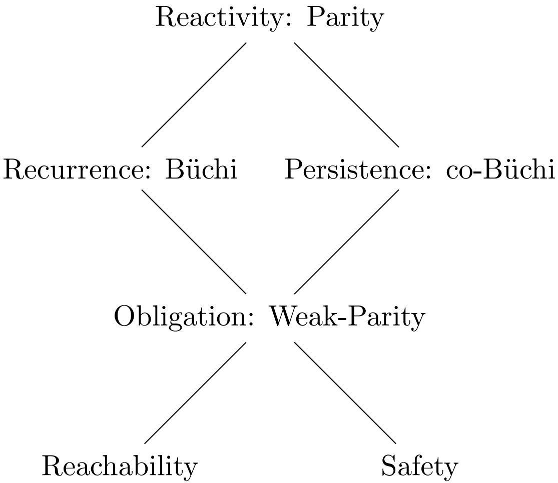
Reachability and Büchi Games

Barbara Jobstmann

CNRS/Verimag

Hierarchy


Reactivity: Parity
Recurrence: Büchi
Persistence: co-Büchi
Obligation: Weak-Parity
Reachability
Safety

Reachability and Safety Games

Reachability and Safety Games

Theorem

In a reachability game (G, F) *with* $G = (S, S_0, E)$ *and* $F \subseteq S$*, the winning regions* W_0 *and* W_1 *are computable. Both players have corresponding memoryless winning strategies.*

Proof.

W_0 are all the states from which Player 0 can "force" a visit to F.

Idea: define the *i-attractor of Player 0*, which are all the states from which Player 0 can "force" a visit to F in i steps.

Then, define the *attractor of Player 0* as union of the i-attractor of Player 0 for all i.

Reachability and Safety Games

Theorem

In a reachability game (G, F) with $G = (S, S_0, E)$ and $F \subseteq S$, the winning regions W_0 and W_1 are computable. Both players have corresponding memoryless winning strategies.

Proof.

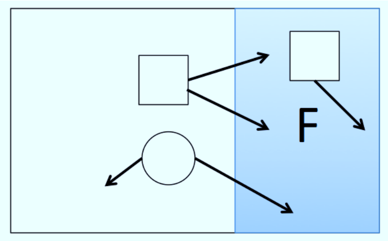
W_0 are all the states from which Player 0 can "force" a visit to F.
Idea: define the *i-attractor of Player 0*, which are all the states from which Player 0 can "force" a visit to F in i steps.
Then, define the *attractor of Player 0* as union of the i-attractor of Player 0 for all i.

Claim (1): Attractor of Player 0 is computable
Claim (2): $W_0 =$ Attractor and $W_1 = S \setminus$ Attractor

“Forced” Visit in Next Step (“Controlled Next”)

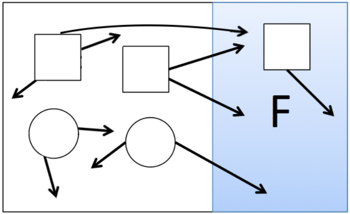
Given a set of states, compute the set of states $\mathrm{CNext}_0(F)$ from which of Player 0 can force to visit F in the next step. I.e., for each state $s \in \mathrm{CNext}_0(F)$ Player 0 can fix a strategy s.t. all plays starting in s visit F in the first step.

$$\begin{aligned}\mathrm{CNext}_0(F) \quad = \quad & \{s \in S_0 \mid \exists s' \in S : (s, s') \in E \wedge s' \in \mathrm{Attr}_0^i(F)\} \cup \\ & \{s \in S_1 \mid \forall s' \in S : (s, s') \in E \rightarrow s' \in \mathrm{Attr}_0^i(F)\}\end{aligned}$$



Observations about $\text{CNext}(F)$

1. Player 0 has a memoryless winning strategy from $\text{CNext}(F)$ to reach F in the next step.
2. Player 1 has a memoryless winning strategy from $S \setminus \text{CNext}(F)$ to avoid F in the next step.
3. $S \setminus \text{CNext}(F)$ is a sub-game, i.e., every state has at least one successor.
4. $S \setminus \text{CNext}(F)$ is a trap for Player 0, i.e., she cannot leave it.



