Infinite Games

Barbara Jobstmann
Verimag/CNRS (Grenoble, France)

November 6, 2009
Motivation: Build Correct HW/SW Systems

- Use logic to specify correctness properties, e.g.:
  - every job sent to the printer is eventually printed
  - two jobs do not overlap (only one job is printed at a time)
  - a job that is canceled will be interrupted

These are conditions on infinite sequences (system runs), and can be specified by automata and logical formulas.
Motivation: Build Correct HW/SW Systems

- Use **logic** to specify correctness properties, e.g.:
  - every job sent to the printer is eventually printed
  - two jobs do not overlap (only one job is printed at a time)
  - a job that is canceled will be interrupted

These are conditions on infinite sequences (system runs), and can be specified by automata and logical formulas.

- Given a **logical specification**, we can do either:
  - **VERIFICATION**: prove that a given system satisfies the specification
  - **SYNTHESIS**: build a system that satisfies the specification
Intuition of Infinite Games

Two players:

1. Printer controller is Player 0
2. Users are Player 1

A play of a game is an infinite sequence of states of printer transition system, where the two players choose moves alternatively.

Player 0 (printer controller) wins the play if all conditions are satisfied independent of the choices Player 1 (user) makes. This corresponds to finding a winning strategy for Player 0 in an infinite game.
Our Aim

Solution of the Synthesis Problem

1. Decide whether there exists such a winning strategy - Realizability Problem
2. If “yes”, then construct the system - Synthesis Problem
Our Aim

Solution of the Synthesis Problem

1. Decide whether there exists such a winning strategy - Realizability Problem

2. If “yes”, then construct the system - Synthesis Problem

Main result:

The synthesis problem is algorithmically solvable for finite-state systems with respect to specifications given as \( \omega \)-automata or linear-time temporal logic.
Other Applications of Games

- Program repair or program sketching
- Nicer and more intuitive proofs for logics over trees
- Verification for logics over trees
Outline

1. Terminology
2. Safety and Reachability games
3. Büchi and coBüchi games
Terminology
Terminology

Two-player games between Player 0 and 1

An infinite game \( \langle G, \phi \rangle \) consists of

- a game graph \( G \) and
- a winning condition \( \phi \).

\( G \) defines the “playground”, in which the two players compete. \( \phi \) defines which plays are won by Player 0. If a play does not satisfy \( \phi \), then Player 1 wins on this play.
Game Graphs

A game graph is a tuple $G = \langle S, S_0, T \rangle$ where:

- $S$ is a finite set of states,
- $S_0 \subseteq S$ is the set of Player-0 states ($S_1 = S \setminus S_0$ are the Player-1 states),
- $T \subseteq S \times S$ is a transition relation. We assume that each state has at least one successor.
Plays

A play is an infinite sequence of states $\rho = s_0s_1s_2\cdots \in S^\omega$ such that for all $i \geq 0 \langle s_i, s_{i+1} \rangle \in T$.

It starts in $s_0$ and it is built up as follows:

If $s_i \in S_0$, then Player 0 chooses an edge starting in $s_i$, otherwise Player 1 picks such an edge.

Intuitively, a token is moved from state to state via edges: From $S_0$-states Player 0 moves the token, from $S_1$-states Player 1 moves the token.
Winning Condition

The winning condition describes the plays won by Player 0. A winning condition or winning objective $\phi$ is a subset of plays, i.e., $\phi \subseteq S^\omega$.

We use logical conditions (e.g., LTL formulas) or automata theoretic acceptance conditions to describe $\phi$.

Example:

- $\Box \Diamond s$ for some state $s \in S$
- All plays that stay within a safe region $F \subseteq S$ are in $\phi$.
- Given a priority function $p : S \rightarrow \{0, 1, \ldots, d\}$, all plays in which the smallest priority visited is even.

