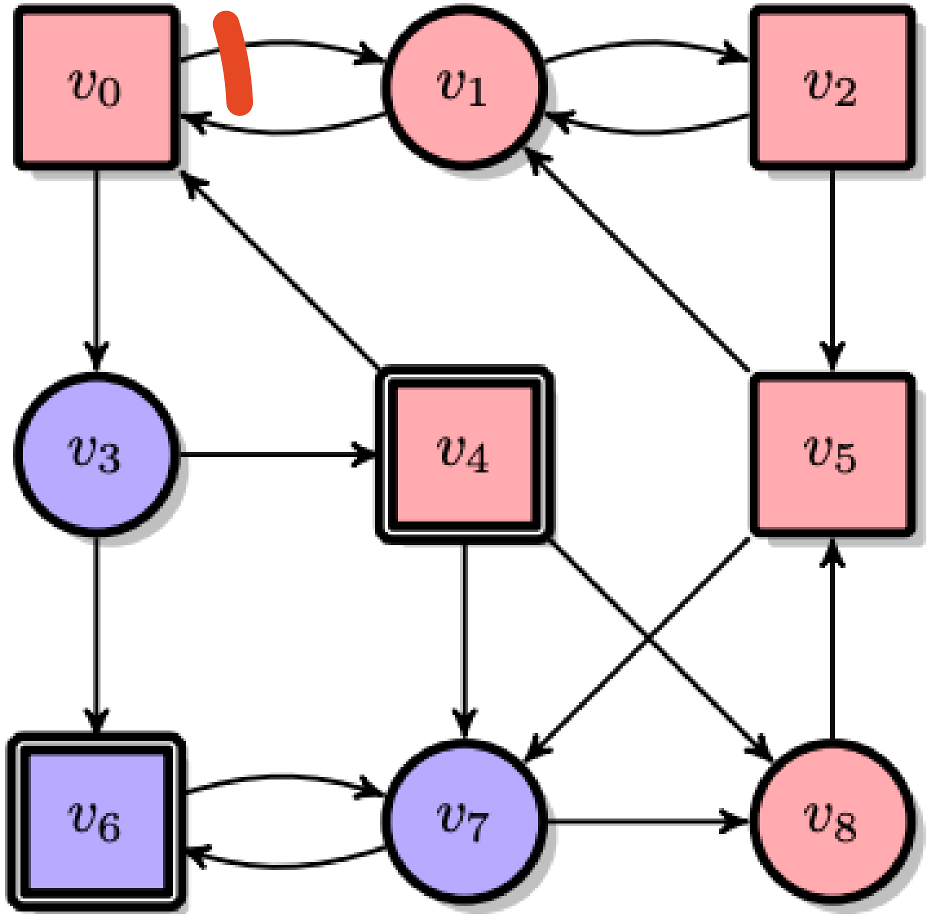
Safety Games

- Dual of reachability games
- Player 0 is not allowed to leave a specific region S of safe vertices.
- While, Player 1's goal is to reach V \ S.
- Safe(S) = $\{v_0, v_1, \ldots, \in V^w \mid \forall\, i\colon v_i \in S\}$
- We can turn a safety game into a reachability game



Büchi Games

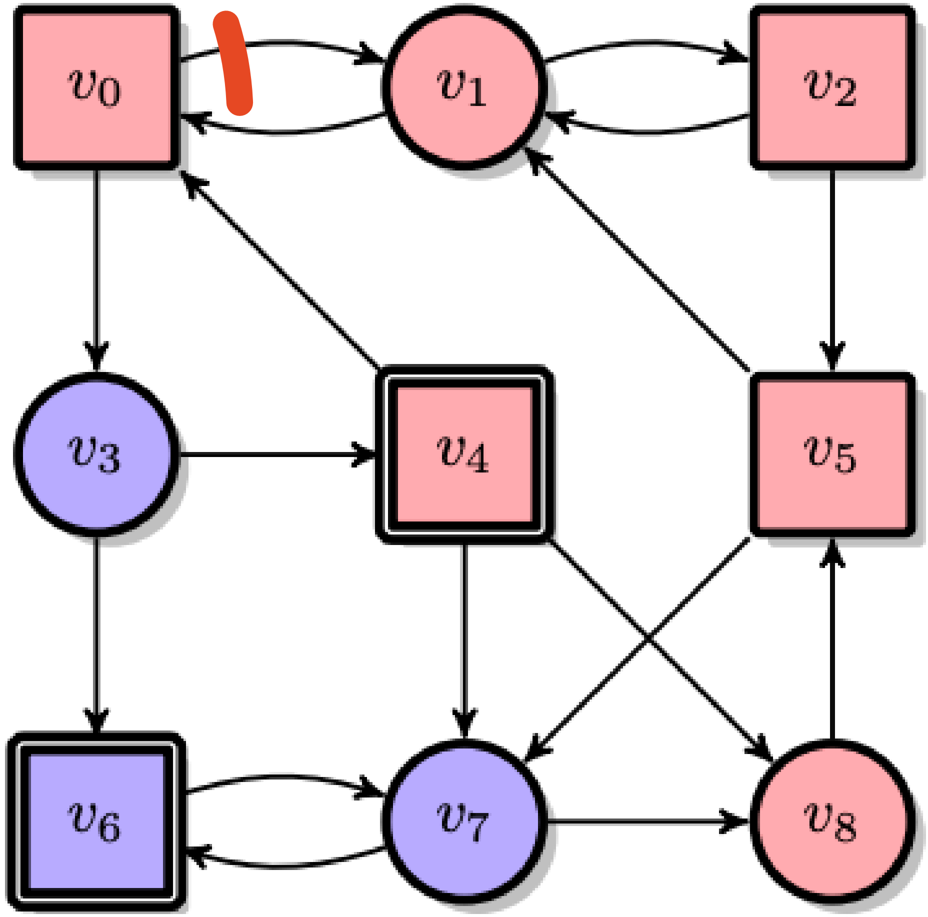
- So far: Visit even once, done. Future irrelevant
- Büchi games-> there must be some element in a given set F that should be visited infinite times.



