The Church Synthesis Problem over Continuous Time

Alex Rabinovich

joint with Daniel Fatal

February 7, 2024

Plan

1. The Church Synthesis Problem.
2. Buchi Landwerber Theorem.
3. From Discrete to Continuous Time.
4. The Church Synthesis over Continuous Time.
5. Some Proofs.
6. Conclusion.

Synthesis Problem

Input: A specification $S(I, O)$
Task: Find a program P which implements S, i.e.,

$$\forall I(S(I, P(I)).$$

Parameterized by Formal Specification and Implementation languages.

Synthesis Problem

Input: A specification $S(I, O)$
Task: Find a program P which implements S, i.e.,

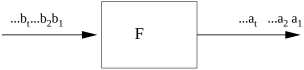
$$\forall I(S(I, P(I)).$$

Parameterized by Formal Specification and Implementation languages.

Church's Problem: Given an $MSO[<]$ formula that defines a relation between input ω-strings and output ω-strings, determine whether there exists an automaton (operator) that implements the specification.

Church's Problem

Consider a bit by bit transformation of bit streams

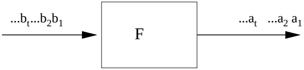


Church's Problem: For a given I-O specification on ω strings - fill the box.

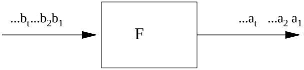
Given a logical specification of the input-output relation R find a mapping (implementation) $F: \ b \to F(b)$ such that $(b, F(b)) \in R$ for all b.

Implementation language - Causal Operators

Implementation language - Causal Operators

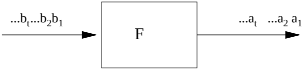


Implementation language - Causal Operators



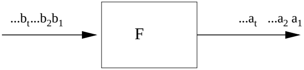
Causal-operator - a_t depends only on $b_1 b_2 \dots b_t$ - is independent from $b_{t+1} b_{t+2} \cdots$

Implementation language - Causal Operators



Causal-operator - a_t depends only on $b_1 b_2 \dots b_t$ - is independent from $b_{t+1} b_{t+2} \cdots$ - C-operators.

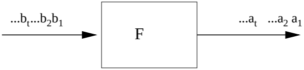
Implementation language - Causal Operators



Causal-operator - a_t depends only on $b_1 b_2 \dots b_t$ - is independent from $b_{t+1} b_{t+2} \cdots$ - C-operators.

Strongly Causal-operator - a_t depends only on $b_1 b_2 \dots b_i \dots$ $(i < t)$ - SC -operators.

Implementation language - Causal Operators

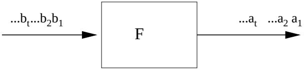


Causal-operator - a_t depends only on $b_1 b_2 \ldots b_t$ - is independent from $b_{t+1} b_{t+2} \cdots$ - C-operators.

Strongly Causal-operator - a_t depends only on $b_1 b_2 \ldots b_i \ldots$ $(i < t)$ - SC -operators.

SC-operators = Player I strategy; C-operators = Player II strategy .

Implementation language - Causal Operators



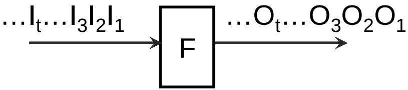
Causal-operator - a_t depends only on $b_1 b_2 \ldots b_t$ - is independent from $b_{t+1} b_{t+2} \cdots$ - C-operators.

Strongly Causal-operator - a_t depends only on $b_1 b_2 \ldots b_i \ldots$ $(i < t)$ - SC -operators.

SC-operators = Player I strategy; C-operators = Player II strategy .

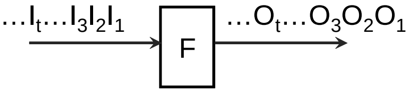
C-operators computable by finite automata, recursive C-operators.

Examples



Consider R_1 defined by
If all $I(t) = 0$ then all $O(t) = 0$; otherwise all $O(t) = 1$.

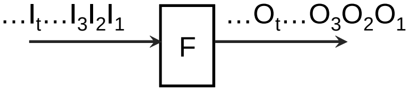
Examples



Consider R_1 defined by
If all $I(t) = 0$ then all $O(t) = 0$; otherwise all $O(t) = 1$.

$\forall I \exists O \;\; R_1(I, O)$.

Examples

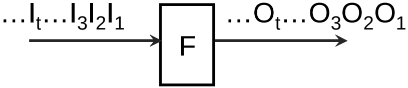


Consider R_1 defined by
If all $I(t) = 0$ then all $O(t) = 0$; otherwise all $O(t) = 1$.

$\forall I \exists O \;\; R_1(I, O)$.

