Daniel Dahrendorf
Seminar: Games in Verification and Synthesis
(University of Saarland, Reactive Systems Group)
July 17, 2008

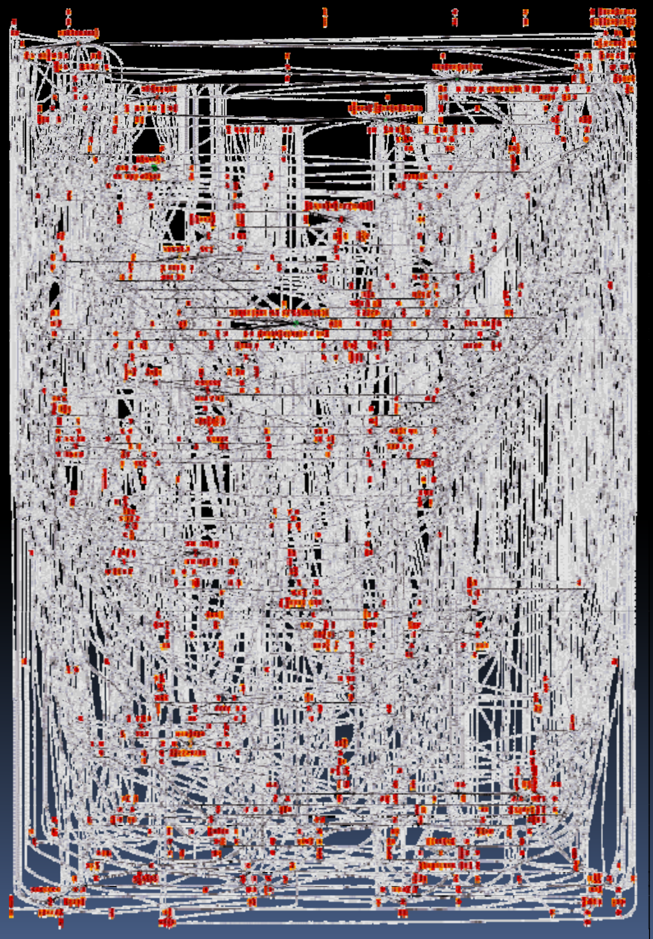
COUNTEREXAMPLE GUIDED CONTROL

PAPER BY: THOMAS A. HENZINGER, RANJIT JHALA AND RUPAK MAJUMDAR

Motivation

- Consider 2 player games:
  system (player 1) vs. environment (player 2)
- Transition systems model the interaction between a system and the environment
- Model checker can check these transition systems for control

Motivation



Very large transition system

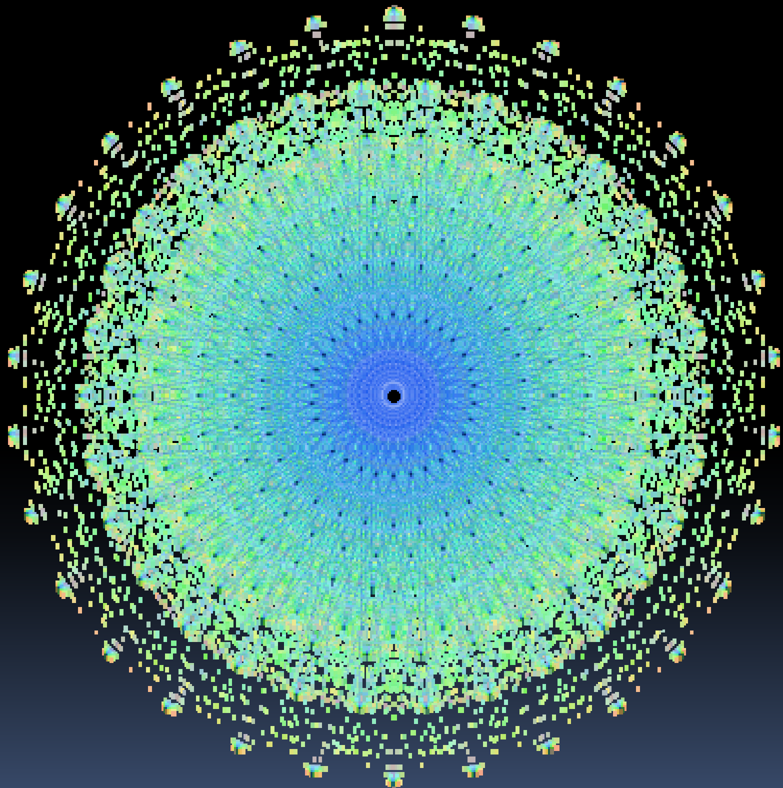
Solving such systems is practically infeasible

Motivation

Imagine a program component with variables over unbounded data domains (e.g. integers).

How would a transition system of such a program looks like?

Motivation



Infinite transition system

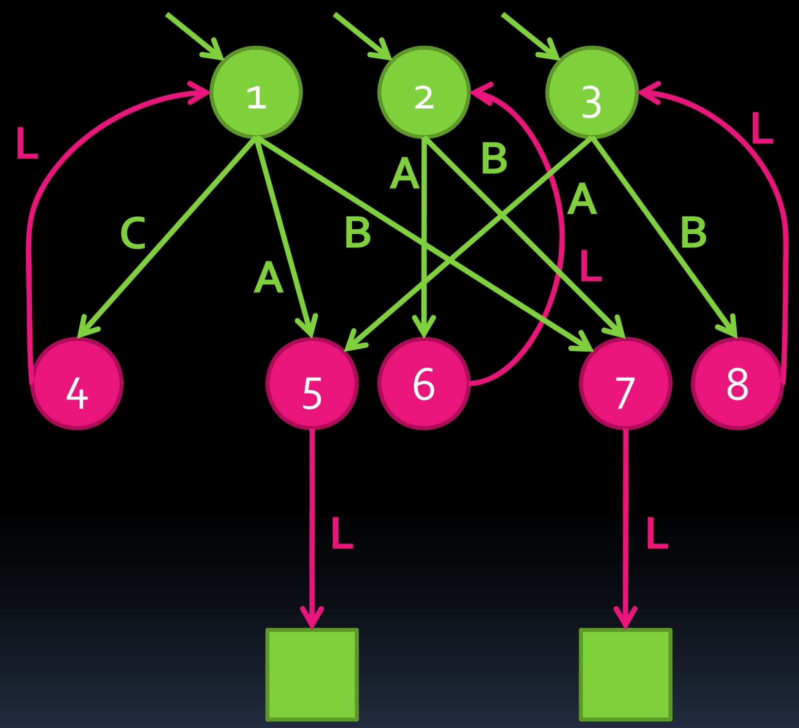
No direct application of our finite state algorithms possible

What should we do?

Is there a possibility for a automatically simplification of such transition systems?

Lets see…

Two-player game structure



Φ : set of propositions

Λ : set of labels

$\mathcal{G}$ = (V_1, V_2, δ, P):

V_1 : player 1 nodes

V_2 : player 2 nodes

$(V = V_1 \cup V_2)$

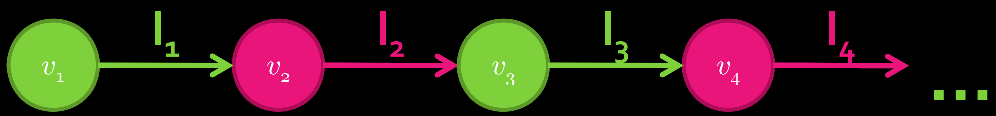
δ : $\delta \subseteq V \times \Lambda \times V$