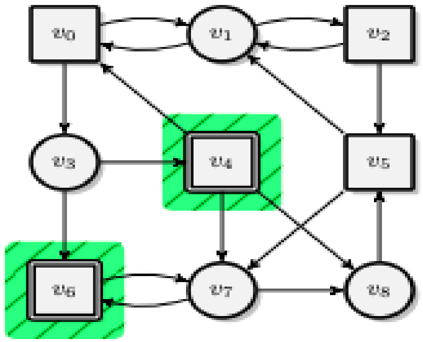
Büchi Games

- So far: Visit even once, done. Future irrelevant
- Büchi games-> there must be some element in a given set F that should be visited infinite times.
- Approach: start with F, and move according to following recurrence

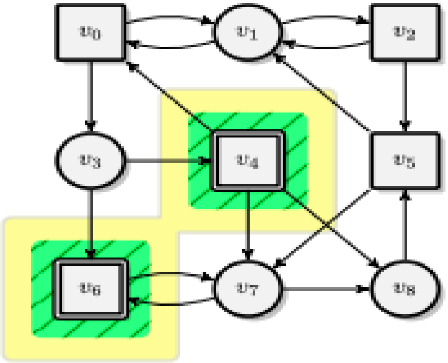
- $F^0 = F$,
- $W_1^n = V \setminus \mathrm{Attr}_0(F^n)$ *for every* $n \geq 0$, *and*
- $F^{n+1} = F \setminus \mathrm{CPre}_1(W_1^n)$ *for every* $n \geq 0$.



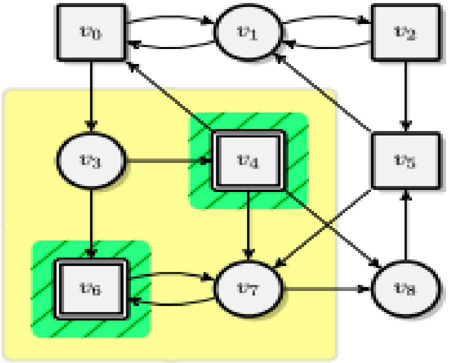
Büchi Games (Example)



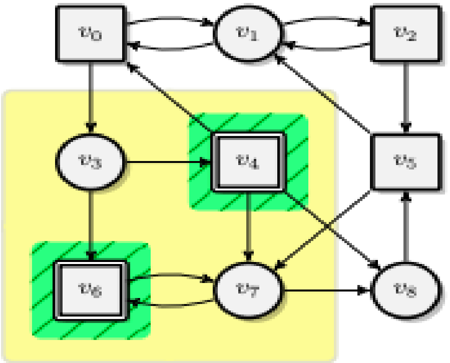
$\{v_4, v_6\}^0 =$
$\{v_4, v_6\}$



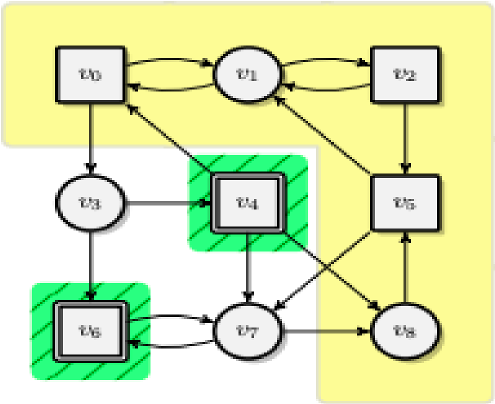
$\mathrm{Attr}_0^0(\{v_4, v_6\}) =$
$\{v_4, v_6\}$