Games are named after their winning condition, e.g., Safety game, Reachability game, LTL game, Parity game,...
Types of Games

Given a play $\rho$, we define

- $\text{Occ}(\rho) = \{ s \in S \mid \exists i \geq 0 : s_i = s \}$
- $\text{Inf}(\rho) = \{ s \in S \mid \forall i \geq 0 \exists j > i : s_j = s \}$

Given a set $F \subseteq S$,

Reachability Game $\phi = \{ \rho \in S^\omega \mid \text{Occ}(\rho) \cap F \neq \emptyset \}$

Safety Game $\phi = \{ \rho \in S^\omega \mid \text{Occ}(\rho) \subseteq F \}$

Büchi Game $\phi = \{ \rho \in S^\omega \mid \text{Inf}(\rho) \cap F \neq \emptyset \}$

Co-Büchi Game $\phi = \{ \rho \in S^\omega \mid \text{Inf}(\rho) \subseteq F \}$
Types of Games

Given a priority function $p : S \rightarrow \{0, 1, \ldots, d\}$ or an LTL formula $\varphi$

- **Weak-Parity Game** $\phi = \{\rho \in S^\omega \mid \min_{s \in \text{Occ}(\rho)} p(s) \text{ is even} \}$
- **Parity Game** $\phi = \{\rho \in S^\omega \mid \min_{s \in \text{Inf}(\rho)} p(s) \text{ is even} \}$
- **LTL Game** $\phi = \{\rho \in S^\omega \mid \rho \models \varphi \}$

We will refer to the type of a game and give $F$, $p$, or $\varphi$ instead of defining $\phi$.

We will also talk about Muller and Rabin games.
Strategies

A strategy for Player 0 from state $s$ is a function

$$f : S^* S_0 \rightarrow S$$

specifying for any sequence of states $s_0, s_1, \ldots s_k$ with $s_0 = s$ and $s_k \in S_0$ a successor state $s_j$ such that $(s_k, s_j) \in T$.

A play $\rho = s_0 s_1 \ldots$ is compatible with strategy $f$ if for all $s_i \in S_0$ we have that $s_{i+1} = f(s_0 s_1 \ldots s_i)$.

(Definitions for Player 1 are analogous.)

Given strategies $f$ and $g$ from $s$ for Player 0 and 1, respectively. We denote by $G_{f,g}$ the (unique) play that is compatible with $f$ and $g$. 
Winning Strategies and Regions

Given a game \((G, \phi)\) with \(G = (S, S_0, E)\), a strategy \(f\) for Player 0 from \(s\) is called a \textit{winning strategy} if for all Player-1 strategies \(g\) from \(s\), if \(G_{f,g} \in \phi\) holds. Analogously, a Player-1 strategy \(g\) is winning if for all Player-0 strategies \(f\), \(G_{f,g} \not\in \phi\) holds.

Player 0 (resp. 1) wins from \(s\) if s/he has a winning strategy from \(s\).
Winning Strategies and Regions

Given a game \((G, \phi)\) with \(G = (S, S_0, E)\), a strategy \(f\) for Player 0 from \(s\) is called a winning strategy if for all Player-1 strategies \(g\) from \(s\), if \(G_{f,g} \in \phi\) holds. Analogously, a Player-1 strategy \(g\) is winning if for all Player-0 strategies \(f\), \(G_{f,g} \notin \phi\) holds.

Player 0 (resp. 1) wins from \(s\) if s/he has a winning strategy from \(s\). The winning regions of Player 0 and 1 are the sets

\[
W_0 = \{ s \in S \mid \text{Player 0 wins from } s \}
\]

\[
W_1 = \{ s \in S \mid \text{Player 1 wins from } s \}
\]

Note each state \(s\) belongs at most to \(W_0\) or \(W_1\). Otherwise pick winning strategies \(f\) and \(g\) from \(s\) for Player 0 and 1, respectively, then \(G_{f,g} \in \phi\) and \(G_{f,g} \notin \phi\): Contradiction.
Questions About Games

Solve a game \((G, \phi)\) with \(G = (S, S_0, T)\):

1. Decide for each state \(s \in S\) if \(s \in W_0\).
2. If yes, construct a suitable winning strategy from \(s\).

Further interesting question:

- Optimize construction of winning strategy (e.g., time complexity)
- Optimize parameters of winning strategy (e.g., size of memory)
Safety game \((G, F)\) with \(F = \{s_0, s_1, s_3, s_4\}\), i.e., \(\text{Occ}(\rho) \subseteq F\)

A winning strategy for Player 0 (from state \(s_0, s_3, \text{and } s_4\)):
- From \(s_0\) choose \(s_3\) and from \(s_4\) choose \(s_3\)

A winning strategy for Player 1 (from state \(s_1\) and \(s_2\)):
- From \(s_1\) choose \(s_2\), from \(s_2\) choose \(s_4\), and from \(s_3\) choose \(s_4\)
Safety game \((G, F)\) with \(F = \{s_0, s_1, s_3, s_4\}\), i.e., \(\text{Occ}(\rho) \subseteq F\).

A winning strategy for Player 0 (from state \(s_0\), \(s_3\), and \(s_4\)):

- From \(s_0\) choose \(s_3\) and from \(s_4\) choose \(s_3\)

A winning strategy for Player 1 (from state \(s_1\) and \(s_2\)):

- From \(s_1\) choose \(s_2\), from \(s_2\) choose \(s_4\), and from \(s_3\) choose \(s_4\)

\(W_0 = \{s_0, s_3, s_4\}\), \(W_1 = \{s_1, s_2\}\)
LTL game \((G, \varphi)\) with \(\varphi = \Diamond s_0 \land \Diamond s_4\) (visit \(s_0\) and \(s_4\))

Winning strategy for Player 0 from \(s_0\):

- From \(s_0\) to \(s_3\), from \(s_3\) to \(s_4\), and from \(s_4\) to \(s_1\).

Note: this strategy is not winning from \(s_3\) or \(s_4\).

Winning strategy for Player 0 from \(s_3\):

- From \(s_0\) to \(s_3\), from \(s_4\) to \(s_3\), and from \(s_3\) to \(s_0\) on first visit, otherwise to \(s_4\).
Determinacy

Recall: the winning regions are disjoint, i.e., $W_0 \cap W_1 = \emptyset$

Question: Is every state winning for some player?

A game $(G, \phi)$ with $G = (S, S_0, E)$ is called determined if $W_0 \cup W_1 = S$ holds.

Remarks:

1. We will show that all automata theoretic games we consider here are determined.

2. There are games which are not determined (e.g., Tic-Tac-Toe)
Strategy Types

In general, a strategy is a function $f : S^+ \rightarrow S$.

1. **Computable or recursive strategies**: $f$ is computable

2. **Finite-state strategies**: $f$ is computable with a finite-state automaton meaning that $f$ has bounded information about the past (history).

3. **Memoryless or positional strategies**: $f$ only depends on the current state of the game (no knowledge about history of play)
Positional Strategies

Given a game \((G, \phi)\) with \(G = (S, S_0, E)\), a strategy \(f : S^+ \to S\) is called positional or memoryless if for all words \(w, w' \in S^+\) with \(w = s_0 \ldots s_n\) and \(w' = s'_0 \ldots s'_m\) such that \(s_n = s'_m\), \(f(w) = f(w')\) holds.

A positional strategy for Player 0 is representable as

1. a function \(f : S_0 \to S\)

2. a set of edges containing for every Player-0 state \(s\) exactly one edge starting in \(s\) (and for every Player-1 state \(s'\) all edges starting in \(s'\))
Finite-state Strategies

A strategy automaton over a game graph $G = (S, S_0, E)$ is a finite-state automaton $A = (M, m_0, \delta, \lambda)$ with alphabet $S$, where

- $M$ is a finite set of states (called memory),
- $m_0 \in M$ is an initial state (the initial memory content),
- $\delta : M \times S \to M$ is a transition function (the memory update fct),
- $\lambda : M \times S \to S$ is a labeling function (called the choice function).
Finite-state Strategies

A strategy automaton over a game graph $G = (S, S_0, E)$ is a finite-state automaton $A = (M, m_0, \delta, \lambda)$ with alphabet $S$, where

- $M$ is a finite set of states (called memory),
- $m_0 \in M$ is an initial state (the initial memory content),
- $\delta : M \times S \rightarrow M$ is a transition function (the memory update fct),
- $\lambda : M \times S \rightarrow S$ is a labeling function (called the choice function).