P : $V \rightarrow 2^{\Phi}$

$init$: $init \in \Phi$

Runs and strategies

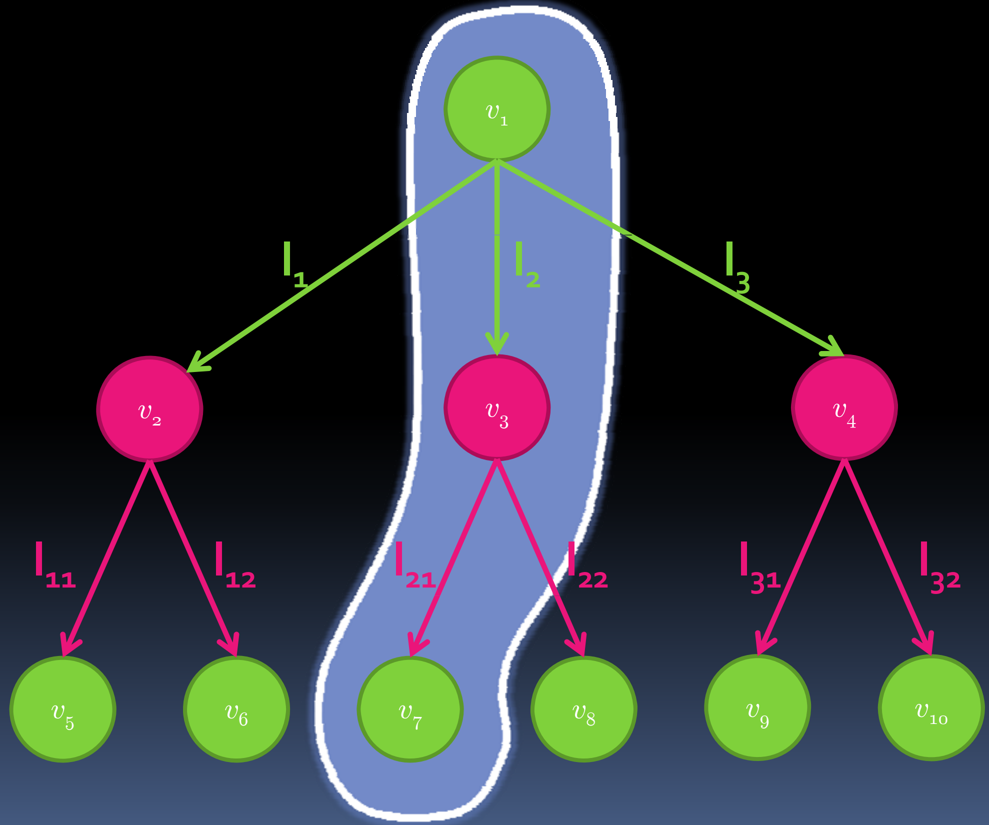
- Run: v_1
  finite or infinite sequence $v_1 v_2 v_3 \ldots$ of states



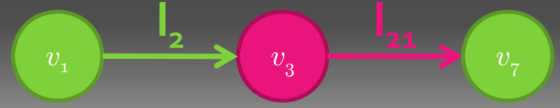
- Strategy for player i: function f_i: $V^* \cdot V_i \rightarrow \Lambda$

Outcome

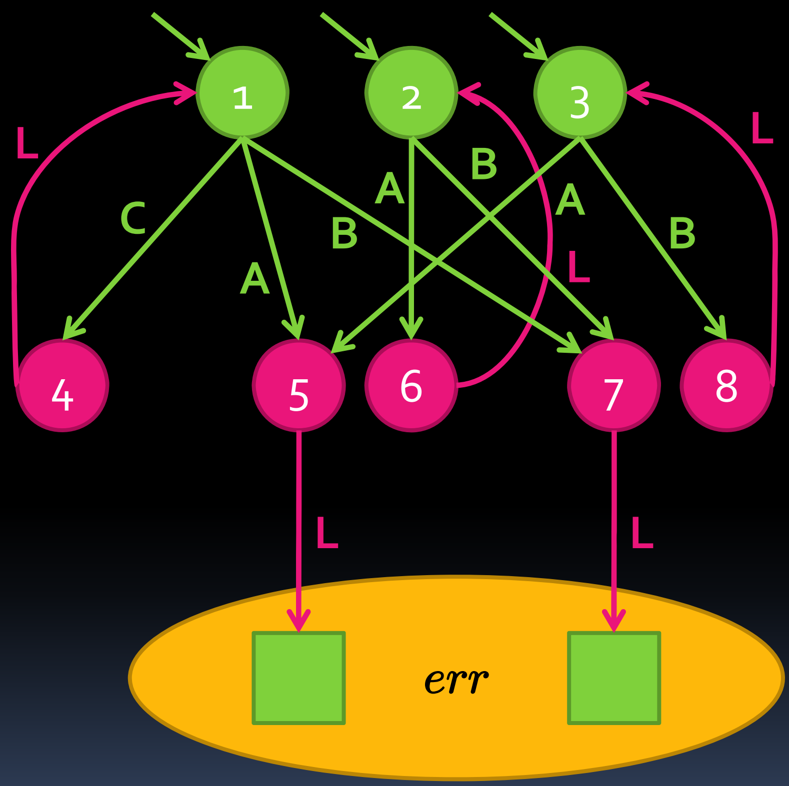
- Outcome for strategies f_1 and f_2: $\Omega_{f_1,f_2}(v)$



Possible outcome:



Two-player safety game



safety game $(\mathcal{G}, \varphi)$:

$\mathcal{G}$: game structure

φ : LTL formula over Φ

φ has the form $\Box\overline{err}$

Goal of player 1:

Avoid states which satisfy err

Winning / spoiling strategies

- Let Π_1 be the set of runs where player 1 wins
  - Infinite run that never visits an error state

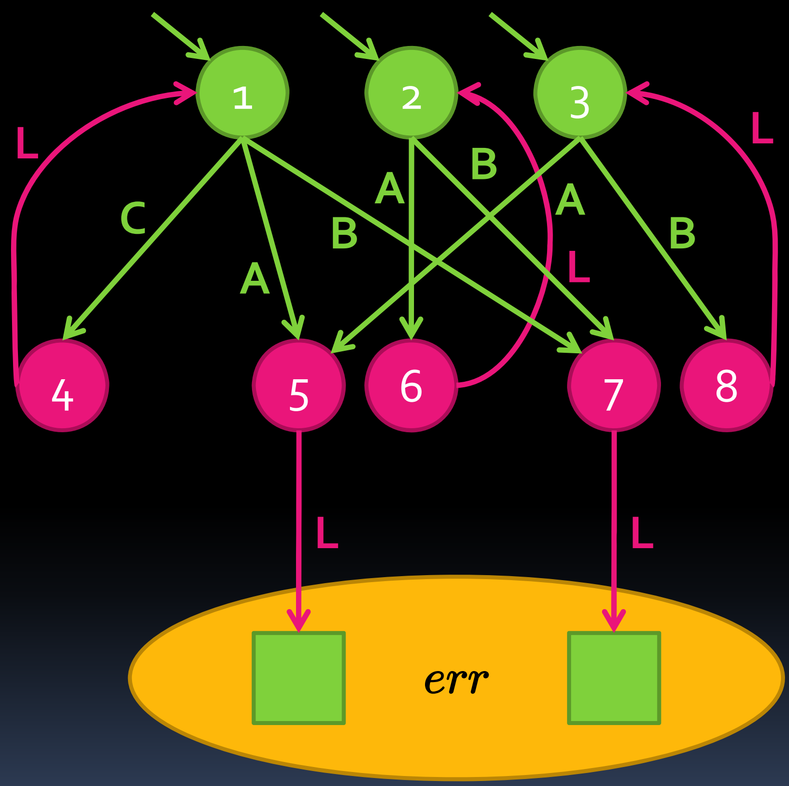
- Strategy f1 is winning for player 1 if for all strategies f2 of player 2 and all initial states v:

  $$\Omega_{f_1,f_2}(v) \subseteq \Pi_1$$

- Strategy f2 is spoiling for player 2 if for all strategies f1 of player 1 and a initial state v:

  $$\Omega_{f_1,f_2}(v) \not\subseteq \Pi_1$$

Example: Winning strategy



Winning strategy for player 1:

- At state 1 she plays C
- At state 2 she plays A
- At state 3 she plays B

Problem

Game graph might be

- Very large
  - solving the game is practically infeasible
- Infinite
  - Algorithms for finite state case cannot be applied directly

The solution: abstraction

Obtain a simplification of the game which is

1. less expensive to solve

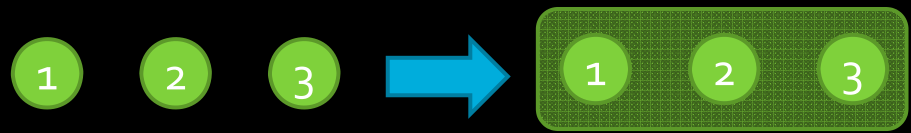
   ⇒ smaller / finite state space

2. sound

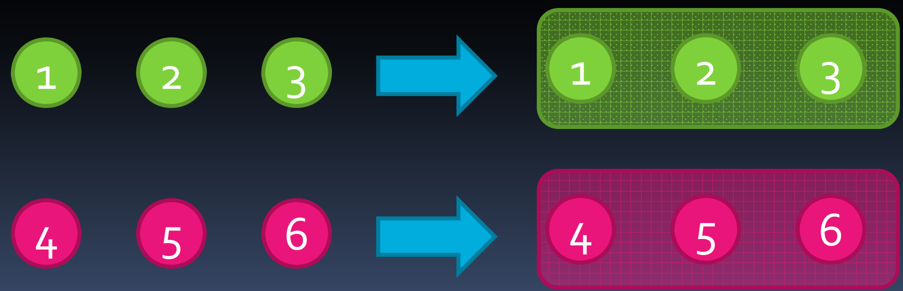
   ⇒ if player 1 wins the abstract game he wins also the concrete game