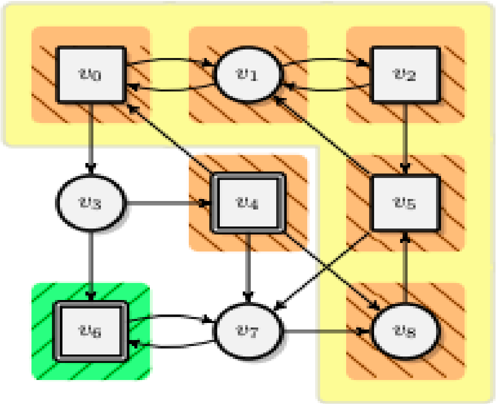
$\mathrm{Attr}_0^1(\{v_4, v_6\}) =$
$\{v_4, v_6\} \cup \{v_3, v_7\}$



$\mathrm{Attr}_0(\{v_4, v_6\}) =$
$\{v_3, v_4, v_6, v_7\}$

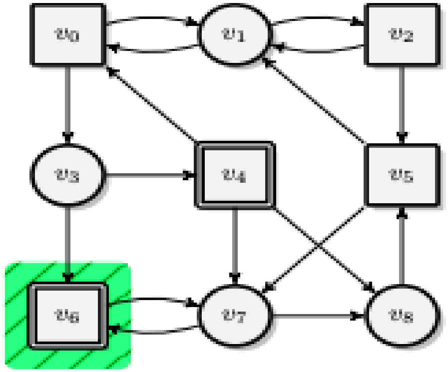


$W_1^0(\{v_4, v_6\}) =$
$\{v_0, v_1, v_2, v_5, v_8\}$

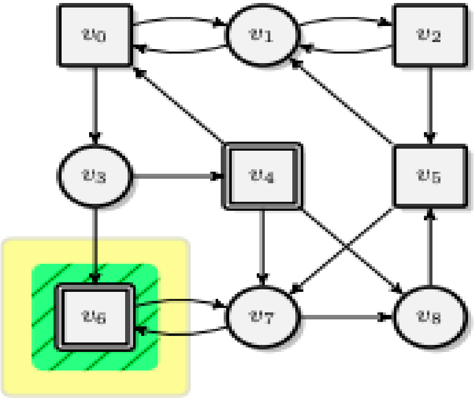


$\mathrm{CPre}_1(\{v_0, v_1, v_2, v_5, v_8\}) =$
$\{v_0, v_1, v_2, v_4, v_5, v_8\}$

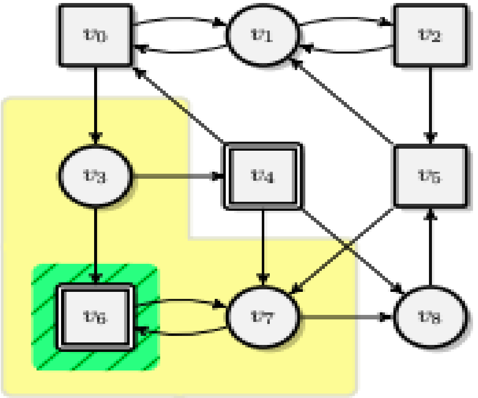
Büchi Games (Example explained)



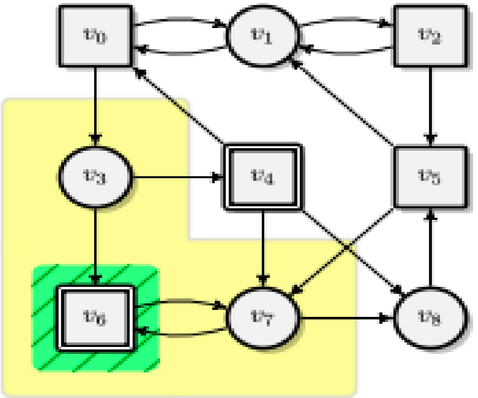
$\{v_4, v_6\}^1 =$
$\{v_6\}$



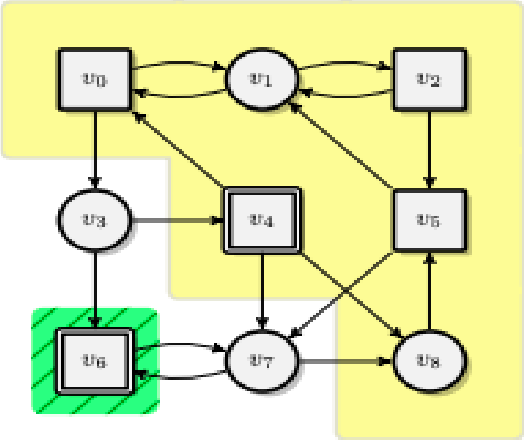
$\mathrm{Attr}_0^0(\{v_6\}) =$
$\{v_6\}$



