What is this talk about?

- To illustrate how the co-contravariant fixed point problem in synchronous languages can be handled by means of a game theory approach.
- To show that the Must/Cannot analysis of combinational Esterel corresponds to the computation of winning strategies of a particular game (the maze).
- Sequencing is modelled by a notion of stratified plays.
Synchronous Configurations: A Declarative View

Programming Synchronous Reactions

(reactive) instant

At every instant, signals may be present or absent, emitted or not emitted

Transitions

"if a present and b absent, then emit c and d"
Programming Synchronous Reactions

Transitions

Positive

\[ a, \neg b \rightarrow c, d \]

\( \text{trigger / action} \)

\[ P, \neg N \rightarrow A \]

"if a present and b absent, then emit c and d"

\[ \text{Declarative} \]

\[ a \land \neg b \supset c \land d \]

How do we code configurations?

Reducible to parallel composition of transitions

Transitions

Encoding

Nondeterministic (Statecharts)

\[ (a \land \neg t_2 \supset b) \land (c \land \neg t_1 \supset d) \]
How do we code configurations?

Reducible to parallel composition of transitions

Transitions

Encoding

Deterministic (Esterel, SyncCharts)

(a ⊒ b) ∧ (c ∧ ¬a ⊒ d)

How do we execute configurations?

Reducible to parallel composition of transitions

Transitions

Encoding

Nondeterministic:

(a ∧ ¬t₂ ⊒ b) ∧ (c ∧ ¬t₁ ⊒ d)

Sequential

(a ⊒ b) ∧ (c ∧ ¬a ⊒ d)

(a ∧ ¬c ⊒ b) ∧ (c ⊒ d)
How do we code configurations?

Reducible to parallel composition of transitions

Sequential

Eager

Lazy

Transitions

Encoding

Sequential (Eager)

(a \supset b \land t_1) \land (c \land t_1 \supset d)
How do we code configurations?

Reducible to parallel composition of transitions

Transitions

Encoding

Sequential (Lazy)

(a ⊃ b ∧ t₁) ∧ (c ∧ t₁ ⊃ d) ∧ (c ∧ ¬a / d)

How do we code configurations?

Reducible to parallel composition of transitions

environment input + configuration ⇒ reaction

Configuration

t₁∥t₂∥...∥tₙ ⇒ reaction (event)

reducible to parallel composition of transitions

Choice and priorities can be represented using negations
How do we code configurations?

configuration ⇒ reaction

Configuration
\[ t_1 \parallel t_2 \parallel \ldots \parallel t_N \parallel /a, e \] ⇒ reaction (event)

reducible to parallel composition of transitions
Choice and priorities can be represented using negations
Environment stimulus can be accounted for as part of the configuration

Configurations

**Kernel syntax**
(for "instantaneous" reactions in the combinational fragment)

\[
\begin{align*}
C & := \text{nothing} \\
& \quad | \text{emit } s \\
& \quad | \text{present } s \text{ then } C \\
& \quad | \text{present } s \text{ else } C \\
& \quad | C \parallel C \\
\end{align*}
\]

\[
\begin{align*}
C & := \text{true} \\
& \quad | s \\
& \quad | s / C \\
& \quad | s / C \\
& \quad | s / C \\
\end{align*}
\]

present s then C else D s / C || ¬s / D
Have a break ... play a Maze

Finite 2-Player Games

Finite game graph (maze)

- Players $A, B$ take alternate turns:
  - visible corridor ( ) = turn changes to opponent
  - secret corridor ( ) = player keeps his/her turn

- Winning Rule: “last player loses”
Finite 2-Player Games

A starts

Finite 2-Player Games

A

J. Aguado, Dec/02/03 SYNCHRON’03 Marseille
Finite 2-Player Games

A loses
Finite 2-Player Games

[Diagram of a finite 2-player game]

Finite 2-Player Games

[Diagram of another finite 2-player game]
Finite 2-Player Games

B loses
Strategies

Let \( M = (R, \downarrow, \tau) \) be a finite maze and \( \forall \rightarrow \).