Abstract states

1. An abstract state $v^\alpha \in V^\alpha$ represents a set $[\![v^\alpha]\!] \subseteq V$ of concrete states

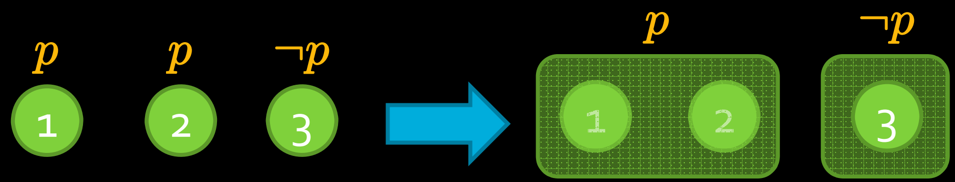


2. Player structure is preserved:

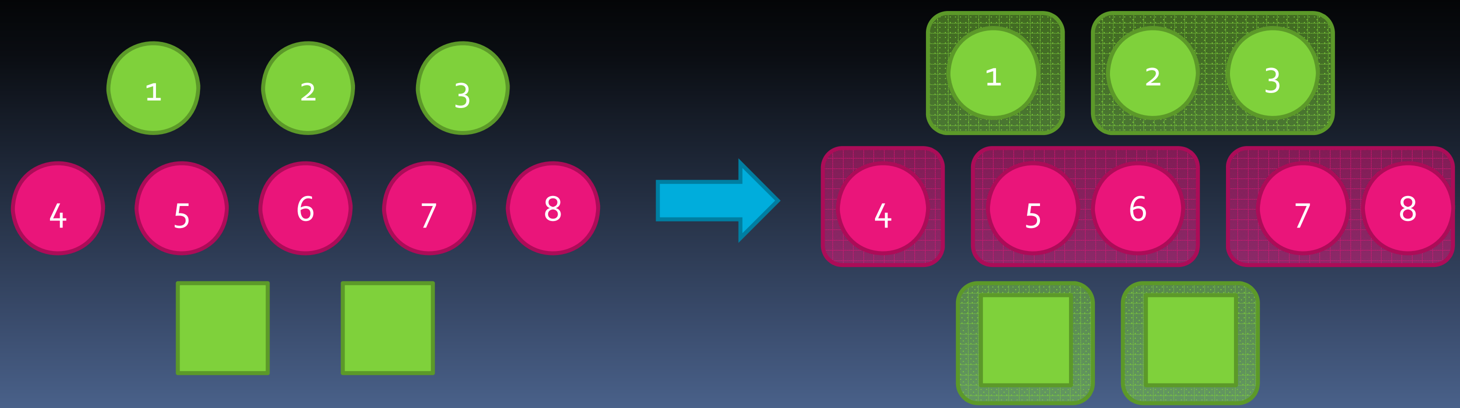


Abstract states

(3) Propositions are preserved:



(4) The abstract states cover the whole concrete state space:



How to ensure soundness?

- Restrict the power of player 1

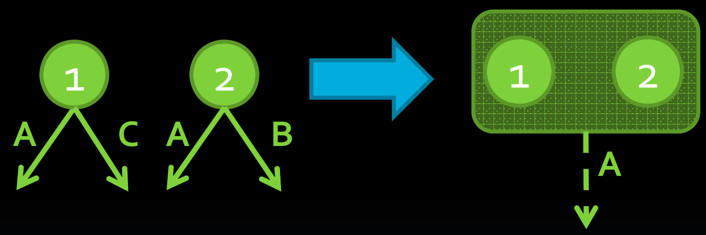
  $\Rightarrow$ Player 1 has fewer moves in the abstraction

- Increase the power of player 2

  $\Rightarrow$ Player 2 has more moves in the abstraction

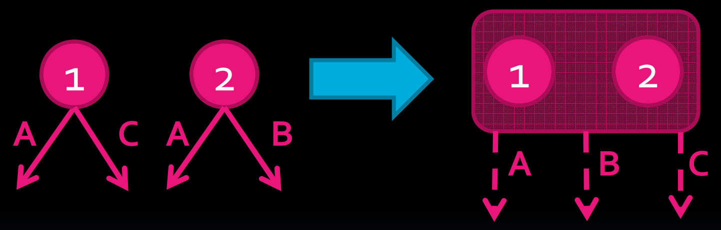
Abstract moves for player 1

From each abstract v^α player 1 state only moves are allowed which could be played from each concrete state $v \in [\![v^\alpha]\!]$:



Abstract moves for player 2

From each abstract v^α player 2 state all moves are allowed which could be played from a concrete state $v \in [\![v^\alpha]\!]$:



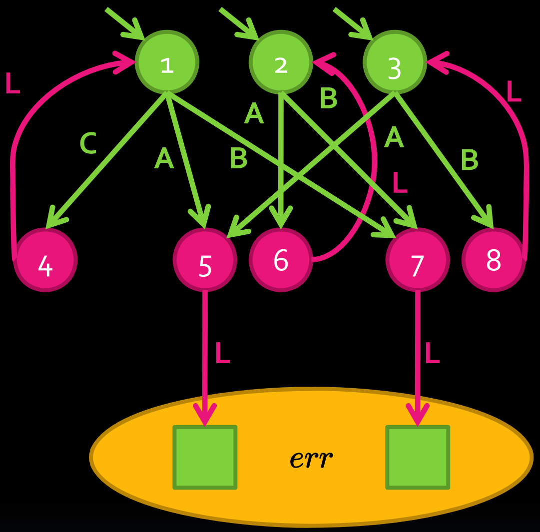
Abstraction

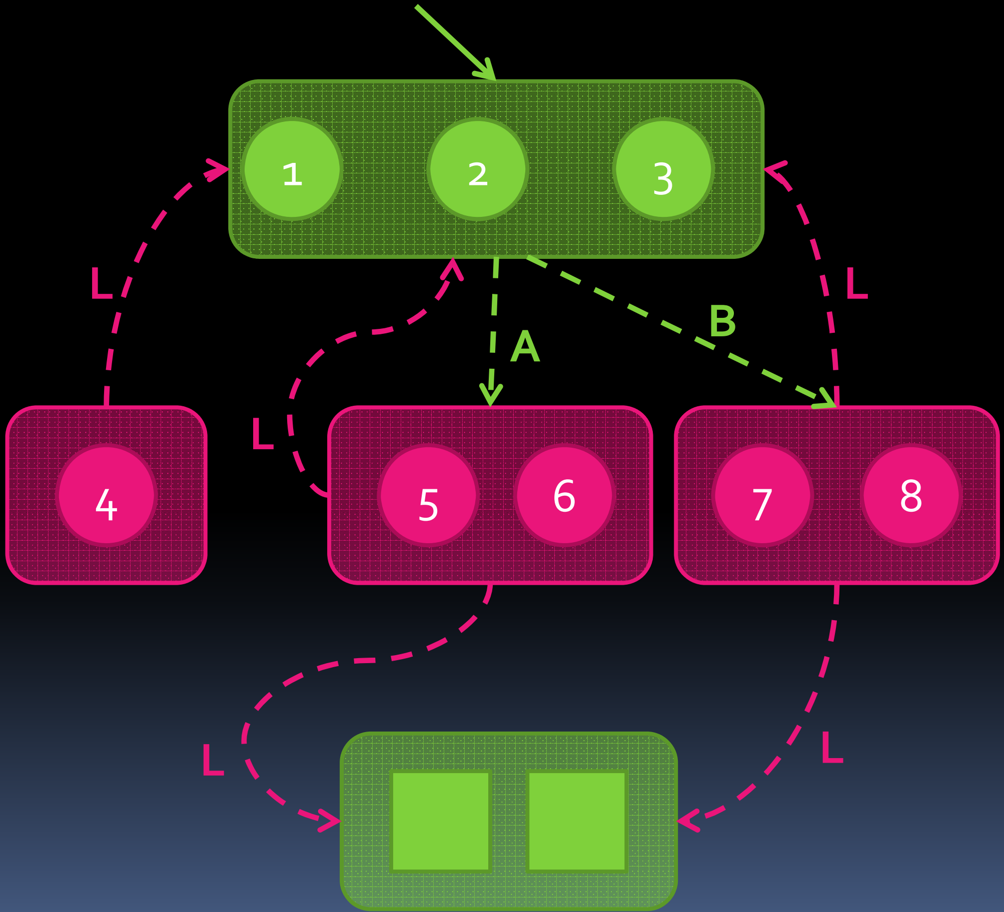
Abstraction for a game structure $\mathcal{G}$:

game structure $\mathcal{G}^\alpha = (V_1^\alpha, V_2^\alpha, \delta^\alpha, P^\alpha)$