$\mathrm{Attr}_0^1(\{v_6\}) =$
$\{v_6\} \cup \{v_3, v_7\}$



$\mathrm{Attr}_0(\{v_6\}) =$
$\{v_3, v_6, v_7\}$

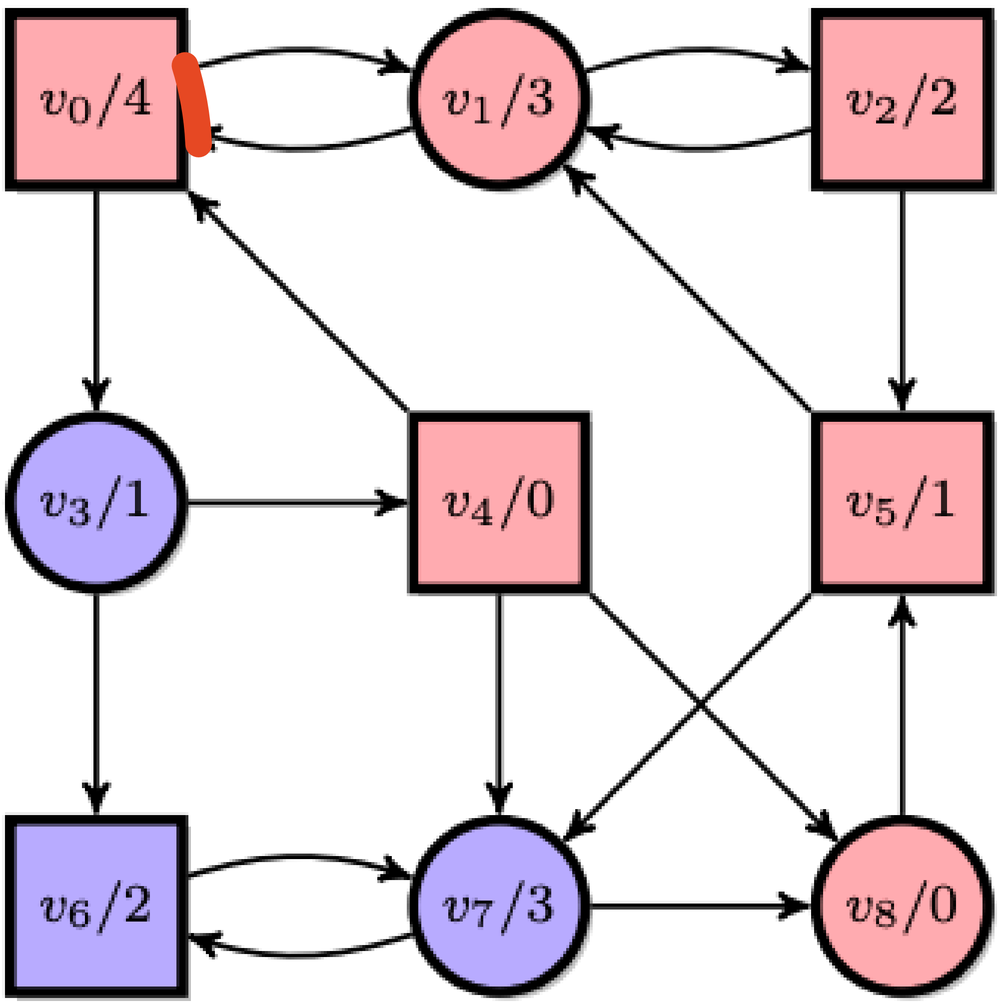


$W_1^1(\{v_4, v_6\}) =$
$\{v_0, v_1, v_2, v_4, v_5, v_8\}$

Parity Games

Intro

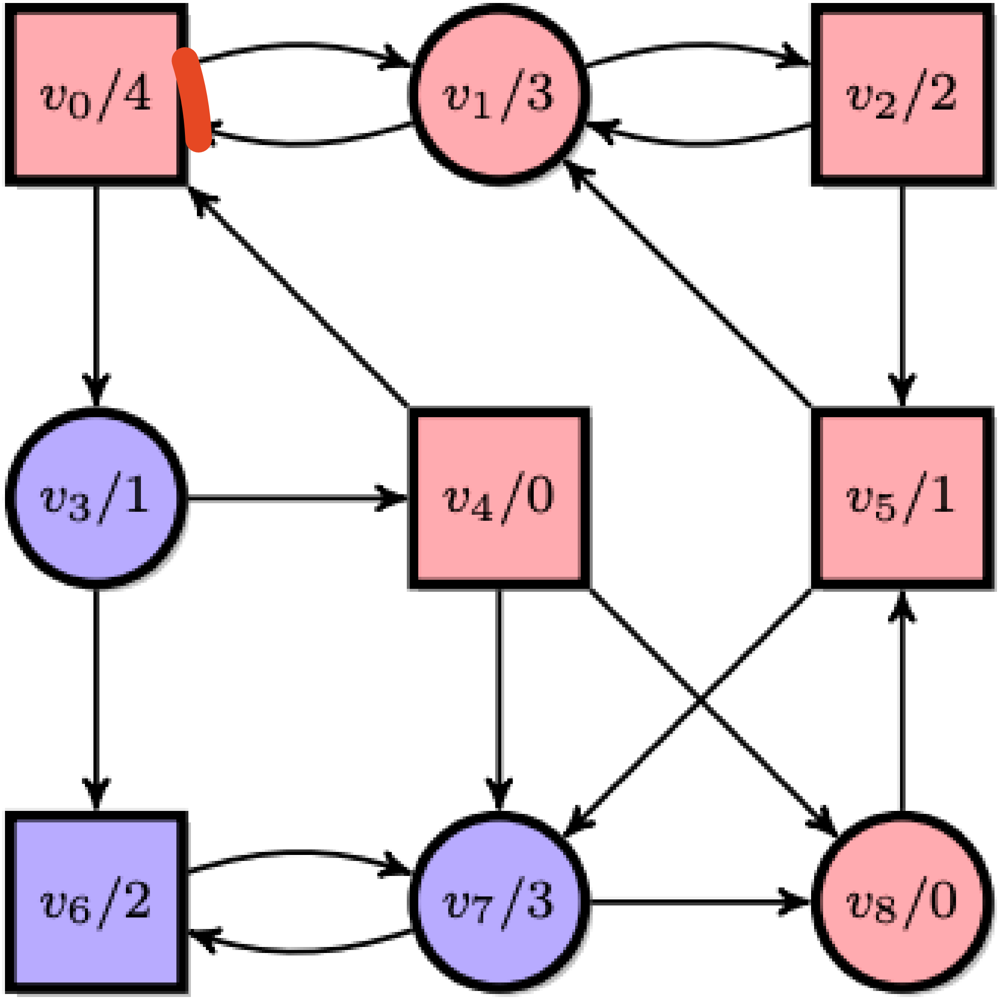
- Generalization of Büchi games
- Vertices of the arena are colored by natural numbers
- Player i wins a play ρ if, and only if, the minimal color seen infinitely often in ρ has parity i.
- $\text{Parity}(\Omega) = \{ \rho \in V^\omega \mid$ $\min \text{Inf}(\Omega(\rho_0)\Omega(\rho_1)\Omega(\rho_2) \cdots)$ is even $\}$



Parity Games

Solution

- Can be solved using recursive algorithm involving computing attractors
- Exponential upper bound on the running time

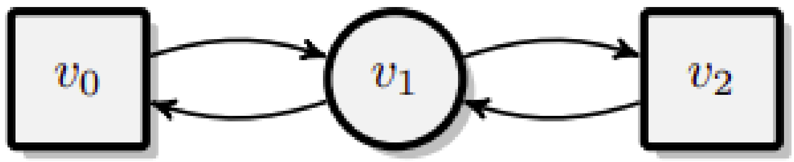


Finite-state Strategies and Reductions

- Finite-state Strategies
- Reductions
- Weak muller games
- Muller games
- Limits on Reductions

Finite-state Strategies

Example:
Let game with F = V