Is it possible to implement R_1 by a causal operator?

Examples



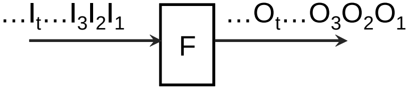
Consider R_1 defined by
If all $I(t) = 0$ then all $O(t) = 0$; otherwise all $O(t) = 1$.

$\forall I \exists O \quad R_1(I, O)$.

Is it possible to implement R_1 by a causal operator?

Consider R_2 defined by
If infinitely often $I(t) = 0$ then infinitely often $O(t) = 0$

Examples



Consider R_1 defined by
If all $I(t) = 0$ then all $O(t) = 0$; otherwise all $O(t) = 1$.

$\forall I \exists O \quad R_1(I, O)$.

Is it possible to implement R_1 by a causal operator?

Consider R_2 defined by
If infinitely often $I(t) = 0$ then infinitely often $O(t) = 0$

Is it possible to implement R_2 by a causal operator?

Example

Consider R defined by the conjunction of three conditions on the input-output stream (I, O):

1. $\forall t(I(t) = 1 \to O(t) = 1)$
2. never $O(t) = O(t+1) = 0$
3. If infinitely often $I(t) = 0$ then infinitely often $O(t) = 0$

Example

Consider R defined by the conjunction of three conditions on the input-output stream (I, O):

1. $\forall t(I(t) = 1 \to O(t) = 1)$
2. never $O(t) = O(t+1) = 0$
3. If infinitely often $I(t) = 0$ then infinitely often $O(t) = 0$

Common-Sense Solution

1. for input 1 produce output 1
2. for input 0 produce
   - output 1 if last output was 0
   - output 0 if last output was 1

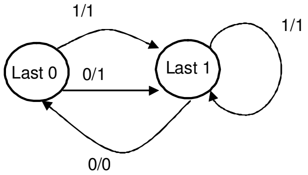
Example

Consider R defined by the conjunction of three conditions on the input-output stream (I, O):

1. $\forall t(I(t) = 1 \rightarrow O(t) = 1)$
2. never $O(t) = O(t+1) = 0$
3. If infinitely often $I(t) = 0$ then infinitely often $O(t) = 0$

Common-Sense Solution

1. for input 1 produce output 1
2. for input 0 produce
   - output 1 if last output was 0
   - output 0 if last output was 1



Can be described by a finite state automaton with output.

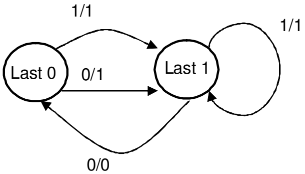
Example

Consider R defined by the conjunction of three conditions on the input-output stream (I, O):

1. $\forall t(I(t) = 1 \rightarrow O(t) = 1)$
2. never $O(t) = O(t+1) = 0$
3. If infinitely often $I(t) = 0$ then infinitely often $O(t) = 0$

Common-Sense Solution

1. for input 1 produce output 1
2. for input 0 produce
   - output 1 if last output was 0
   - output 0 if last output was 1



Can be described by a finite state automaton with output.
Equivalently, can be defined by an $MSO[<]$ formula $\Psi(X, Y)$.

Büchi-Landweber Theorem

In the examples the input-output specification $R(I, O)$ can be formalized in the Monadic second-order logic of order ($MSO[<]$).

Theorem (Büchi-Landweber(69))

Let $\Psi(X, Y)$ be an $MSO[<]$ formula.

1. *Determinacy: exactly one of the following holds for Ψ*
   a. *There is a C-operator F such that $\omega \models \forall X.\ \Psi(X, F(X))$.*
   b. *There is a SC-operator G such that $\omega \models \forall Y.\ \neg\Psi(G(Y), Y)$.*
2. *Decidability: it is decidable whether 1 (a) or 1 (b) holds.*
3. *Definability:*
   a. *If 1 (a) holds then there is an $MSO[<]$ formula U that defines a C-operator which implements Ψ.*
   b. *Similarly for 1 (b).*
4. *Computability: There is an algorithm such that for each $MSO[<]$ formula $\Psi(X, Y)$:*
   a. *If 1 (a) holds, constructs an $MSO[<]$ formula that defines F.*
   b. *Similarly for 1 (b).*

Church's view

Church (Cornell 1957) does not explicitly restrict to finite state systems. He has a vague and general formulation about "logistic systems" and "circuits" and discussed infinite state systems.

"Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The *synthesis problem* is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit)."

Church's view

Church (Cornell 1957) does not explicitly restrict to finite state systems. He has a vague and general formulation about "logistic systems" and "circuits" and discussed infinite state systems.

"Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The *synthesis problem* is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit)."

Following the Büchi-Landweber paper the community narrowed the view of Church's Problem to the finite-state case. Equivalently, to the $MSO[<]$-definable C-operators.

Continuous Time

Trakhtenbrot (1995) suggested to lift the Classical Automata Theory from Discrete to Continuous Time.

Continuous Time

The set of Non-Negative Reals $(\mathcal{R}^{\geq 0}, <)$ instead of $\omega = (Nat, <)$.

Continuous Time

The set of Non-Negative Reals $(\mathcal{R}^{\geq 0}, <)$ instead of $\omega = (Nat, <)$.

$MSO[<]$ has a natural interpretation over the non-negative reals $(\mathcal{R}^{\geq 0}, <)$.

Continuous Time

The set of Non-Negative Reals $(\mathcal{R}^{\geq 0}, <)$ instead of $\omega = (Nat, <)$.

$MSO[<]$ has a natural interpretation over the non-negative reals $(\mathcal{R}^{\geq 0}, <)$.
Unfortunately, the satisfiability problem for $MSO[<]$ over the reals is undecidable (Shelah 1975).

Continuous Time

The set of Non-Negative Reals $(\mathcal{R}^{\geq 0}, <)$ instead of $\omega = (Nat, <)$.

$MSO[<]$ has a natural interpretation over the non-negative reals $(\mathcal{R}^{\geq 0}, <)$.
Unfortunately, the satisfiability problem for $MSO[<]$ over the reals is undecidable (Shelah 1975).

Reasons - quantification over arbitrary predicates, e.g. P is True at the rationals and False at the irrationals.

Continuous Time

The set of Non-Negative Reals $(\mathcal{R}^{\geq 0}, <)$ instead of $\omega = (Nat, <)$.

$MSO[<]$ has a natural interpretation over the non-negative reals $(\mathcal{R}^{\geq 0}, <)$.
Unfortunately, the satisfiability problem for $MSO[<]$ over the reals is undecidable (Shelah 1975).

Reasons - quantification over arbitrary predicates, e.g. P is True at the rationals and False at the irrationals.

In Computer Science, it is natural to restrict to finitely variable (non-Zeno) predicates,

Continuous Time

The set of Non-Negative Reals $(\mathcal{R}^{\geq 0}, <)$ instead of $\omega = (Nat, <)$.

$MSO[<]$ has a natural interpretation over the non-negative reals $(\mathcal{R}^{\geq 0}, <)$.
Unfortunately, the satisfiability problem for $MSO[<]$ over the reals is undecidable (Shelah 1975).

Reasons - quantification over arbitrary predicates, e.g. P is True at the rationals and False at the irrationals.

In Computer Science, it is natural to restrict to finitely variable (non-Zeno) predicates,

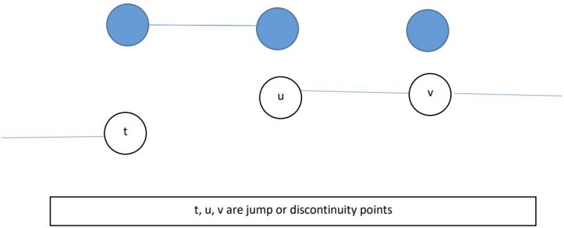
$P \subseteq \mathcal{R}^{\geq 0}$ is a finitely variable (non-Zeno) predicate, if there is an unbounded sequence $0 = a_0 < a_1 < \cdots < a_i < \cdots$ such that P is constant on every interval (a_i, a_{i+1}).

Finite Variability

A signal S a function from $\mathcal{R}^{\geq 0}$ to a finite alphabet Σ.
A signal S is finitely variable if there is an unbounded sequence $0 = a_0 < a_1 < \cdots < a_i < \cdots$ such that P is constant on every interval (a_i, a_{i+1}).

Finite Variability

A signal S a function from $\mathcal{R}^{\geq 0}$ to a finite alphabet Σ.
A signal S is finitely variable if there is an unbounded sequence $0 = a_0 < a_1 < \cdots < a_i < \cdots$ such that P is constant on every interval (a_i, a_{i+1}).



Decidability of $MSO[<]$

$FVsig$ is the structure for $MSO[<]$ over $(\mathcal{R}^{\geq 0}, <)$ with the finite variability predicates for the monadic variables.

Decidability of $MSO[<]$

$FVsig$ is the structure for $MSO[<]$ over $(\mathcal{R}^{\geq 0}, <)$ with the finite variability predicates for the monadic variables.

Theorem

$MSO[<]$ over $FVsig$ is decidable.