and a concretization function $[\![\cdot]\!] : V^\alpha \to 2^V$

Example: abstraction





Question

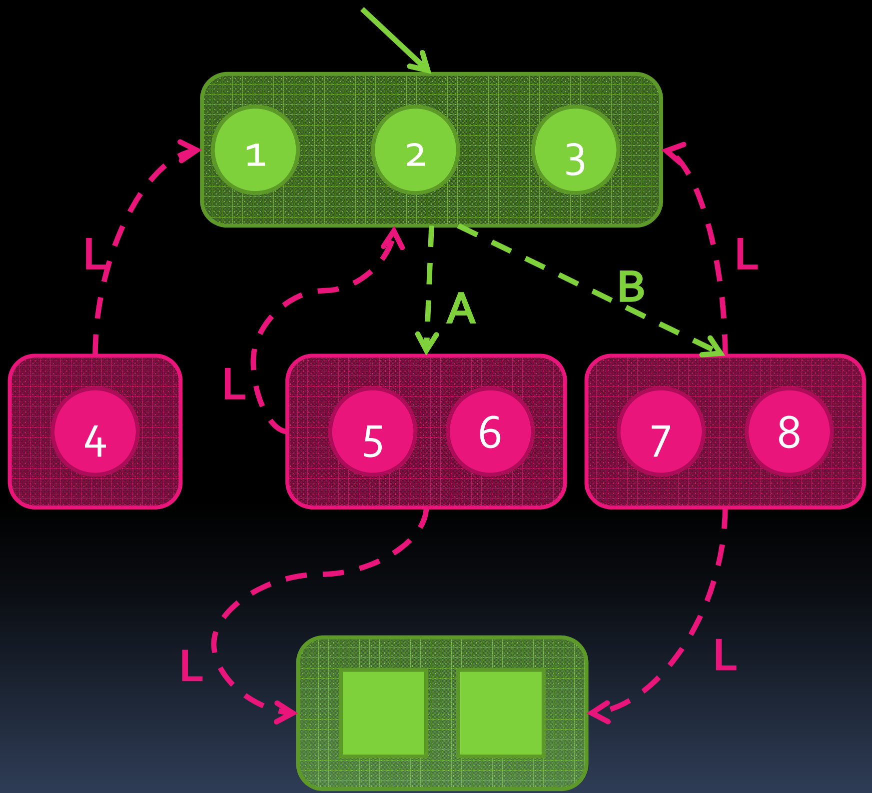
If player 1 wins ($\mathcal{G}^{\alpha}$,φ) she wins also ($\mathcal{G}$,φ)

But what if player 2 has a spoiling strategy for ($\mathcal{G}^{\alpha}$,φ)?

Counterexample

- Spoiling strategy for player 2 = counterexample that player 1 can't win the game
- Genuine counterexample corresponds to one in the concrete game
- Spurious counterexample arises due coarseness of the abstraction

Example: Spurious counterexample



Spoiling strategy for player 2:

- At state {5,6} she plays L $\Rightarrow$ error state
- At state {7,8} she plays L $\Rightarrow$ error state

Counterexample

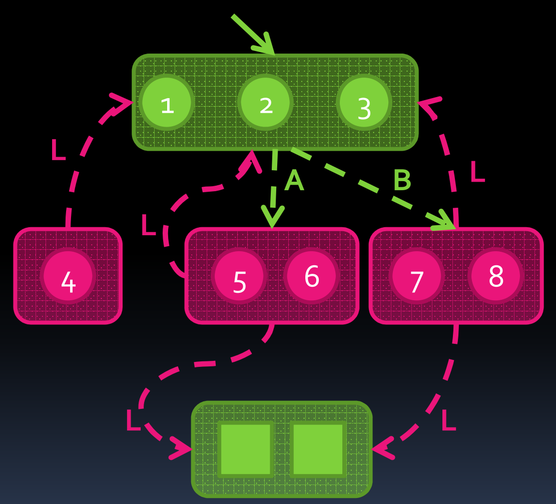
We want to:

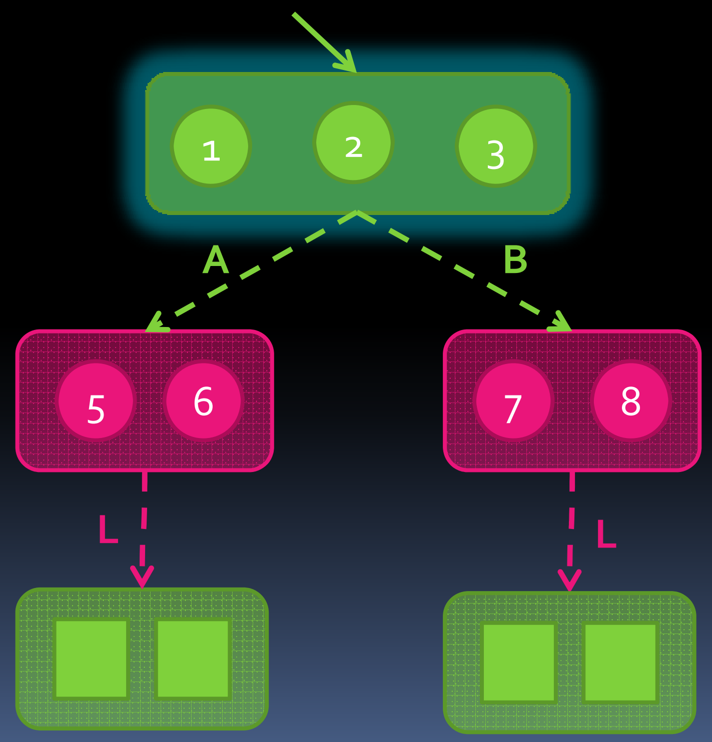
- Determine whether a counterexample is genuine
- Automatically refine the abstraction to rule out a spurious counterexample

Counterexamples are represented by finite labeled trees: Abstract counterexample tree (ACT)

Abstract counterexample trees (ACT)

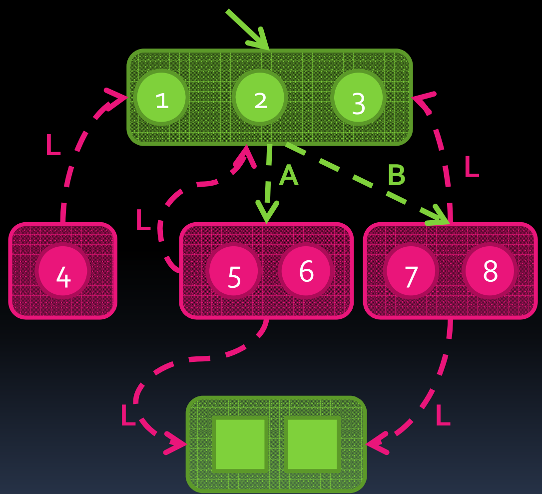
Root labeled by $v^\alpha \Rightarrow [\![v^\alpha]\!] \subseteq [init]$

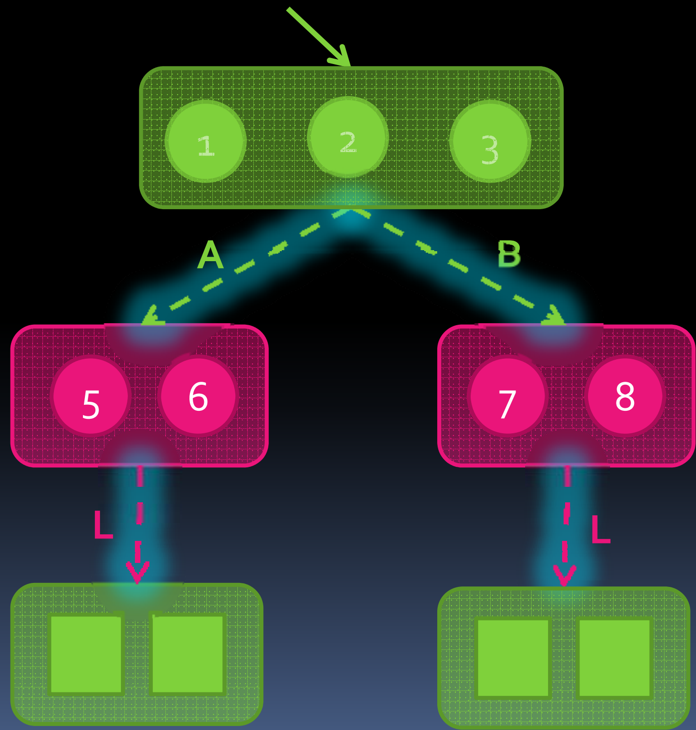




Abstract counterexample trees (ACT)