Finite-state Strategies

- Memory Structure

  Let A = (V, V0, V1, E) be an arena.
  A memory structure M = (M, init, upd) for A.

Finite-state Strategies

- Memory Structure

  Let A = (V, V0, V1, E) be an arena.
  A memory structure M = (M, init, upd) for A.

  Init: V -> M

Finite-state Strategies

- Memory Structure

  Let A = (V, V0, V1, E) be an arena.
  A memory structure M = (M, init, upd) for A.

  Init: V -> M

  Upd: M * V -> M

Finite-state Strategies

- Memory Structure

  Let A = (V, V0, V1, E) be an arena.
  A memory structure M = (M, init, upd) for A.

  Init: V -> M

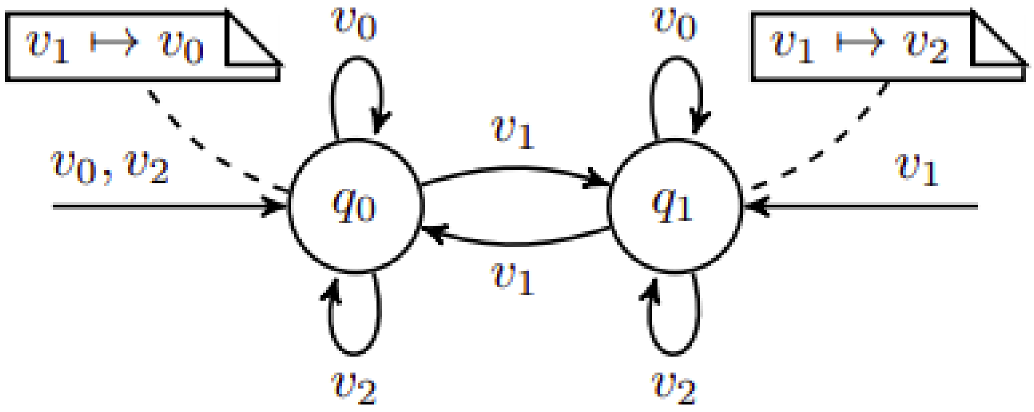
  Upd: M * V -> M

- Next Move function

  nxt: Vi * M -> V satisfying (v, nxt(v,m)) belongs to E

Finite-state Strategies

- If some strategy requires a memory structure and some next move function, it is finite state strategy.
- All early games (that are positional strategy-based games) like Büchi, reachability can also be implemented by finite state strategy



