An Quick Introduction
to Controller Synthesis

Moves - April 20, 2007

JF Raskin
U.L.B.
Context

- Make a model of the environment

Environment

- Make clear the control objective: **Bad**

- Make a model of your control strategy: **ControllerMod**

- Verify:

  Does Environment $\parallel$ ControllerMod avoid **Bad**?
Context

• Make a model of the environment

Environment

• Make clear the control objective:

Bad

Make the algorithmic synthesis of Controller

• Verify:

Does Environment \parallel ControllerMod avoid Bad?
The synthesis problem
The synthesis problem

\[ ? \parallel \text{Env} \models \phi \]
The synthesis problem

\[ ? \parallel \text{Env} \models \phi \]
The synthesis problem

\[ \text{Env} \models \phi \]

Using algorithmic methods
The synthesis problem

Specialize process A into C such that

\[ A \geq C \text{ and } C \parallel B \models \phi \]

So, C must refine A and control B to enforce \( \phi \)
Basic technics: finite state case
Are transition systems adequate for synthesis?

- For the verification problem, the semantics of processes is usually given by transition systems.
- When we consider the transition system for $A || B$, we lose the information about the components.
For the verification problem, the semantics of processes is usually given by transition systems.

When we consider the transition system for $A || B$, we lose the information about the components.

So, we need richer models where identities of processes are explicit: two-player game structures.
Two-player game structures
Rounded positions belong to Player I
Rounded positions belong to Player 1

Square positions belong to Player 2
A game is played as follows: in each round, the game is in a position, if the game is in a rounded position, Player 1 resolves the choice for the next state, if the game is in a square position, Play 2 resolves the choice. The game is played for an infinite number of rounds.
Play : 0000
Play : 0000 0100
Play: 0000 0100 0101
Play : 0000 0100 0101 1101 1110 1111
Play : 0000 0100 0101 1101 ...
Two-player Game Structure

A **two-player game structure** is a tuple
\[ G = \langle Q_1, Q_2, \iota, \delta \rangle \]
where:

- \( Q_1 \) and \( Q_2 \) are two (finite and) disjoint sets of positions
- \( \iota \in Q_1 \cup Q_2 \) is the **initial** position of the game
- \( \delta \subseteq (Q_1 \cup Q_2) \times (Q_1 \cup Q_2) \) is the **transition relation** of the game

We assume that
\[ \forall q \in Q_1 \cup Q_2 : \exists q' \in Q_1 \cup Q_2 : \delta(q, q') \]
Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$, 

$w = q_0 q_1 \ldots q_n \ldots$ is a play in $G$ if
Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$, 

$w = q_0q_1 \ldots q_n \ldots$ is a play in $G$ if

$\forall i \geq 0 : q_i \in Q_1 \cup Q_2$
Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$,

$w = q_0 q_1 \ldots q_n \ldots$ is a **play** in $G$ if

Notations

Let $w = q_0 q_1 \ldots q_n \ldots$:

- $w(i)$ denotes position $i$
- $w(0, i)$ denotes the prefix up to position $i$
- $\text{last}(w(0, i)) = w(i)$
Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$,

$w = q_0 q_1 \ldots q_n \ldots$ is a play in $G$ if

1) $w(0) = \iota$
2) $\forall i \geq 0 : \delta(w(i), w(i + 1))$

We denote the set of plays in $G$ by: $\text{Plays}(G)$ and

$$\text{PrefPlays}(G) = \{q_0q_1 \ldots q_n \mid \exists w \in \text{Plays}(G) \land \forall 1 \leq i \leq n : w(i) = q_i \}$$

$$\text{PrefPlays}_k(G) = \{w \in \text{PrefPlays}(G) \land \text{last}(w) \in Q_k \}$$
Who is winning?

Play: 0000 0100 0101 1101 ...
Who is winning?

Play: 0000 0100 0101 1101 ...

Is this a **good** or a **bad** play for **Player k**?
Who is winning?