Computing the Attractor

Construction of $i-$Attractor $\mathrm{Attr}_0^i(F)$:

$$\begin{aligned}\mathrm{Attr}_0^0(F) &= F\\ \mathrm{Attr}_0^{i+1}(F) &= \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F))\end{aligned}$$

$$\mathrm{Attr}_0(F) = \bigcup_{i=0}^{\infty} \mathrm{Attr}_0^i(F)$$

Computing the Attractor

Construction of $i-$Attractor $\mathrm{Attr}_0^i(F)$:

$$\begin{aligned} \mathrm{Attr}_0^0(F) &= F \\ \mathrm{Attr}_0^{i+1}(F) &= \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F)) \\ \\ \mathrm{Attr}_0(F) &= \bigcup_{i=0}^{\infty} \mathrm{Attr}_0^i(F) \end{aligned}$$

Then $\mathrm{Attr}_0^0(F) \subseteq \mathrm{Attr}_0^1(F) \subseteq \mathrm{Attr}_0^2(F) \subseteq \dots$ and since S is finite, there exists $k \leq |S|$ s.t. $\mathrm{Attr}_0^k(F) = \mathrm{Attr}_0^{k+1}(F)$.

The attractor of Player 0 can be computed as

$$\mathrm{Attr}_0(F) = \bigcup_{i=0}^{|S|} \mathrm{Attr}_0^i(F)$$

Computing the Attractor

Construction of $i-$Attractor $\mathrm{Attr}_0^i(F)$:

$$\begin{aligned}
\mathrm{Attr}_0^0(F) &= F \\
\mathrm{Attr}_0^{i+1}(F) &= \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F)) \\
\\
\mathrm{Attr}_0(F) &= \bigcup_{i=0}^{\infty} \mathrm{Attr}_0^i(F) = \bigcup_{i=0}^{|S|} \mathrm{Attr}_0^i(F)
\end{aligned}$$

Computing the Attractor

Construction of $i-$Attractor $\mathrm{Attr}_0^i(F)$:

$$\begin{aligned}
\mathrm{Attr}_0^0(F) &= F \\
\mathrm{Attr}_0^{i+1}(F) &= \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F)) \\
\mathrm{Attr}_0(F) &= \bigcup_{i=0}^{\infty} \mathrm{Attr}_0^i(F) = \bigcup_{i=0}^{|S|} \mathrm{Attr}_0^i(F)
\end{aligned}$$

Or, in fixpoint notation:

$$\begin{aligned}
f_{(G,F)}(X) &= F \cup \mathrm{CNext}_0(X) \\
\mathrm{Attr}_0(F) &= \mu X. f_{(G,F)} = \mu X. F \cup \mathrm{CNext}_0(X)
\end{aligned}$$

Computing the Attractor

Construction of $i-$Attractor $\mathrm{Attr}_0^i(F)$:

$$\begin{aligned}
\mathrm{Attr}_0^0(F) &= F \\
\mathrm{Attr}_0^{i+1}(F) &= \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F)) \\
\\
\mathrm{Attr}_0(F) &= \bigcup_{i=0}^{\infty} \mathrm{Attr}_0^i(F) = \bigcup_{i=0}^{|S|} \mathrm{Attr}_0^i(F)
\end{aligned}$$

Or, in fixpoint notation:

$$\begin{aligned}
f_{(G,F)}(X) &= F \cup \mathrm{CNext}_0(X) \\
\mathrm{Attr}_0(F) &= \mu X. f_{(G,F)} = \mu X. F \cup \mathrm{CNext}_0(X)
\end{aligned}$$

Claim (1): proved.

Computing the Attractor

Construction of $i-$Attractor $\mathrm{Attr}_0^i(F)$:

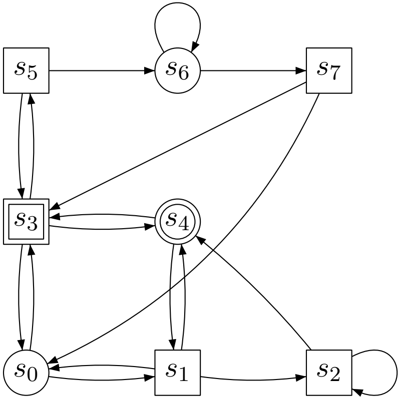
$$
\begin{aligned}
\mathrm{Attr}_0^0(F) &= F \\
\mathrm{Attr}_0^{i+1}(F) &= \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F)) \\
\\
\mathrm{Attr}_0(F) &= \bigcup_{i=0}^{\infty} \mathrm{Attr}_0^i(F) = \bigcup_{i=0}^{|S|} \mathrm{Attr}_0^i(F)
\end{aligned}
$$

Or, in fixpoint notation:

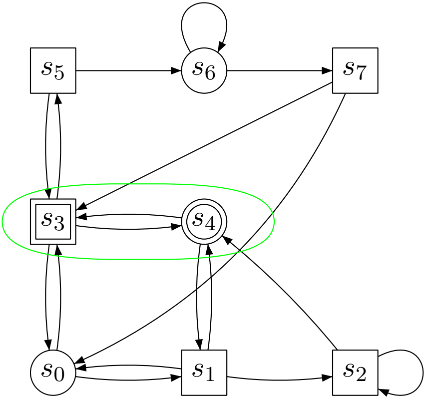
$$
\begin{aligned}
f_{(G,F)}(X) &= F \cup \mathrm{CNext}_0(X) \\
\mathrm{Attr}_0(F) &= \mu X. f_{(G,F)} = \mu X. F \cup \mathrm{CNext}_0(X)
\end{aligned}
$$

Claim (1): proved. Claim (2): $W_0 = \mathrm{Attr}_0(F)$ and $W_1 = S \setminus \mathrm{Attr}_0(F)$

Example

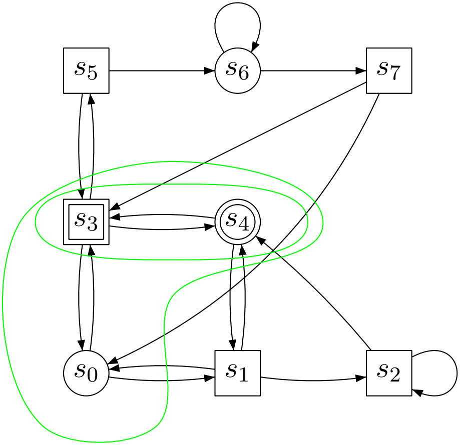


Example



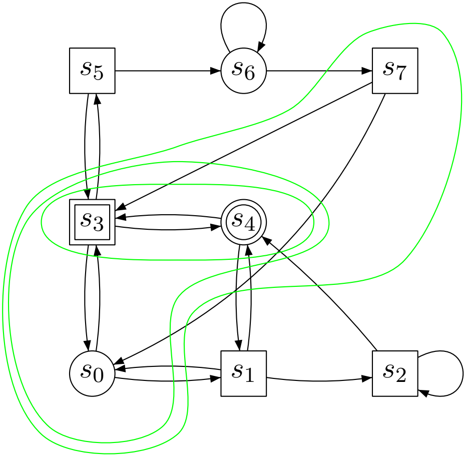
$$\mathrm{Attr}_0^0 = \{s_3, s_4\}$$

Example



$$\mathrm{Attr}_0^0 = \{s_3, s_4\}$$
$$\mathrm{Attr}_0^1 = \{s_0, s_3, s_4\}$$

Example

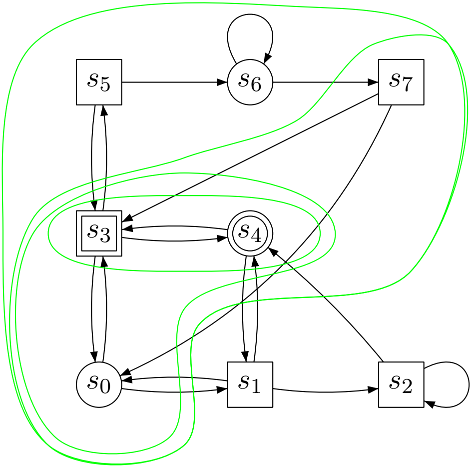


$\mathrm{Attr}_0^0 = \{s_3, s_4\}$

$\mathrm{Attr}_0^1 = \{s_0, s_3, s_4\}$

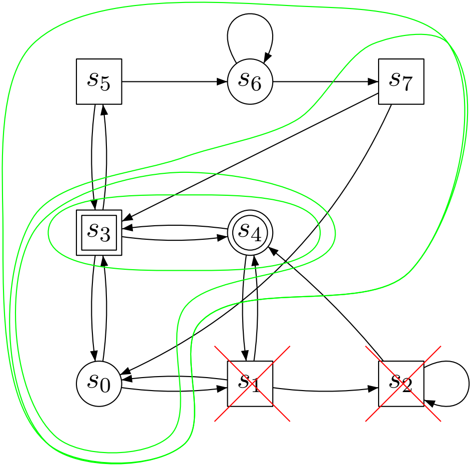
$\mathrm{Attr}_0^2 = \{s_0, s_3, s_4, s_7\}$