The strategy for Player 0 computed by $A$ is the function

$$f_A(s_0 \ldots s_k) := \lambda(\delta(m_0, s_0 \ldots s_{k-1}), s_k) \text{ with } s_k \in S_0$$

and the usual extension of $\delta$ to words: $\delta(m_0, \epsilon) = m_0$ and $\delta(m_0, s_0 \ldots s_k) = \delta(\delta(m_0, s_0 \ldots s_{k-1}), s_k)$. Any strategy $f$, such that there exists an $A$ with $f_A = f$, is called finite-state strategy.
Recall Example

Objective: visit $s_0$ and $s_4$, i.e, $\{s_0, s_4\} \subseteq \text{Occ}(\rho)$

Winning strategy for Player 0 from $s_0$, $s_3$ and $s_4$:

- From $s_0$ to $s_3$, from $s_4$ to $s_3$, and from $s_3$ to $s_0$ on first visit, otherwise to $s_4$. 

$$
\begin{align*}
&\text{Transition probabilities:} \\
&s_0/s_3 \quad s_1/s_4 \quad s_2/s_3 \\
&s_1/s_4 \\
&s_2/s_3 \\
&s_3/s_4 \\
&s_4/s_3 \\
&\text{McCulloch-Pitts model:} \\
&m_0 \quad m_1 \\
\end{align*}
$$
Extended Game
Extended Game

\[
\begin{align*}
&\text{Extended Game} \\
&m_0 \\
&\text{Extended Game} \\
&\text{Extended Game}
\end{align*}
\]
Extended Game

- $s_0/s_3$
- $s_1/-$
- $s_2/-$
- $s_4/s_3$

Diagram:

$m_0$

$s_0$ — $s_1$ — $s_2$ — $s_3$ — $s_4$ — $s_0$

$m_1$

$s_0$ — $s_1$ — $s_2$ — $s_3$ — $s_4$ — $s_0$
Extended Game

Note: the strategy in the extended game graph is memoryless.
Reachability and Safety Games
Reachability and Safety Games

Theorem

Given a reachability game \((G, F)\) with \(G = (S, S_0, E)\) and \(F \subseteq S\), then the winning regions \(W_0\) and \(W_1\) of Player 0 and 1, respectively, are computable, and both players have corresponding memoryless winning strategies.

Proof.

Define

\[
\text{Attr}^i_0(F) := \{ s \in S \mid \text{Player 0 can force a visit from } s \text{ to } F \text{ in less than } i \text{ moves}\}
\]
Example
Example

\[ \text{Attr}_0^0 = \{s_3, s_4\} \]
Example

\[
\text{Attr}_0^0 = \{s_3, s_4\} \\
\text{Attr}_0^1 = \{s_0, s_3, s_4\}
\]
Example

\begin{align*}
\text{Attr}_0^0 &= \{s_3, s_4\} \\
\text{Attr}_0^1 &= \{s_0, s_3, s_4\} \\
\text{Attr}_0^2 &= \{s_0, s_3, s_4, s_7\}
\end{align*}
Example

\[
\begin{align*}
\text{Attr}_0^0 &= \{s_3, s_4\} \\
\text{Attr}_0^1 &= \{s_0, s_3, s_4\} \\
\text{Attr}_0^2 &= \{s_0, s_3, s_4, s_7\} \\
\text{Attr}_0^3 &= \{s_0, s_3, s_4, s_6, s_7\} \\
\text{Attr}_0^4 &= \{s_0, s_3, s_4, s_5, s_6, s_7\}
\end{align*}
\]
Example