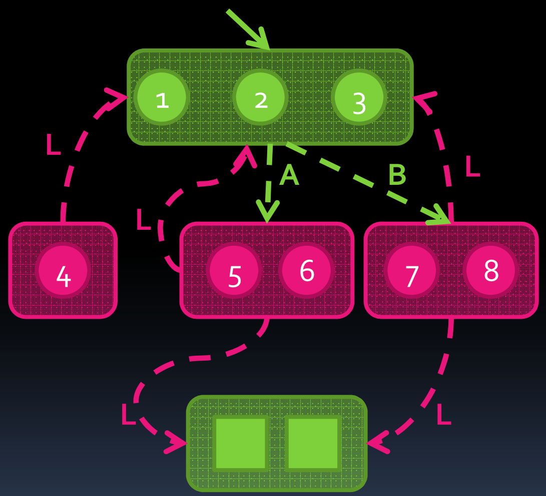
n': w^α l-child of n: $v^\alpha \Rightarrow \delta^\alpha(v^\alpha, l, w^\alpha)$

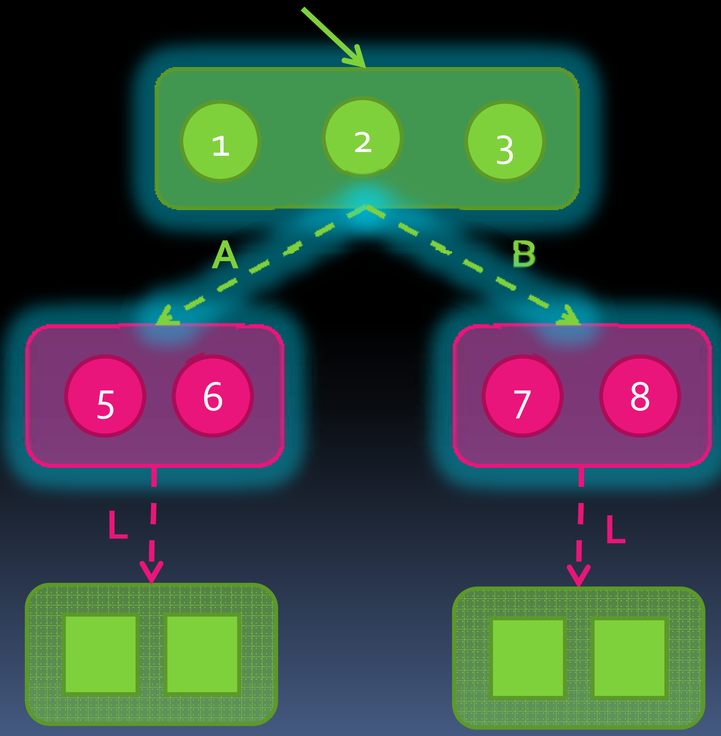




Abstract counterexample trees (ACT)

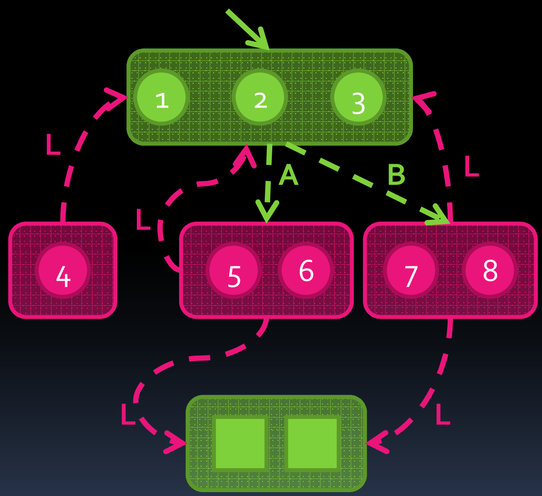
n: v^α non-leaf player 1 node $\Rightarrow$
for each $l \in L^\alpha(v^\alpha)$ n has at least one l-child

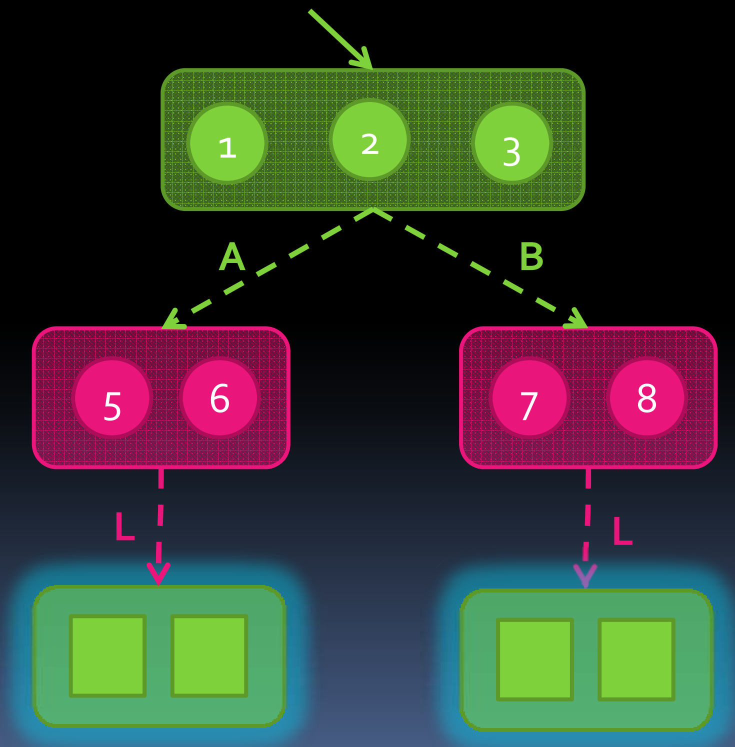




Abstract counterexample trees (ACT)

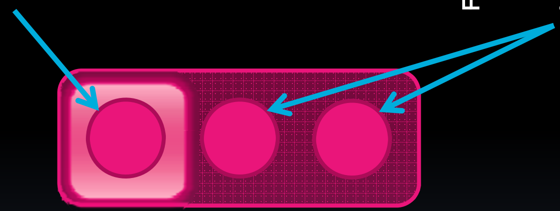
Leaf labeled by v^α: $L^\alpha(v^\alpha) = \emptyset$ or $[\![v^\alpha]\!] \subseteq [err]$





Analyze a counterexample

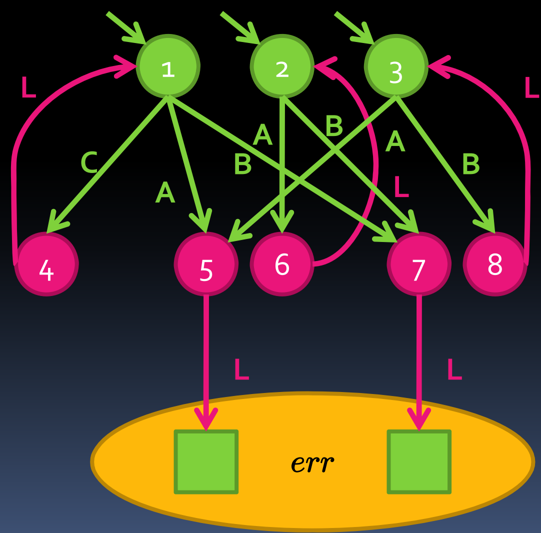
- Concrete node $v \in [\![v^\alpha]\!]$ is part of a spoiling strategy
  $\Rightarrow$ A successor of v is part of a spoiling strategy
- Divide each v^α in a good set r and a bad set $[\![v^\alpha]\!] \setminus r$