Reductions

- Aim is to obtain a new game G' where Player 0 is known to have a positional winning strategy.
- Reducing G to G' simplifies the winning condition
- But increases the size because of memory structure
- $\rho \in \text{Win}$ if, and only if, $\text{ext}(\rho) \in \text{Win}'$.

Weak Muller Game

- Let $A = (V, V0, V1, E)$ be an arena and let $F \subseteq 2^V$ be a family of subsets of A's vertices. Then, the weak Muller condition wMuller(F) is defined as :

  $$\mathrm{wMuller}(F) := \{\rho \in V^{\omega} \mid \mathrm{Occ}(\rho) \in F\}$$

  We call a game $G = (A, \mathrm{wMuller}(F))$ a weak Muller game.

Weak Muller Game (Reduction)

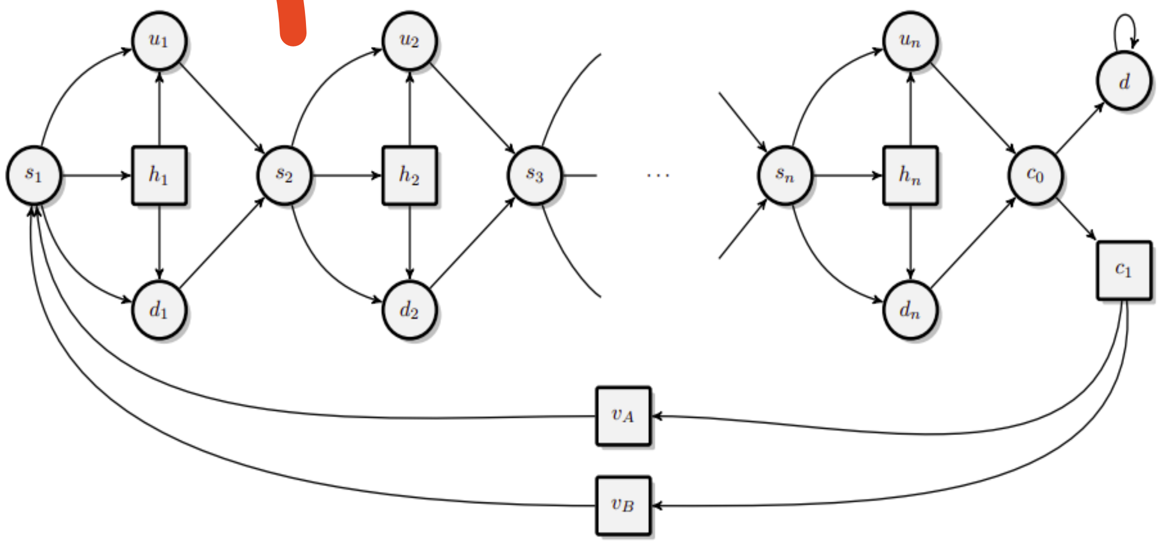
Weak Muller games are reducible to weak max-parity games

- $\mathrm{wMaxParity}(\Omega) := \{\rho \in V^{\omega} \mid \max \mathrm{Occ}(\Omega(\rho 0)\Omega(\rho 1)\Omega(\rho 2)\cdots) \text{ is even}\}$.
- Player i wins a play, if the parity of the maximal color occurring during the play is i
- $\Omega(v, S) = \begin{cases} 2 \cdot |S| & \text{if } S \in F, \\ 2 \cdot |S| - 1 & \text{if } S \text{ does not} \in F. \end{cases}$
- As the set of visited vertices increases until it gets stationary at some point, larger sets are assigned larger colors than smaller sets.