$\text{Attr}^0_0 = \{s_3, s_4\}$
$\text{Attr}^1_0 = \{s_0, s_3, s_4\}$
$\text{Attr}^2_0 = \{s_0, s_3, s_4, s_7\}$
$\text{Attr}^3_0 = \{s_0, s_3, s_4, s_6, s_7\}$
$\text{Attr}^4_0 = \{s_0, s_3, s_4, s_5, s_6, s_7\}$
Computing the Attractor

Construction of $\text{Attr}^i_0(F)$:

$\text{Attr}^0_0(F) = F$

$\text{Attr}^{i+1}_0(F) = \text{Attr}^i_0(F) \cup$

$\{s \in S_0 \mid \exists s' \in S : (s, s') \in E \land s' \in \text{Attr}^i_0(F)\} \cup$

$\{s \in S_1 \mid \forall s' \in S : (s, s') \in E \rightarrow s' \in \text{Attr}^i_0(F)\}$

Then

$\text{Attr}^0_0(F) \subseteq \text{Attr}^1_0(F) \subseteq \text{Attr}^2_0(F) \subseteq \ldots$ and since $S$ is finite, there exists $k \leq |S|$ s.t. $\text{Attr}^k_0(F) = \text{Attr}^{k+1}_0(F)$.

The 0-Attractor is defined as:

$$\text{Attr}_0(F) := \bigcup_{i=0}^{|S|} \text{Attr}^i_0(F)$$
0-Attractor

To show $W_0 = \text{Attr}_0(F)$ and $W_1 = S \setminus \text{Attr}_0(F)$, we construct winning strategies for Player 0 and 1. Define distance from state $s$ to $F$:

$$d(s, F) := \begin{cases} 
\min \{ i \mid s \in \text{Attr}_0^i(F) \} & \text{if } s \in \text{Attr}_0(F), \\
\infty & \text{otherwise}.
\end{cases}$$
0-Attractor

To show $W_0 = \text{Attr}_0(F)$ and $W_1 = S \setminus \text{Attr}_0(F)$, we construct winning strategies for Player 0 and 1. Define distance from state $s$ to $F$:

$$d(s, F) := \begin{cases} 
\min\{i \mid s \in \text{Attr}_i^0(F)\} & \text{if } s \in \text{Attr}_0(F), \\
\infty & \text{otherwise}.
\end{cases}$$

Proof.

$\text{Attr}_0(F) \subseteq W_0$

(a) $\forall s \in S_0 \cap \text{Attr}_0(F) \setminus F \exists s' \in S: (s, s') \in E \land d(s', F) < d(s, F)$

(b) $\forall s \in S_1 \cap \text{Attr}_0(F) \setminus F, \forall s' \in S: (s, s') \in E \land d(s', F) < d(s, F)$

In $\text{Attr}_0(F) \setminus F$, Player 0 can decrease distance by picking edges according to (a) and Player 1 cannot avoid decreasing the distance because of (b). So, $F$ is reached after a finite number of moves.
0-Attractor cont.

Proof cont.

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

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

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

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.,

\[
\phi_S = \{ \rho \in S^\omega \mid \text{Occ}(\rho) \subseteq F \},
\]

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

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

Then, \(S^\omega \setminus \phi_R = \{ \rho \in S^\omega \mid \text{Occ}(\rho) \cap (S \setminus F) = \emptyset \}
\]
\[= \{ \rho \in S^\omega \mid \text{Occ}(\rho) \subseteq F \}.\]

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 \((G, S \setminus F)\).
Homework

Given a reachability game \((G, F)\) with \(G = (S, S_0, E)\), find an algorithm that computes the winning regions and strategies in time \(O(|E|)\)-time.
Summary

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

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

In LTL, $\Diamond F = \text{reachability}$ and $\Box F = \text{safety}$.