A winning condition (for Player $k$) is a set of plays

$$W \subseteq (Q_1 \cup Q_2)$$
Game
= 
Two-player game structure 
+ 
Winning condition for Player $k$
Players are playing **according to strategies.**

A **Player $k$ strategy** in $G$ is a function:

$$\lambda : \text{PrefPlays}_k(G) \rightarrow Q_1 \cup Q_2$$

with the restriction that:

$$\forall w \in \text{PrefPlays}_k(G) : \delta(\text{last}(w), \lambda(w))$$
Outcome of a strategy

\( w \) is a possible \textbf{outcome} of the Player \( k \) strategy \( \lambda \) if

\[
\forall i \geq 0 : w(i) \in Q_k : w(i + 1) = \lambda(w(0, i))
\]

\( w \) is a play where Player \( k \) plays according to strategy \( \lambda \)
Outcome of a strategy

\( w \) is a possible **outcome** of the Player \( k \) strategy \( \lambda \) if

\[
\forall i \geq 0 : w(i) \in Q_k : w(i + 1) = \lambda(w(0, i))
\]

The set of plays that have this property is denoted

\[ \text{Outcome}_k(G, \lambda) \]
Winning strategy

• Given a pair \((G, W)\)

• We say that Player \(k\) wins the game \((G, W)\) if and only if:

\[ \exists \lambda : \text{Outcome}_k(G, \lambda) \subseteq W \]
Winning strategy

• Given a pair \((G, W)\)

• We say that Player \(k\) wins the game \((G, W)\)
  if and only if:

\[
\exists \lambda : \text{Outcome}_k(G, \lambda) \subseteq W
\]

That is, no matter how the other player resolves his choices, when player \(k\) plays according to \(\lambda\), the resulting play belongs to \(W\). Player \(k\) can force the play to be in \(W\).
Winning strategy

• Given a pair $(G, W)$

• We say that Player $k$ wins the game $(G, W)$ if and only if:

$$\exists \lambda : \text{Outcome}_k(G, \lambda) \subseteq W$$

We say $\lambda$ that is a winning strategy for player $k$ in the game $(G, W)$
Winning strategies

= 

Controllers that enforce winning plays
Winning conditions

- **Not all** winning conditions are reasonable
- One often assumes that the set of winning plays is a **regular set**
- We show here how to solve **reachability** and **safety** games
A Reachability Game

Does Player I, who owns the rounded positions, have a strategy (against any choices of Player II) to reach the set \{1101, 1111\}?
A Safety Game

Does Player I, who owns the rounded positions, have a strategy (against any choices of Player II) to stay within the set of states $Q \setminus \{1111\}$?
Symbolic algorithms to solve games
Player $k$ Controllable Predecessors

$X$ is a set of positions

$1\text{CPre}_G(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \land q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') : q' \in X\}$

Set of Player I positions where he has a choice of successor that lies in $X$

Set of Player II positions where all her choices for successors lie in $X$
Player $k$ Controllable Predecessors

$1CPr_{G}(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \land q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') \land q' \in X\}$

Symmetrically

$2CPr_{G}(X) = \{q \in Q_2 \mid \exists q' : \delta(q, q') \land q' \in X\} \cup \{q \in Q_1 \mid \forall q' : \delta(q, q') \land q' \in X\}$
Player $k$ Controllable Predecessors

1\text{CPre}_G(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \land q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') : q' \in X\}

2\text{CPre}_G(X) = \{q \in Q_2 \mid \exists q' : \delta(q, q') \land q' \in X\} \cup \{q \in Q_1 \mid \forall q' : \delta(q, q') : q' \in X\}

Monotonic functions over $\langle 2^{Q_1 \cup Q_2}, \subseteq \rangle$
$X = \{1000, 0101, 1111\}$
\[ X = \{1000, 0101, 1111\} \]

\[ 1\text{CPr}(X) = \{0000\} \cup \{0100, 1101\} \]