Example



$\mathrm{Attr}_0^0 = \{s_3, s_4\}$
$\mathrm{Attr}_0^1 = \{s_0, s_3, s_4\}$
$\mathrm{Attr}_0^2 = \{s_0, s_3, s_4, s_7\}$
$\mathrm{Attr}_0^3 = \{s_0, s_3, s_4, s_6, s_7\}$
$\mathrm{Attr}_0^4 = \{s_0, s_3, s_4, s_5, s_6, s_7\}$
$\mathrm{Attr}_0^5 = \mathrm{Attr}_0^4$

Example



$\mathrm{Attr}_0^0 = \{s_3, s_4\}$
$\mathrm{Attr}_0^1 = \{s_0, s_3, s_4\}$
$\mathrm{Attr}_0^2 = \{s_0, s_3, s_4, s_7\}$
$\mathrm{Attr}_0^3 = \{s_0, s_3, s_4, s_6, s_7\}$
$\mathrm{Attr}_0^4 = \{s_0, s_3, s_4, s_5, s_6, s_7\}$
$\mathrm{Attr}_0^5 = \mathrm{Attr}_0^4$

Duality Between Players

Assume we have a partition of the state space $S = P_0 \cup P_1$ (i.e., $P_0 \cap P_1 = \emptyset$) and we want to prove $W_0 = P_0$ and $W_1 = P_1$.

We need to prove $P_0 \supseteq W_0$, $P_0 \subseteq W_0$, $P_1 \supseteq W_1$, and $P_1 \subseteq W_1$.

Since we know that $W_0 \cap W_1 = \emptyset$ holds, it is sufficient to prove $P_0 \subseteq W_0$ and $P_1 \subseteq W_1$.

$$\begin{aligned} P_0 &\subseteq W_0 \\ S \setminus P_0 &\supseteq S \setminus W_0 \\ P_1 &\supseteq S \setminus W_0 \supseteq W_1 \\ P_1 &\supseteq W_1 \end{aligned} \qquad \begin{aligned} P_1 &\subseteq W_1 \\ S \setminus P_1 &\supseteq S \setminus W_1 \\ P_0 &\supseteq S \setminus W_1 \supseteq W_0 \\ P_0 &\supseteq W_0 \end{aligned}$$

Strategy of Player 0

To show $\mathrm{Attr}_0(F) \subseteq W_0$ and $S \setminus \mathrm{Attr}_0(F) \subseteq W_1$, construct strategies, s.t. Player 0 reaches F in finitely many steps from $\mathrm{Attr}_0(F)$ and Player 1 avoids F for infinitely many steps from $S \setminus \mathrm{Attr}_0(F)$.

Strategy of Player 0

To show $\mathrm{Attr}_0(F) \subseteq W_0$ and $S \setminus \mathrm{Attr}_0(F) \subseteq W_1$, construct strategies, s.t. Player 0 reaches F in finitely many steps from $\mathrm{Attr}_0(F)$ and Player 1 avoids F for infinitely many steps from $S \setminus \mathrm{Attr}_0(F)$.

Proof.

$\mathrm{Attr}_0^i(F) \subseteq W_0$ (by induction over i).

Hypothesis: for every state $s \in \mathrm{Attr}_0^i(F)$ Player 0 has a positional strategy starting from s to reach F in $\leq i$ steps.

- (Base) $s \in \mathrm{Attr}_0^0(F) = F$
- (Induction) $s \in \mathrm{Attr}_0^{i+1}(F) = \mathrm{Attr}_0^i(F) \cup \mathrm{CNext}_0(\mathrm{Attr}_0^i(F))$
  If $s \in \mathrm{Attr}_0^i(F)$, hypothesis applies, else $s \in \mathrm{CNext}_0(\mathrm{Attr}_0^i) \setminus \mathrm{Attr}_0^i$ and Player 0 can force a visit to $\mathrm{Attr}_0^i(F)$ in one step and from where she needs (by induction) at most i steps to reach F.
  So, F is reached after a finite number of moves.

Strategy of Player 1

Proof cont.