A strategy is a partial mapping \( \alpha : R \rightarrow R \) such that
\[ \forall r. \text{if } \alpha(r) \text{ is defined then } r \rightarrow \alpha(r). \]

A pair of strategies (\( \alpha, \beta \)) and a start room \( r \) determines a unique play in \( M \), denoted \( \text{play}(\alpha, \beta, r) \), where
- player A uses function \( \alpha \)
- player B uses function \( \beta \)

to determine his/her next move, as long as \( \alpha \) and \( \beta \) are defined (maximality).

Winning Positions

Let \( \text{length}(\alpha, \beta, r) \) be the length (possibly \( \infty \)).
Let \( \text{last}(\alpha, \beta, r) \in \{A, B, \bot\} \) the last player in \( \text{play}(\alpha, \beta, r) \).

A room \( r \in R \) is called a

- winning position (for the starting player A)
  \[ \text{if } \exists \alpha. \forall \beta. \text{last}(\alpha, \beta, r) = B \]
- losing position (for the starting player A)
  \[ \text{if } \forall \alpha. \exists \beta. \text{last}(\alpha, \beta, r) = A \]
Winning Positions

Let \( \text{length}(\alpha, \beta, r) \) be the length (possibly \( \infty \)).
Let \( \text{last}(\alpha, \beta, r) \in \{A, B, \perp\} \) the last player in play(\( \alpha, \beta, r \)).

A room \( r \in R \) is called an
\[ \begin{align*}
\text{n-winning position (for the starting player A)} \\
& \text{if } \exists \alpha. \forall \beta. \text{length}(\alpha, \beta, r) \leq n \land \text{last}(\alpha, \beta, r) = B \\
\text{n-losing position (for the starting player A)} \\
& \text{if } \forall \alpha. \exists \beta. \text{length}(\alpha, \beta, r) \leq n \land \text{last}(\alpha, \beta, r) = A \\
\text{n-safe position (for the starting player A)} \\
& \text{if } \exists \alpha. \forall \beta. \text{length}(\alpha, \beta, r) > n \lor \text{last}(\alpha, \beta, r) = B
\end{align*} \]

Computing Winning Positions

Bounded Winning
\[\begin{align*}
R_{\text{win}}^n & := \{ r \in R \mid r \text{ is n-winning} \} \\
R_{\text{safe}}^n & := \{ r \in R \mid r \text{ is n-safe} \}
\end{align*}\]

Iterative Approximation:
\[\begin{align*}
R_{\text{win}}^{-1} & := \emptyset \\
R_{\text{safe}}^{-1} & := \text{all} \\
R_{\text{win}}^{n+1} & := \{ r \in R \mid \exists s \in R. (r \stackrel{\tau}{\rightarrow} s \land s \in R_{\text{win}}^n) \lor (r \stackrel{\perp}{\rightarrow} s \land s \notin R_{\text{safe}}^n) \} \\
R_{\text{safe}}^{n+1} & := \{ r \in R \mid \exists s \in R. (r \stackrel{\tau}{\rightarrow} s \land s \in R_{\text{safe}}^n) \lor (r \stackrel{\perp}{\rightarrow} s \land s \notin R_{\text{win}}^n) \}
\end{align*}\]
Computing Winning Positions

The iteration constructs approximation sequences
\[ \emptyset \subseteq R_0^{\text{win}} \subseteq R_1^{\text{win}} \subseteq R_2^{\text{win}} \subseteq \ldots \subseteq R_n^{\text{win}} \subseteq \ldots \]
all \[ \supseteq R_0^{\text{safe}} \supseteq R_1^{\text{safe}} \supseteq R_2^{\text{safe}} \supseteq \ldots \supseteq R_n^{\text{safe}} \supseteq \ldots \]
in which \[ \emptyset \subseteq R_{n+1}^{\text{win}} \subseteq R_{n+1}^{\text{safe}} \subseteq \text{all}. \]

Theorem (standard result)