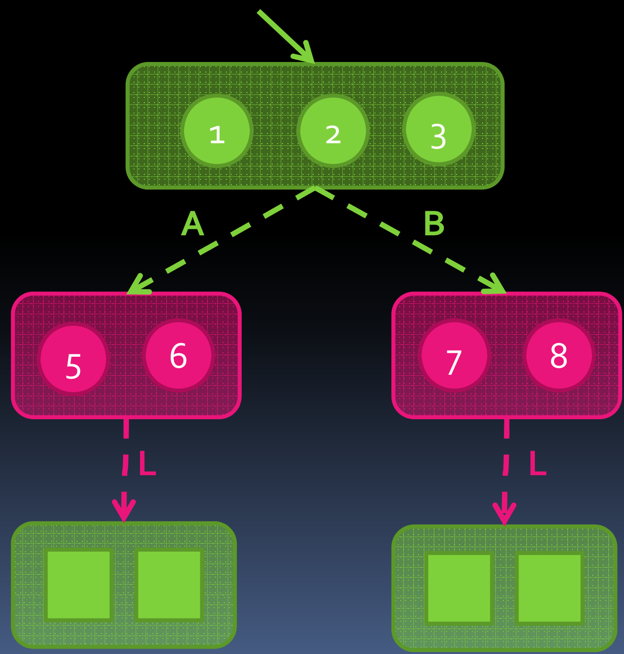


- ACT T^α is genuine iff its root contains a non-empty good set

Analyze a counterexample

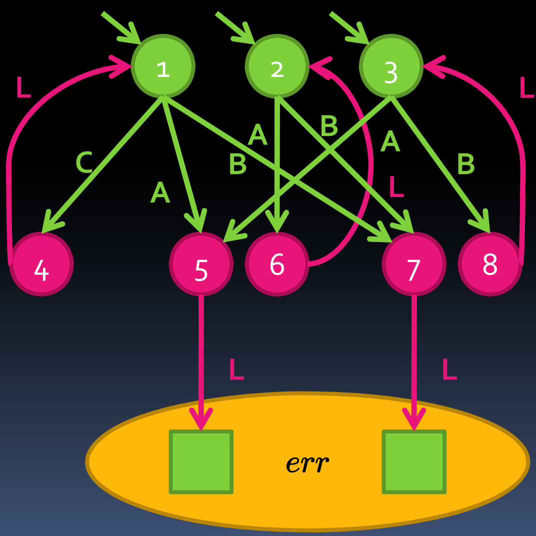
Annotate each node n of a given ACT T^{α} with r (Focusing) under following rules:

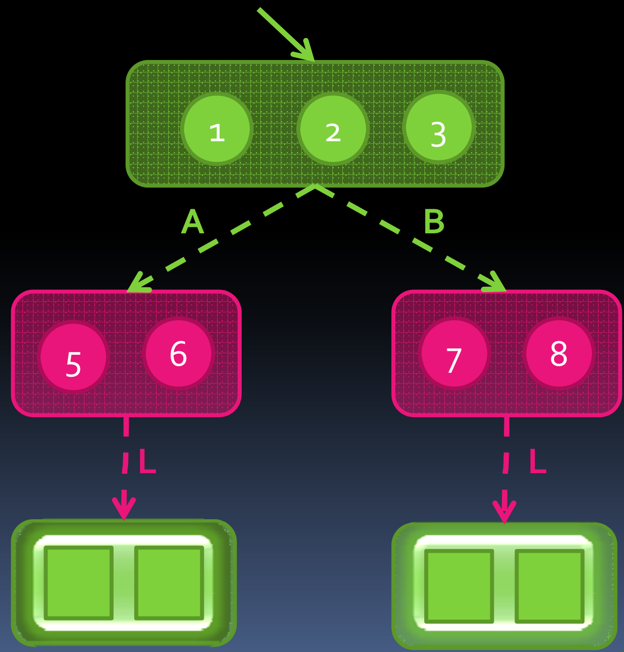




Analyze a counterexample

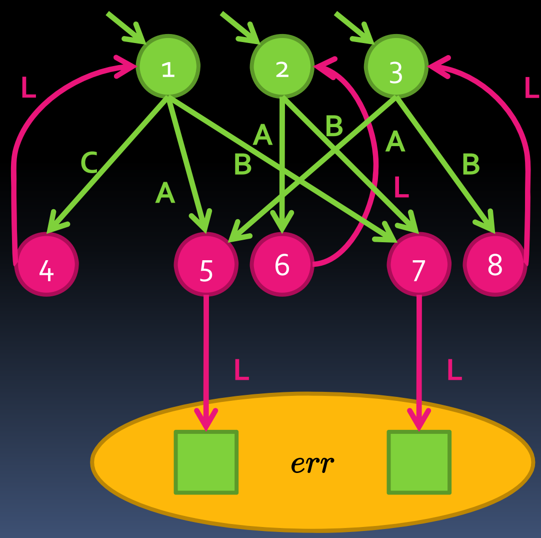
Node is a leaf: $r = [\![v^\alpha]\!]$

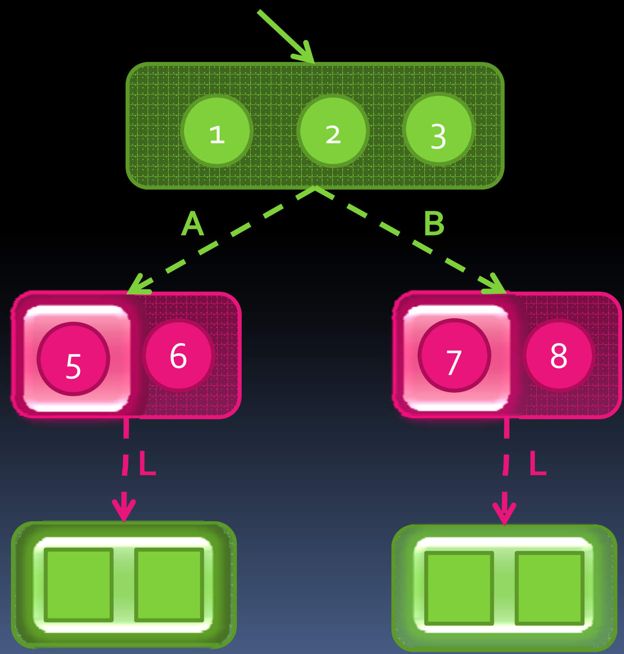




Analyze a counterexample

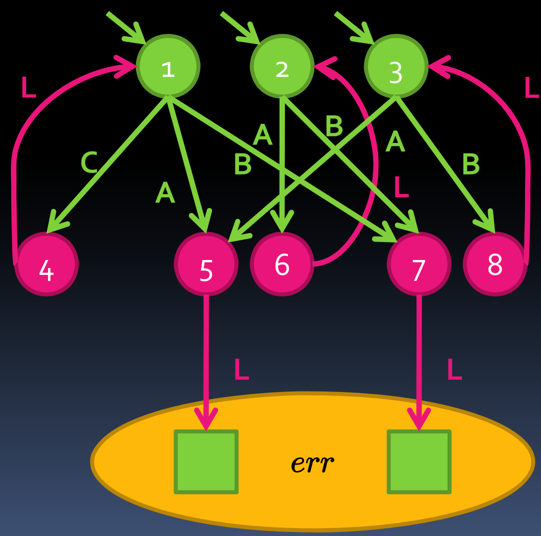
Node is a player 2 state: $v \in r$ if there is successor of v from where player 2 can spoil

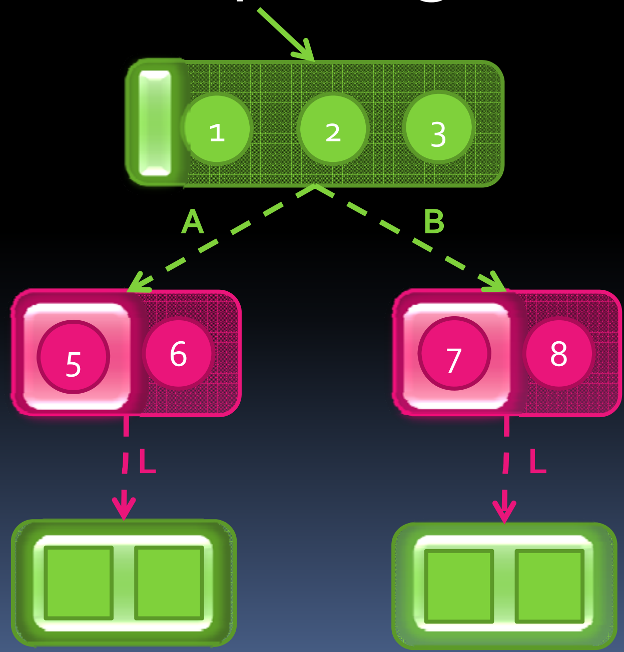




Analyze a counterexample

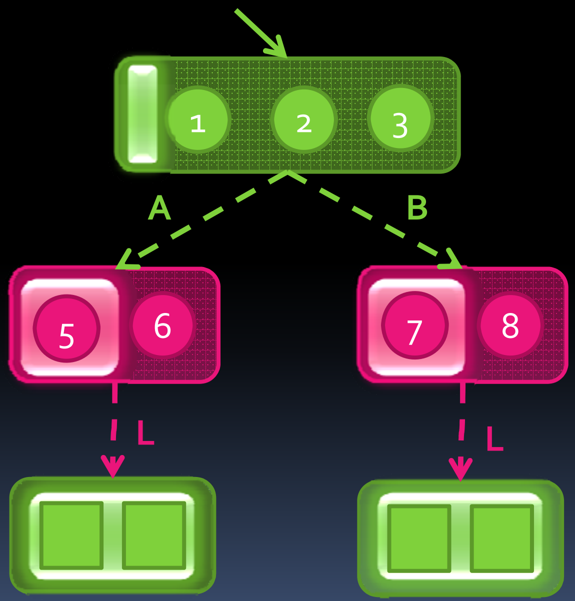
Node n is a non leaf player 1 state: $v \in r$
if all moves of v are also outgoing edges of n
and for every edge of n there is a spoiling succ. of v