Rounded positions, there exists a red successor
\[ X = \{1000, 0101, 1111\} \]

\[ 1\text{CPre}(X) = \{0000\} \cup \{0100, 1101\} \]

- Rounded positions, there exists a red successor
- Squared positions, all successors are red
Fixpoints to Solve Games

Reachability game for set $Q$

$$\mu X \cdot Q \cup \mathbb{1} \text{CPre}(X)$$

Safety game for set $Q$

$$\nu X \cdot Q \cap \mathbb{1} \text{CPre}(X)$$
Fixpoint for a safety game

Does Player I, who owns the rounded positions, have a strategy to stay within the set of states $Q \setminus \{1111\}$?
Fixpoint for a safety game

We must compute

$$\nu X \cdot (Q \setminus \{1111\}) \cap 1\text{CPre}(X)$$

To do that, we use the Tarski fixpoint theorem.
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap \mathit{1CPre}(Q) \]
Fixpoint for a safety game

$X_0 = (Q \setminus \{1111\}) \cap 1CPr(Q)$
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPr}(Q) \]

\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPr}(X_0) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap \text{1CPre}(Q) \]

\[ X_1 = (Q \setminus \{1111\}) \cap \text{1CPre}(X_0) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap \text{1CPr}(Q) \]

\[ X_1 = (Q \setminus \{1111\}) \cap \text{1CPr}(X_0) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap \text{1CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap \text{1CPre}(X_0) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]

\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
Fixpoint for a safety game

\begin{align*}
X_0 &= (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \\
X_1 &= (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \\
X_2 &= (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1)
\end{align*}
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
\[ X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
\[ X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
\[ X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
\[ X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1) \]
Fixpoint for a safety game

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
\[ X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1) = X_1 \]

This is the greatest fixpoint
Fixpoint for a safety game

This is the greatest fixpoint

\[ X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q) \]
\[ X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0) \]
\[ X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1) = X_1 \]

\( X_2 \) is exactly the set of positions from which Player I can avoid entering \( \{1111\} \), no matter how Player II behaves.
Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$ be a TGS, let $\operatorname{Reach}(G, Q)$ be a reachability game defined on $G$, Player I has a winning strategy for this game iff

$$\iota \in \mu X \cdot Q \cup 1 \text{CPre}(X)$$
Let $G = \langle Q_1, Q_2, \nu, \delta \rangle$ be a TGS, let Safe$(G, \mathcal{Q})$ be a safety game defined on $G$, Player I has a winning strategy for this game iff

$$\nu \in \nu X \cdot Q \cap 1\text{CPre}(X)$$
Some more results

Any finite state game with regular objective can be solved.
Some more results

Any finite state game with regular objective can be solved.

Strategies for safety and reachability games are positional (no need for memory).
Some more results

Any finite state game with regular objective can be solved.

Strategies for safety and reachability games are positional (no need for memory).

For more complicated games, like LTL games, finite memory is needed.
Some more results

Any finite state game with **regular objective** can be solved.

Strategies for safety and reachability games are **positional** (no need for memory).

For more complicated games, like LTL games, finite **memory** is needed.

**Determinacy theorem**: In positional games (where a position is owned by a player), games are determinate in the following sense:

For any regular set of plays $W$,

Player I has a strategy to win $(G, W)$ iff Player II does not have a strategy to win $(G, \overline{W})$.
From the red states, and only from those states, Player II has a strategy to reach the state $1111$. 
Conclusion

• **Synthesis**: games, players, and strategies are **powerful metaphors** for controller synthesis;

• **To go beyond**, we need more/better **theoretical foundations**: games of imperfect information, randomized strategies, optimal strategies, infinite state games, etc.
General references on games and synthesis


References on timed and hybrid games


References on implementability issues and robustness


K. Altisen and S. Tripakis. Implementation of timed automata: an issue of semantics or modeling?. In FORMATS'05 (to appear). A previous version of this paper is available as VERIMAG Technical Report TR-2005-12.