$S \setminus \text{Attr}_0(F) \subseteq W_1$

Assume $s \in S \setminus \text{Attr}_0(F)$, then $s \notin \text{CNext}_0(\text{Attr}_0(F))$ and we have two cases:

(a) $s \in S_0 \cap S \setminus \text{Attr}_0(F)$ $\forall s' \in S$: $(s, s') \in E \rightarrow s' \notin \text{Attr}_0(F)$

(b) $s \in S_1 \cap S \setminus \text{Attr}_0(F)$ $\exists s' \in S$: $(s, s') \in E \wedge s' \notin \text{Attr}_0(F)$

In $S \setminus \text{Attr}_0(F)$ Player 1 can choose edges according to (b) leading again to $S \setminus \text{Attr}_0(F)$ and by (a) Player 0 cannot escape from $S \setminus \text{Attr}_0(F)$. So, F will be avoided forever.

$$W_0 = \text{Attr}_0(F) \text{ and } W_1 = S \setminus \text{Attr}_0(F)$$

Safety Games

Given a safety game (G, F) with $G = (S, S_0, E)$, i.e.,

$$\phi_S = \{\rho \in S^\omega \mid \mathrm{Occ}(\rho) \subseteq F\},$$

consider the reachability game $(G, S \setminus F)$, i.e.,

$$\phi_R = \{\rho \in S^\omega \mid \mathrm{Occ}(\rho) \cap (S \setminus F) \neq \emptyset\}.$$

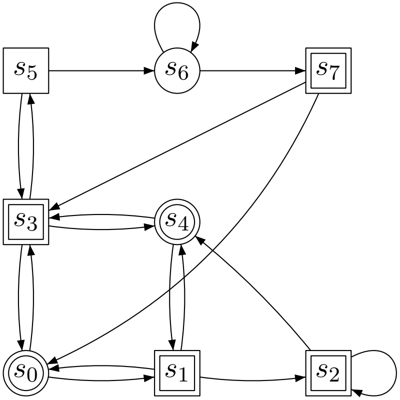
$$\begin{aligned}\text{Then, } S^\omega \setminus \phi_R &= \{\rho \in S^\omega \mid \mathrm{Occ}(\rho) \cap (S \setminus F) = \emptyset\} \\ &= \{\rho \in S^\omega \mid \mathrm{Occ}(\rho) \subseteq F\}.\end{aligned}$$

Player 0 has a safety objective in (G, F).

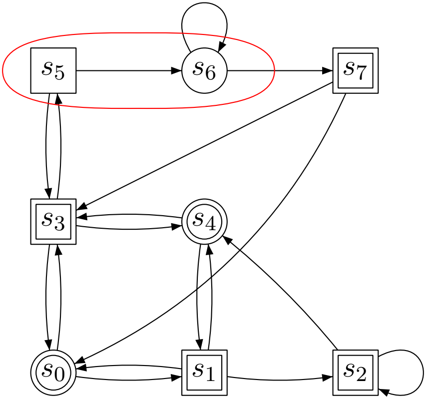
Player 1 has a reachability objective in (G, F).

So, W_0 in the safety game (G, F) corresponds to W_1 in the reachability game $(G, S \setminus F)$.

Example: View of Player 1 (box)

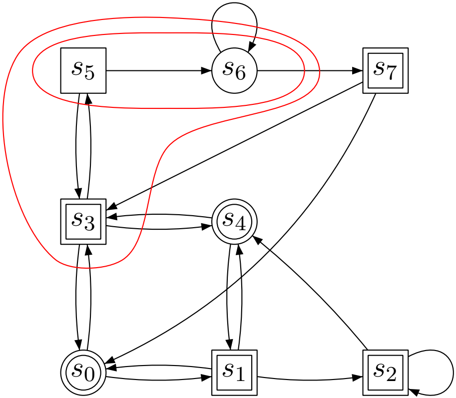


Example: View of Player 1 (box)



$\mathrm{Attr}_1^0 = \{s_5, s_6\} =$ "Bad" states

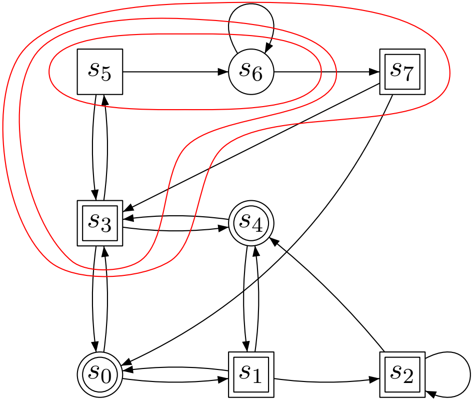
Example: View of Player 1 (box)



$\mathrm{Attr}_1^0 = \{s_5, s_6\}$ = "Bad" states

$\mathrm{Attr}_1^1 = \{s_3, s_5, s_6\}$

Example: View of Player 1 (box)