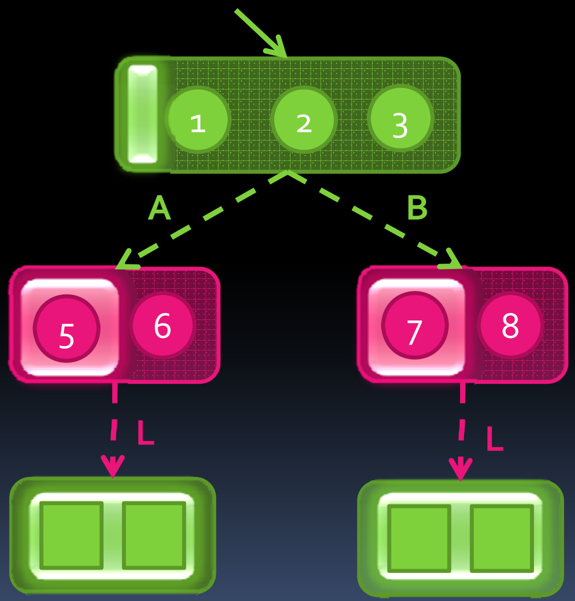
Abstraction refinement

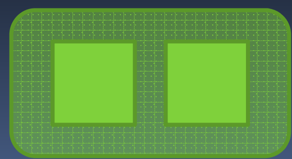
Use the annotated ACT T^{α} to refine the abstraction (Shattering):



Abstraction refinement

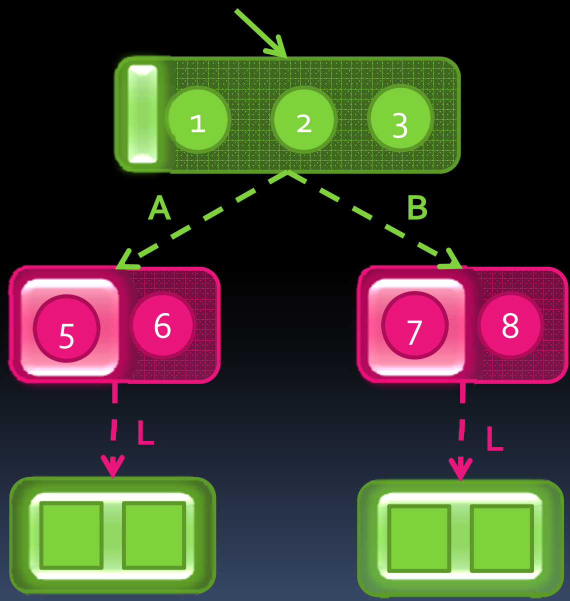
Split player 1 nodes in the good set $\{r\}$

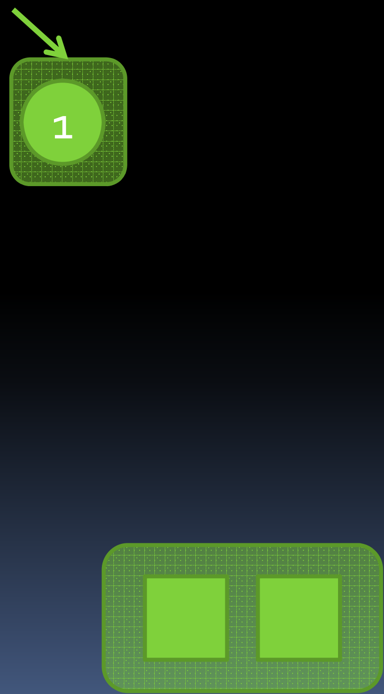




Abstraction refinement

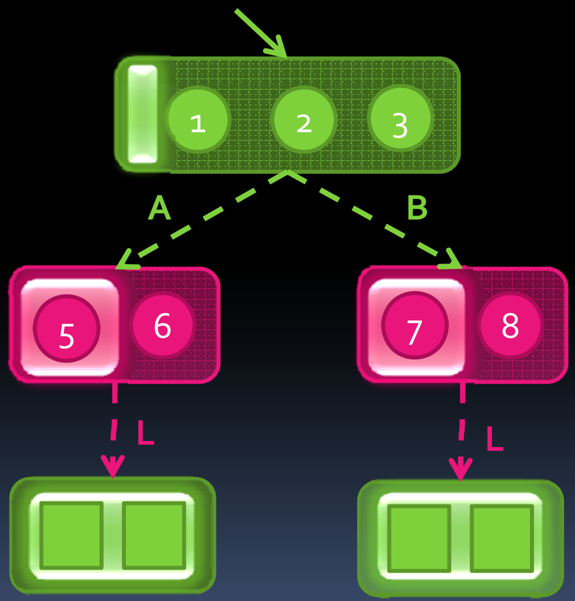
Split player 1 nodes in bad sets of nodes which have moves which are no edges of n

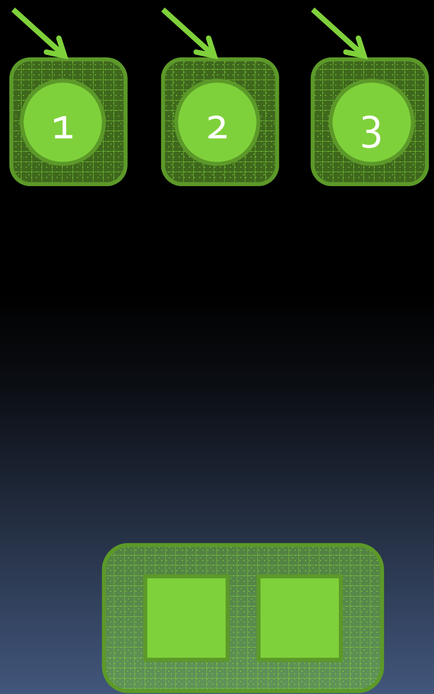




Abstraction refinement

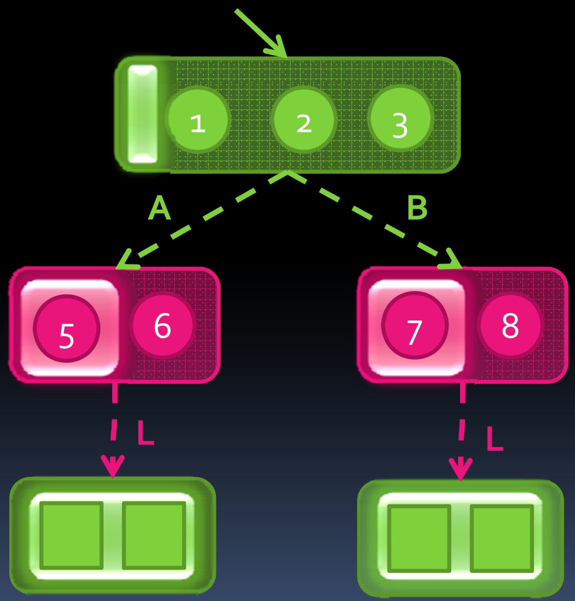
Split player 1 nodes in bad sets of nodes which have a move such that the succ. is not in a good set