Next, $\Box \Diamond F = \text{B"uchi}$ and $\Diamond \Box F = \text{Co-B"uchi}$. 
Büchi and Co-Büchi Games
Büchi Game

Given a Büchi game \((G, F)\) over the game graph \(G = (S, S_0, E)\) with the set \(F \subseteq S\) of Büchi states, we aim to

- determine the winning regions of Player 0 and 1
- compute their respective winning strategies

Recall, Player 0 wins \(\rho\) iff she visits infinitely often states in \(F\), i.e.,
\[
\phi = \{\rho \in S^\omega \mid \inf(\rho) \cap F \neq \emptyset\}.
\]
Idea

Compute for \( i \geq 1 \) the set \( \text{Recur}_0^i \) of states \( s \in F \) from which Player 0 can force at least \( i \) revisits to \( F \).

Then,

\[
F \supseteq \text{Recur}_0^1(F) \supseteq \text{Recur}_0^2(F) \supseteq \ldots
\]

We compute the winning region of Player 0 with

\[
\text{Recur}_0(F) := \bigcap_{i \leq 1} \text{Recur}_0^i(F)
\]

Again, since \( F \) is finite, there exists \( k \) such that

\[
\text{Recur}_0(F) = \text{Recur}_0^k(F).
\]

Claim: \( W_0 = \text{Attr}_0(\text{Recur}_0(F)) \)
One-Step Attractor

We count revisits, so we need the set of states from which Player 0 can force a revisit to $F$, i.e., state from which she can force a visit in $\geq 1$ steps.

We define a slightly modified attractor:

$$A^0_0 = \emptyset$$

$$A^i_{i+1} = A^i_0 \cup \{s \in S_0 | \exists s' \in S : (s, s') \in E \land s' \in A^i_0 \cup F\} \cup \{s \in S_1 | \forall s' \in S : (s, s') \in E \rightarrow s' \in A^i_0 \cup F\}$$

$$\text{Attr}^+_0(F) = \bigcup_{i \geq 0} A^i_0$$

$\text{Attr}^+_0(F)$ is the set of states from which Player 0 can force a revisit to $F$. 
Visit versus Revisit

\[ \text{Visit: } F \rightarrow \text{Attr}_0(F) \]

\[ \text{Revisit: } F \rightarrow \text{Attr}_0^+(F) \]
Recurrence Set

We define

\[
\text{Recur}_0^0(F) := F \\
\text{Recur}_0^{i+1}(F) := F \cap \text{Attr}_0^+(\text{Recur}_0^i(F)) \\
\text{Recur}_0(F) := \bigcap_{i \geq 0} \text{Recur}_0^i(F)
\]

We show that there exists \( k \) such that \( \text{Recur}_0(F) := \bigcap_{i \geq 0} \text{Recur}_0^i(F) \) by proving \( \text{Recur}_0^{i+1}(F) \subseteq \text{Recur}_0^i(F) \) for all \( i \geq 0 \).

Proof.

\begin{itemize}
  \item \( i = 0 \): \( F \cap \text{Attr}_0^+(F) \subseteq F \)
  \item \( i \rightarrow i + 1 \):
    \[
    \text{Recur}_0^{i+2}(F) = F \cap \text{Attr}_0^+(\text{Recur}_0^{i+1}(F)) \subseteq F \cap \text{Attr}_0^+(\text{Recur}_0^i(F)) \\
    = \text{Recur}_0^{i+1}(F)
    \]
\end{itemize}
Recurrence Set cont.

We show that all states in $\text{Attr}_0(\text{Recur}_0(F))$ are winning for Player 0, i.e., $\text{Attr}_0(\text{Recur}_0(F)) \subseteq W_0$. We construct a memoryless winning strategy for Player 0 for all states in $\text{Attr}_0(\text{Recur}_0(F))$.

Proof.

We know that there exists $k$ such that 
$\text{Recur}^k_0(F) = F \cap \text{Attr}^+_0(\text{Recur}^k_0(F))$. So,

- for $s \in \text{Recur}^k_0(F) \cap S_0$ Player 0 can choose an edges back to $\text{Attr}^+_0(\text{Recur}^k_0(F))$ and
- for $s \in \text{Recur}^k_0(F) \cap S_1$ all edges lead back to $\text{Attr}^+_0(\text{Recur}^k_0(F))$.