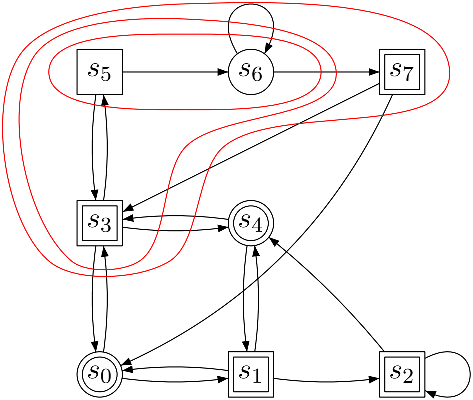


$\mathrm{Attr}_1^0 = \{s_5, s_6\}$ = "Bad" states

$\mathrm{Attr}_1^1 = \{s_3, s_5, s_6\}$

$\mathrm{Attr}_1^2 = \{s_3, s_5, s_6, s_7\}$

Example: View of Player 1 (box)



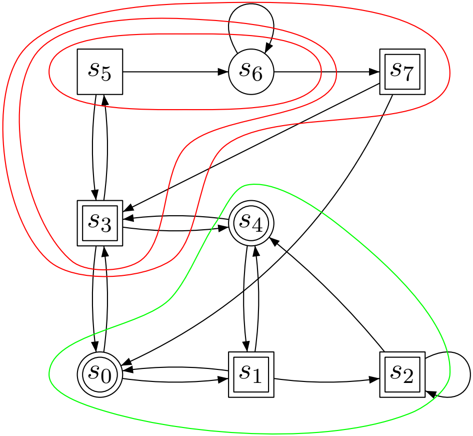
$\mathrm{Attr}_1^0 = \{s_5, s_6\}$ = "Bad" states

$\mathrm{Attr}_1^1 = \{s_3, s_5, s_6\}$

$\mathrm{Attr}_1^2 = \{s_3, s_5, s_6, s_7\}$

$\mathrm{Attr}_1^3 = \mathrm{Attr}_1^2 = \mathrm{Attr}_1$

Example: View of Player 1 (box)



$\mathrm{Attr}_1^0 = \{s_5, s_6\}$ = "Bad" states

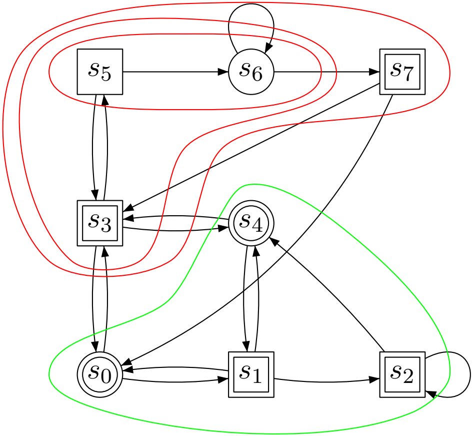
$\mathrm{Attr}_1^1 = \{s_3, s_5, s_6\}$

$\mathrm{Attr}_1^2 = \{s_3, s_5, s_6, s_7\}$

$\mathrm{Attr}_1^3 = \mathrm{Attr}_1^2 = \mathrm{Attr}_1$

$W_0 = S \setminus \mathrm{Attr}_1 = \{s_0, s_1, s_2, s_4\}$

Example: View of Player 1 (box)

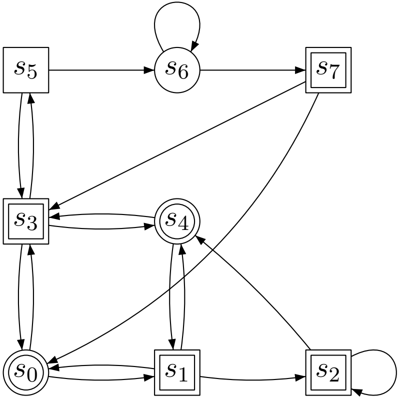


$\text{Attr}_1^0 = \{s_5, s_6\}$ = "Bad" states
$\text{Attr}_1^1 = \{s_3, s_5, s_6\}$
$\text{Attr}_1^2 = \{s_3, s_5, s_6, s_7\}$
$\text{Attr}_1^3 = \text{Attr}_1^2 = \text{Attr}_1$
$W_0 = S \setminus \text{Attr}_1 = \{s_0, s_1, s_2, s_4\}$