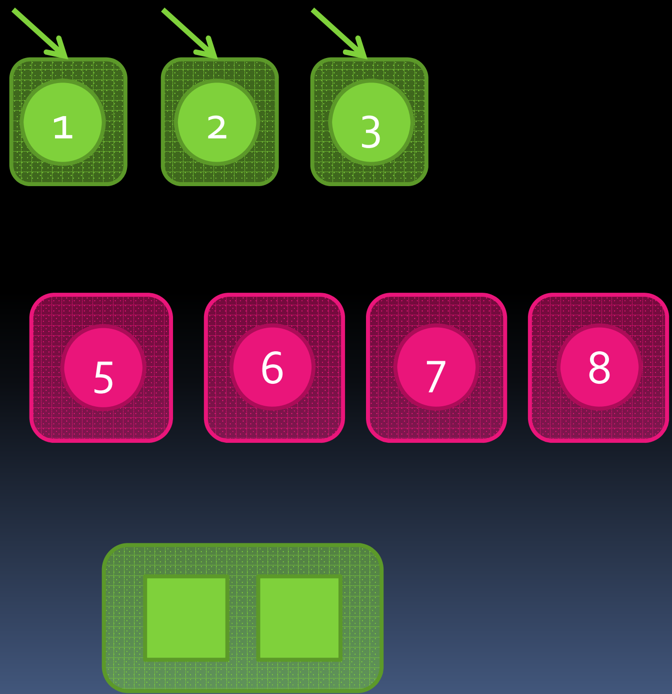




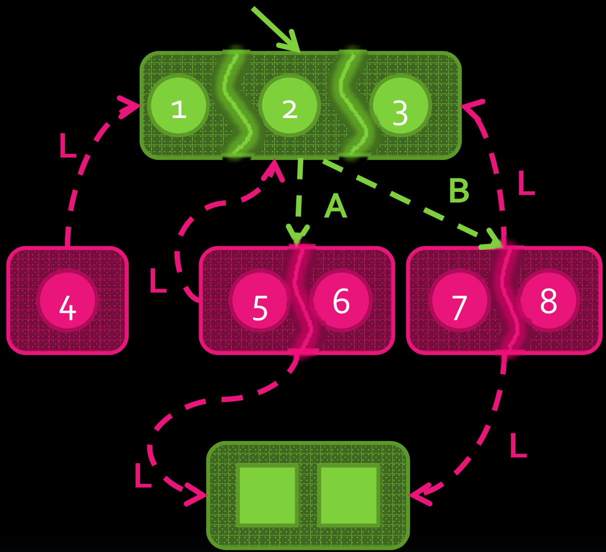
Abstraction refinement

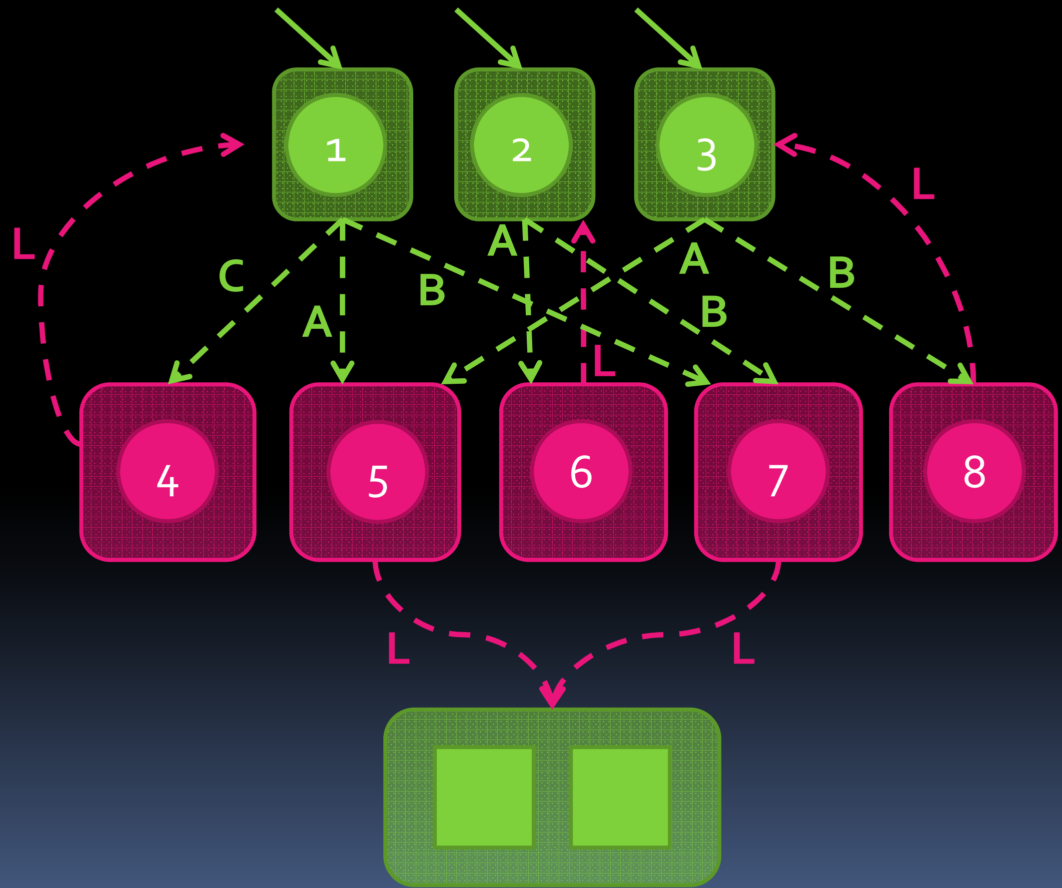
Split player 2 in the good set $\{r\}$ and the bad set $[\![v^\alpha]\!] \setminus r$



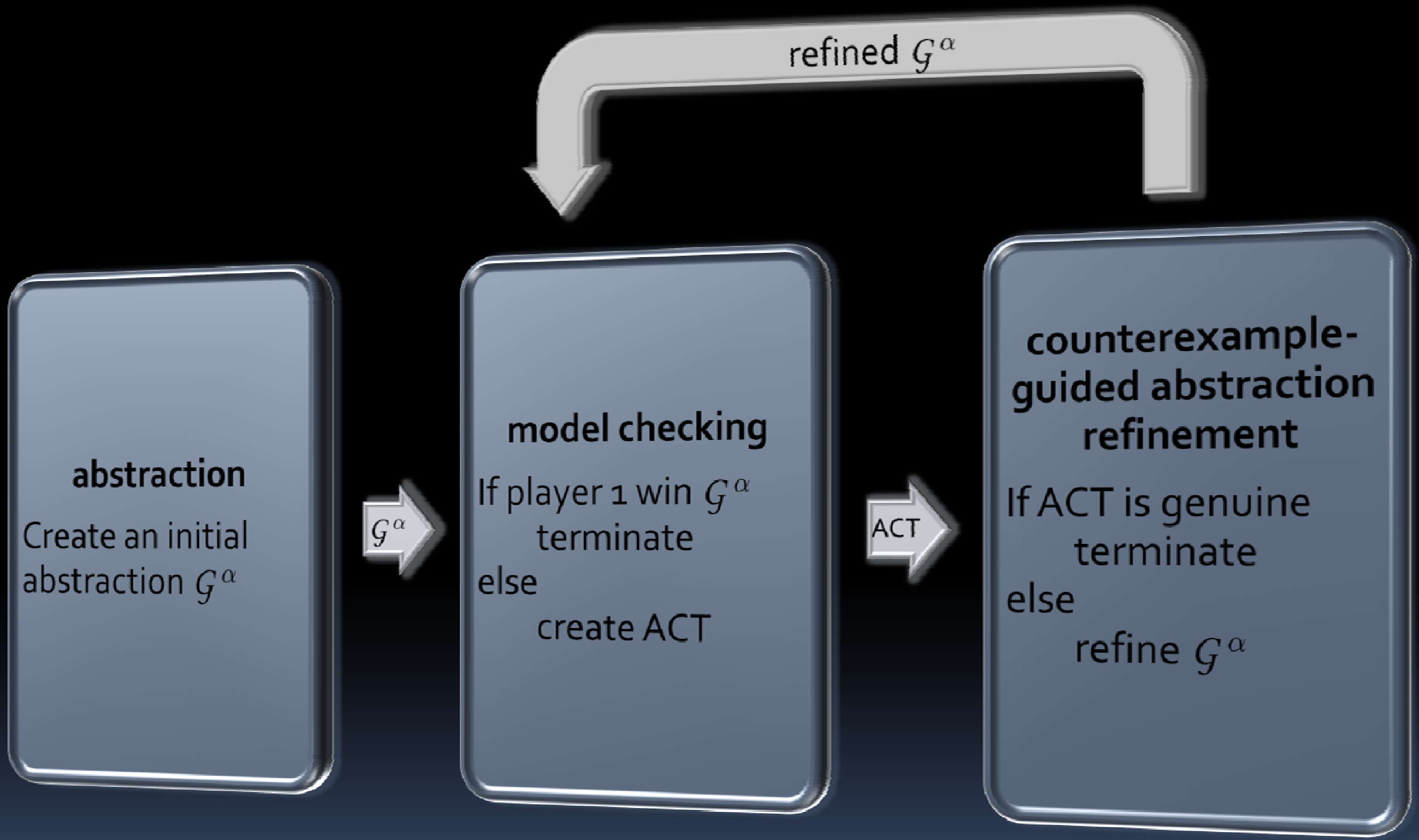


Example: refined Abstraction





Counterexample-Guided Controller Synthesis



Termination

Bad news:

The refinement loop may not terminate in general for infinite state games

Good news:

Termination is guaranteed for finite games and certain state equivalences with finite index

Conclusion

- Abstraction reduces the state space
- Sound abstraction: ensures that if player 1 wins the abstract game he wins also the concrete game
- Automatic refinement guided by spurious spoiling strategies
- Can be extended to arbitrary ω-regular objectives

References

1. T. Henzinger, R. Jhala, and R. Majumdar. Counterexample-guided control. In Proc. 30th Int. Colloquium on Automata, Languages, and Programming (ICALP), volume 2719 of LNCS, pages 886--902, 2003.