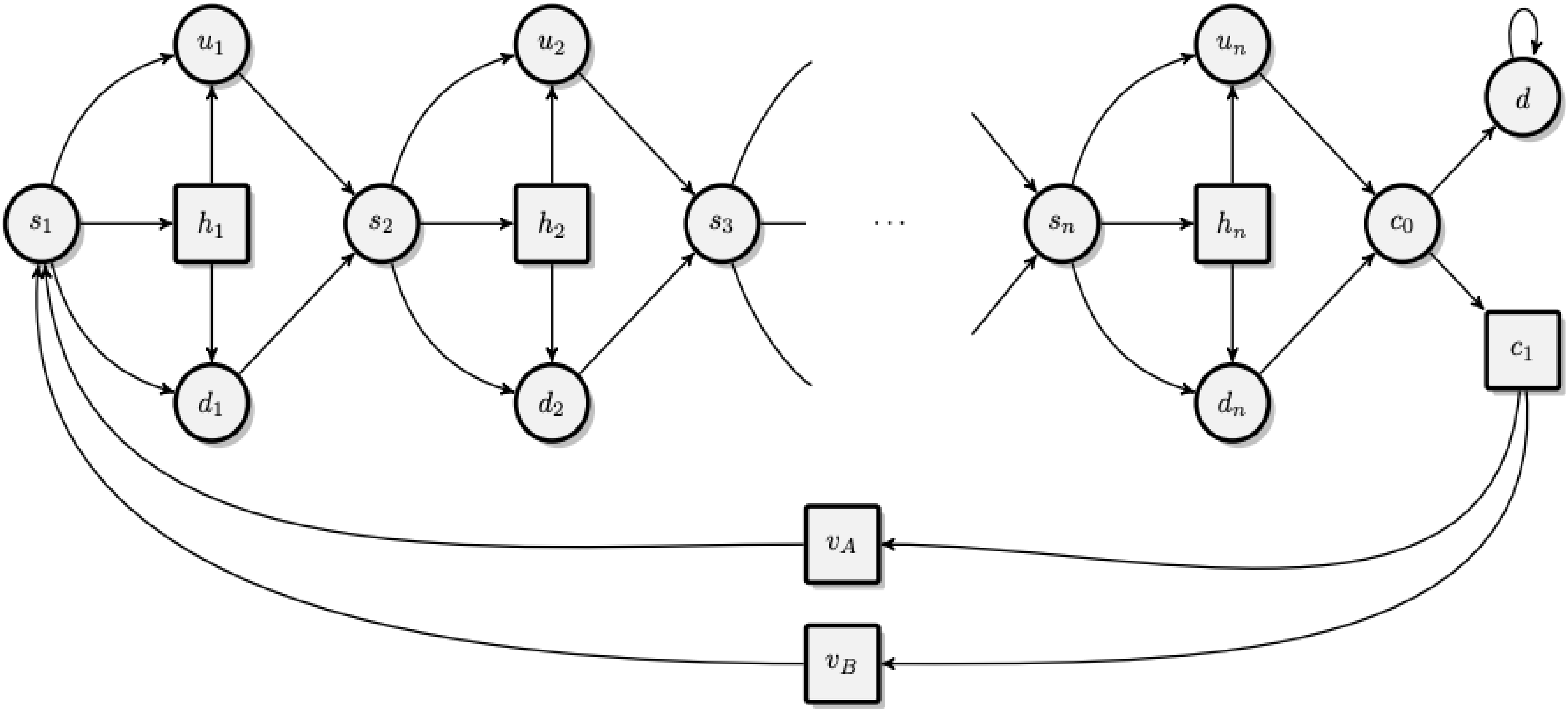
Weak Muller Game (Family)

There exists a family Gn = (An, wMuller(Fn)) of weak Muller games, each having a designated vertex v, such that

- |An| ∈ O(n) and |Fn| = 2,
- Player 0 has a finite-state winning strategy from v, but
- Player 0 has no finite-state winning strategy from v with less than 2^n states.