Let \( R^{\text{win}} = \bigcup_n R_n^{\text{win}} \) and \( R^{\text{safe}} = \bigcap_n R_n^{\text{safe}} \). Then,
\[ \begin{align*}
\bullet r \in R^{\text{win}} & \iff r \text{ winning position} \\
\bullet r \in \text{all} \setminus R^{\text{safe}} & \iff r \text{ losing position} \\
\bullet r \in R^{\text{safe}} \setminus R^{\text{win}} & \iff r \text{ draw position.}
\end{align*} \]

So, what is the game in a synchronous reaction?
Configurations as Mazes

**Maze** $G_c$

- if c is winning position then a is winning position
- if b is losing position then d is winning position

**Configuration C**

\[
c/a \parallel a/x \parallel \neg x/a \parallel \neg d/a \parallel \neg d/x \parallel \neg y/d \parallel c/y \parallel \neg b/c \parallel \neg b/y \parallel \neg b/d \parallel \neg z/b
\]

- rooms = signals, corridors = transitions
- winning position = signal present
- losing position = signal absent

Mazes as Configurations

**Maze** $G_c$

**Configuration C**

\[
c/a \parallel a/x \parallel \neg x/a \parallel \neg d/a \parallel \neg d/x \parallel \neg y/d \parallel c/y \parallel \neg b/c \parallel \neg b/y \parallel \neg b/d \parallel \neg z/b
\]

- $R^{-1}_{\text{win}} = \emptyset$
- $R^{-1}_{\text{safe}} = \{a, b, c, d, x, y, z\}$
The Esterel Game

Maze $G_c$

Configuration C

$c/a \parallel a/x \parallel \neg x/a \parallel \neg d/a \parallel \neg d/x \parallel \neg y/d \parallel c/y \parallel \neg b/c \parallel \neg b/y \parallel \neg b/d \parallel \neg z/b$

$R^0_{\text{win}} = \emptyset$

$R^0_{\text{safe}} = \{a, b, c, d, x, y\}$  

$R^1_{\text{win}} = \{b\}$  

$R^1_{\text{safe}} = \{a, b, c, d, x, y\}$

Maze $G_c$

Configuration C

$c/a \parallel a/x \parallel \neg x/a \parallel \neg d/a \parallel \neg d/x \parallel \neg y/d \parallel c/y \parallel \neg b/c \parallel \neg b/y \parallel \neg b/d \parallel \neg z/b$

$R^1_{\text{win}} = \{b\}$  

$R^1_{\text{safe}} = \{a, b, c, d, x, y\}$
The Esterel Game

Maze $G_c$

Configuration C

Configuration C

$R^2_{\text{win}} = \{ b \}$

$R^2_{\text{safe}} = \{ a, b, d, x, y \}$

$b \text{ present}$

$z, c \text{ absent}$

The Esterel Game

Maze $G_c$

Configuration C

Configuration C

$R^3_{\text{win}} = \{ b \}$

$R^3_{\text{safe}} = \{ a, b, d, x \}$

$b \text{ present}$

$z, c, y \text{ absent}$
The Esterel Game

Maze $G_c$

Configuration C

$e/a \parallel a/x \parallel \neg x/a \parallel d/a \parallel d/x \parallel y/d \parallel e/y \parallel b/e \parallel b/y \parallel b/d \parallel z/b$

$R^4_{\text{win}} = \{ b, d \}$

$R^4_{\text{safe}} = \{a, b, d, x \}$

Fixed point reached

$b,d$ present

$z,c,y$ absent

The Esterel Game

Maze $G_c$

Configuration C

$e/a \parallel a/x \parallel \neg x/a \parallel d/a \parallel d/x \parallel y/d \parallel e/y \parallel b/e \parallel b/y \parallel b/d \parallel z/b$

$R^4_{\text{win}} = \{ b, d \} = R^4_{\text{win}}$

$b,d$ present

$z,c,y$ absent

Fixed point reached
Theorem

Let $C$ be a configuration and $G_C$ be the game associated with $C$. Then, for all signals $a \in \text{Sigs}$:

- $a$ is present in the Esterel reaction of $C$ iff $a$ is a winning position in $G_C$
- $a$ is absent in the Esterel reaction of $C$ iff $a$ is a losing position in $G_C$
- $a$ is non-constructive in the Esterel reaction of $C$ iff $a$ is a draw position