Computed using least fixpoint:

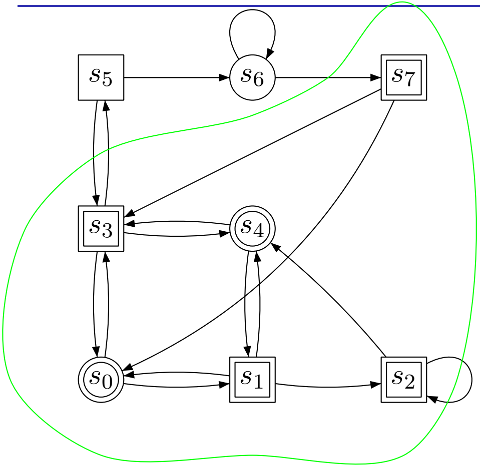
$$\text{Attr}_1(S \setminus F) = \mu X.(S \setminus F) \cup \text{CNext}_1(X)$$

Example: View of Player 0 (circle)



Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

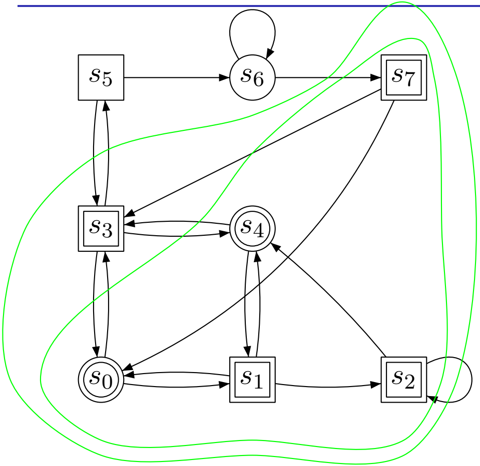
Example: View of Player 0 (circle)



Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

$\text{Safe}_0^0 = \{s_0, s_1, s_2, s_3, s_4, s_7\}$

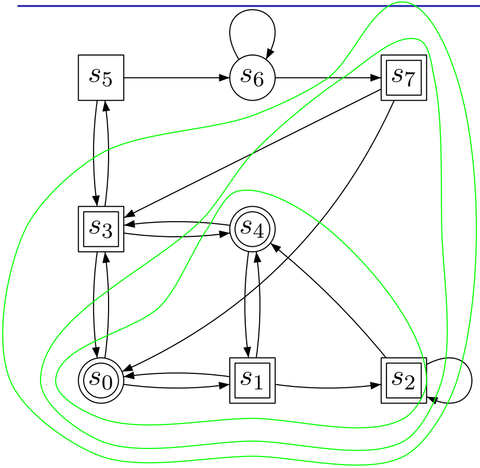
Example: View of Player 0 (circle)



Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

$\text{Safe}_0^0 = \{s_0, s_1, s_2, s_3, s_4, s_7\}$
$\text{Safe}_0^1 = \{s_0, s_1, s_2, s_4, s_7\}$

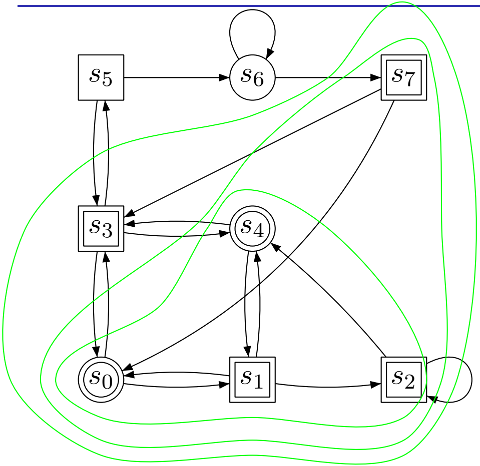
Example: View of Player 0 (circle)



Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

$\text{Safe}_0^0 = \{s_0, s_1, s_2, s_3, s_4, s_7\}$
$\text{Safe}_0^1 = \{s_0, s_1, s_2, s_4, s_7\}$
$\text{Safe}_0^2 = \{s_0, s_1, s_2, s_4\}$

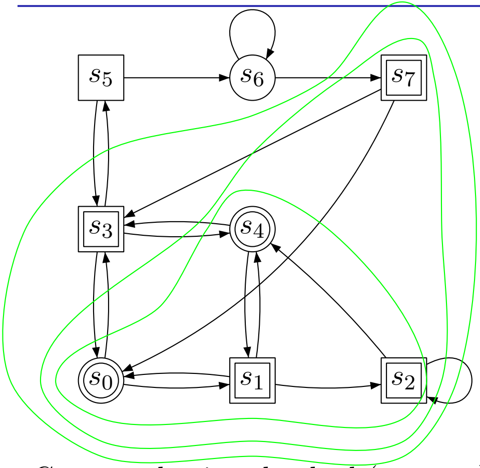
Example: View of Player 0 (circle)



Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

$\text{Safe}_0^0 = \{s_0, s_1, s_2, s_3, s_4, s_7\}$
$\text{Safe}_0^1 = \{s_0, s_1, s_2, s_4, s_7\}$
$\text{Safe}_0^2 = \{s_0, s_1, s_2, s_4\}$
$\text{Safe}_0^3 = \text{Safe}_0^2 = \text{Safe}_0$

Example: View of Player 0 (circle)



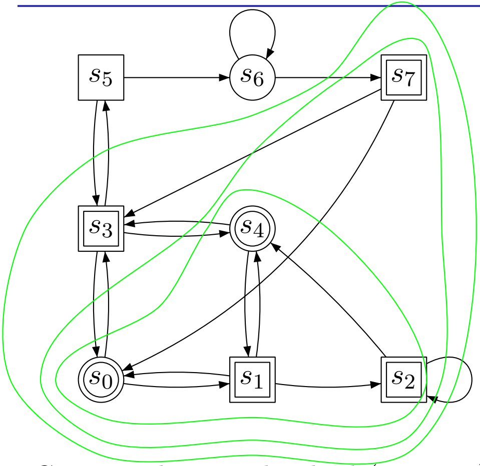
Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

$\text{Safe}_0^0 = \{s_0, s_1, s_2, s_3, s_4, s_7\}$
$\text{Safe}_0^1 = \{s_0, s_1, s_2, s_4, s_7\}$
$\text{Safe}_0^2 = \{s_0, s_1, s_2, s_4\}$
$\text{Safe}_0^3 = \text{Safe}_0^2 = \text{Safe}_0$

Computed using the dual (greatest) fixpoint:

$$\text{Safe}_0(F) = \nu X. \quad F \cap \text{CNext}_0(X)$$

Example: View of Player 0 (circle)



Safe_0^i ... states from which P0 can force at least i-steps within safe reg.

$\text{Safe}_0^0 = \{s_0, s_1, s_2, s_3, s_4, s_7\}$
$\text{Safe}_0^1 = \{s_0, s_1, s_2, s_4, s_7\}$
$\text{Safe}_0^2 = \{s_0, s_1, s_2, s_4\}$
$\text{Safe}_0^3 = \text{Safe}_0^2 = \text{Safe}_0$

Computed using the dual (greatest) fixpoint:

$$\begin{array}{lclccc} \text{Safe}_0(F) & = & \nu X. & F & \cap & \text{CNext}_0(X) \\ \text{Attr}_1(S \setminus F) & = & \mu X. & (S \setminus F) & \cup & \text{CNext}_1(X) \end{array}$$

Summary

We know how to solve reachability and safety games by positional winning strategies.

Reachability = least fixpoint

Safety = greatest fixpoint

The strategies are

- Player 0: Decrease distance to F (called attractor strategy)
- Player 1: Stay outside of $\mathrm{Attr}_0(F)$

In LTL, $\mathsf{eventually}(F)$ = reachability and $\mathsf{always}(F)$ = safety.

Next, $\mathsf{always}(\mathsf{eventually}(F))$ = Büchi and $\mathsf{eventually}(\mathsf{always}(F))$ = Co-Büchi.

Exercise

1. Given a reachability game (G, F) with $G = (S, S_0, E)$ and $F \subseteq Q$, give an algorithm that computes the attractor(F) of Player 0 in time $O(|E|)$.

Exercise

1. Given a reachability game (G, F) with $G = (S, S_0, E)$ and $F \subseteq Q$, give an algorithm that computes the attractor(F) of Player 0 in time $O(|E|)$.

Solution:

1. Preprocessing: Compute for every state $s \in S_1$ outdegree $\text{out}(s)$
2. Set $n(s) := \text{out}(s)$ for each $s \in S_1$
3. To breadth-first search backwards from F with the following conventions
   - mark all $s \in F$
   - mark $s \in S_0$ if reached from marked state
   - mark $s \in S_1$ if $n(s) = 0$, other set $n(s) := n(s) - 1$.

The marked vertices are the ones of $\text{Attr}_0(F)$.