Family of Weak Muller games



Limits on Reduction

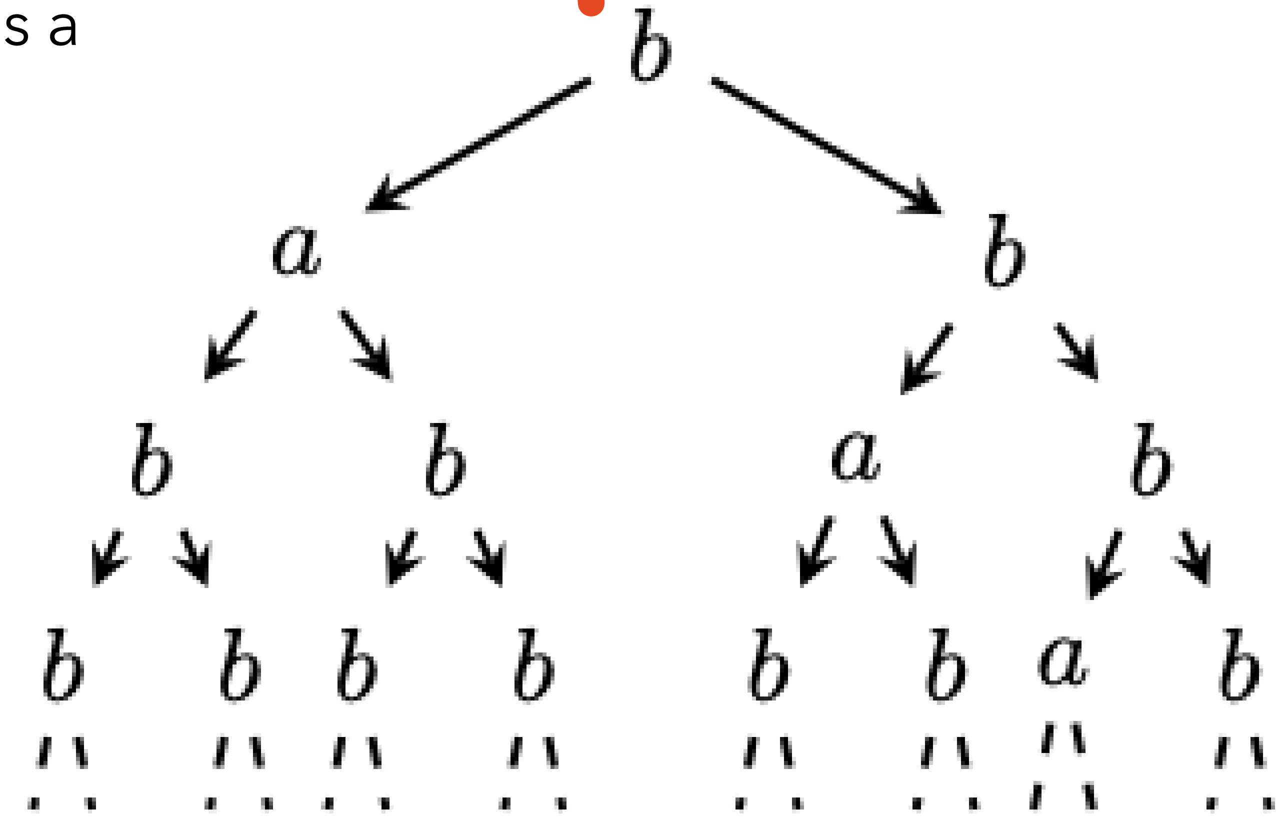
- A Büchi condition is harder than a reachability condition
- Intuitively, In a Büchi game, vertices from F have to be visited infinitely often while in a reachability game it suffices to visit R once, which is a much weaker condition.
- Reductions cannot go “down” the hierarchy: A complicated language cannot be reduced to a simpler one.

Rabin's Theorem

- Infinite Tree
- S2S
- Parity Tree Automata
- Rabin's theorem

Infinite Tree

- A tree over an alphabet Σ is a mapping $t: B* \to \Sigma$
- Ex:
- $te(w) = \begin{cases} a & \text{if } w = 1*0, \\ b & \text{otherwise.} \end{cases}$



S2S

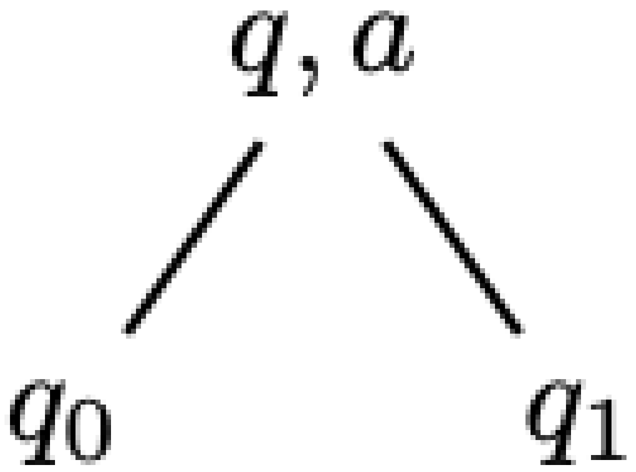
- The Monadic Second-Order Logic of Two Successors
- Syntax of S2S
- If $ is a formula, then Complement, union, intersection, quantification, also formula
- Sentence is a formula without free variables
- Solution in from of trees which satisfy the sentence

S2S (Cont.)

- Satisfiability problem for S2S
- Translates a sentence φ into an automaton A_φ recognizing exactly the trees satisfying φ
- Typically simpler to solve

Parity Tree Automata

- A parity tree automaton A = (Q, Σ, qI , Δ, Ω)



Parity Tree Automata (An example)

- $L1 = \{t : B* \to \{a, b\} \mid t|\pi = b\,\omega$ for some path $\pi\}$, the language of trees containing a path labelled with $b\,\omega$

  - $Q_1 = \{q_I, q_*\}$,
  - $\Omega_1(q_I) = \Omega_1(q_*) = 0$, and
  - $\Delta_1 = \left\{ \begin{array}{c} q_I, b \\ \diagup \qquad \diagdown \\ q_I \qquad\qquad q_* \end{array} , \begin{array}{c} q_I, b \\ \diagup \qquad \diagdown \\ q_* \qquad\qquad q_I \end{array} , \begin{array}{c} q_*, a \\ \diagup \qquad \diagdown \\ q_* \qquad\qquad q_* \end{array} , \begin{array}{c} q_*, b \\ \diagup \qquad \diagdown \\ q_* \qquad\qquad q_* \end{array} \right\}.$

Thank You