For all states in $\text{Attr}_0(\text{Recur}_0(F)) \setminus \text{Recur}_0(F)$, Player 0 can follow the attractor strategy to reach $\text{Recur}_0(F)$.
We show $S \setminus \text{Attr}_0(\text{Recur}_0(F)) \subseteq W_1$.

Proof.

Show: Player 1 can force $\leq i$ visits to $F$ from $s \notin \text{Attr}_0(\text{Recur}_0^i(F))$.

$i = 0$: $s \notin \text{Attr}_0(F)$, so Player 1 can avoid visiting $F$ at all.

$i \rightarrow i + 1$: $s \notin \text{Attr}_0(\text{Recur}_0^{i+1}(F))$.

- $s \notin \text{Attr}_0(\text{Recur}_0^i(F))$, Player 1 plays according to ind. hypotheses.
- Otherwise, $s \in \text{Attr}_0(\text{Recur}_0^i(F)) \setminus \text{Attr}_0(\text{Recur}_0^{i+1}(F))$ and Player 1 can avoid $\text{Attr}_0(\text{Recur}_0^{i+1}(F))$. In particular, $s \notin \text{Recur}_0^{i+1}(F) = F \cap \text{Attr}_0^+(\text{Recur}_0^i(F))$.

- If $s \in \text{Recur}_0^i$, then Player 1 can force to leave $\text{Attr}_0^+(\text{Recur}_0^i(F))$, otherwise $s \in \text{Recur}_0^{i+1}(F)$. (So, by ind. hyp. at most $i + 1$ visits.)
- If $s \in \text{Attr}_0(\text{Recur}_0^i(F)) \setminus \text{Recur}_0^i(F)$, avoid $\text{Attr}_0(\text{Recur}_0^{i+1}(F))$.\)
Recurrence Set cont.
Büchi games

We have shown that Player 0 has a (memoryless) winning strategy in $\text{Attr}_0(\text{Recur}_0(F))$, so $\text{Attr}_0(\text{Recur}_0(F)) \subseteq W_0$. And, Player 1 has a (memoryless) winning strategy in $S \setminus \text{Attr}_0(\text{Recur}_0(F))$, so $S \setminus \text{Attr}_0(\text{Recur}_0(F)) \subseteq W_1$. This implies the following theorem.

Theorem

*Given a Büchi game $((S, S_0, E), F)$, the winning regions $W_0$ and $W_1$ are computable and form a partition, i.e., $W_0 \cup W_1 = S$. Both players have memoryless winning strategies.*
Co-Büchi Games

Given a Co-Büchi Game \( ((S, S_0, E), F) \), i.e.,

\[
\phi_C = \{ \rho \in S^\omega \mid \text{Inf}(\rho) \subseteq F \}
\]

consider the Büchi Game \( ((S, S_0, E), S \setminus F) \), i.e,

\[
\phi_B = \{ \rho \in S^\omega \mid \text{Inf}(\rho) \cap S \setminus F \neq \emptyset \}.
\]

Then, \( S^\omega \setminus \phi_B = \{ \rho \in S^\omega \mid \text{Inf}(\rho) \cap (S \setminus F) = \emptyset \} \)
\[= \{ \rho \in S^\omega \mid \text{Inf}(\rho) \subseteq F \} \]

Player 0 has a co-Büchi objective in \( (G, F) \).
Player 1 has a Büchi objective in \( (G, F) \).

So, \( W_0 \) in the co-Büchi game \( (G, F) \) corresponds to \( W_1 \) in the Büchi game \( (G, S \setminus F) \).
Summary

We know how to solve Büchi and Co-Büchi games by positional winning strategies.

In LTL,

- □ ◇ F = reachability
- □ F = safety
- □ ◇ F = Büchi
- ◇ ◇ F = Co-Büchi

Next, Muller and Parity games.