Casual and Strongly Causal Operators

$F : FVsig \rightarrow FVsig$ is causal if for every t and S, the value of $F(S)$ at t depends only on $S \upharpoonright [0,t]$. i.e., is independent from $S \upharpoonright (t,\infty)$.

Casual and Strongly Causal Operators

$F : FVsig \to FVsig$ is causal if for every t and S, the value of $F(S)$ at t depends only on $S \restriction [0, t]$. i.e., is independent from $S \restriction (t, \infty)$.

$F : FVsig \to FVsig$ is strongly causal if for every t and S, the value of $F(S)$ at t depends only on $S \restriction [0, t)$.

Church Synthesis Problems Continuous vs Discrete

Church Synthesis problem for Continuous time

Input: an $MSO[<]$ formula $\Psi(X, Y)$.

Question: Is there is a C-operator F such that $\forall X.\ \Psi(X, F(X))$ holds in $FVsig$?

Church Synthesis Problems Continuous vs Discrete

Church Synthesis problem for Continuous time
Input: an $MSO[<]$ formula $\Psi(X, Y)$.
Question: Is there is a C-operator F such that $\forall X.\ \Psi(X, F(X))$ holds in $FVsig$?

vs

Church Synthesis problem for Discrete time
Input: an $MSO[<]$ formula $\Psi(X, Y)$.
Question: Is there is a C-operator F such that $\forall X.\ \Psi(X, F(X))$ holds in ω?

Results for Continuous Time

Indeterminacy

Theorem (Indeterminacy)

The synthesis problem for continuous time is indeterminate. There exists an $MSO[<]$ formula $\Psi(X,Y)$ such that

1. *There is no C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
2. *There is no SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

vs Discrete case

Theorem (Determinacy)

Let $\Psi(X,Y)$ be an $MSO[<]$ formula. Exactly one of the following holds for Ψ

a. *There is a C-operator F such that $\omega \models \forall X.\ \Psi(X, F(X))$.*
b. *There is a SC-operator G such that $\omega \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

Dichotomy Fails

Theorem (Dichotomy Fails)

There exists an $MSO[<]$ *formula* $\Psi(X, Y)$ *such that both*

1. *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is a SC-operator* G *such that* $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.

Definability Fails

Theorem (Undefinability)

There exists $\Psi(X, Y)$ *such that*

1. *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is no* $MSO[<]$*-definable C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.

vs Discrete case

Theorem (Definability)

If there is a C-operator F *such that* $\omega \models \forall X.\ \Psi(X, F(X))$ *holds, then there is an* $MSO[<]$*-definable C-operator which implements* Ψ.

Two versions of the Church Synthesis Problem for Continuous Time

Input: an $MSO[<]$ formula $\Psi(X, Y)$.
Implementation Question: Is there is a C-operator F such that $\forall X.\ \Psi(X, F(X))$ holds in $FVsig$?
Definable Implementation Question: Is there is an MSO-definable C-operator F such that $\forall X.\ \Psi(X, F(X))$ holds in $FVsig$?

Decidability of Synthesis

Theorem (Computability of Definable Synthesis)

Given an $MSO[<]$ *formula* $\Psi(X, Y)$, *it is decidable whether exists an* *$MSO[<]$-definable C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$ *and if so, there is an algorithm that constructs an* $MSO[<]$ *formula that defines* F.

Decidability of Synthesis

Theorem (Computability of Definable Synthesis)

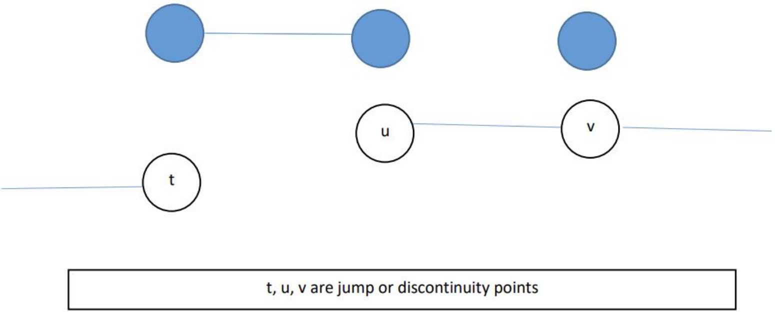
Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether exists an $MSO[<]$-definable C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$ and if so, there is an algorithm that constructs an $MSO[<]$ formula that defines F.

Theorem (Decidability of Synthesis)

Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether exists a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.

Proofs

FV Signals



The signal is constant on the intervals $[t, u]$ and (u, v).

Proof of Indeterminacy

Theorem (Indeterminacy)

There exists an $MSO[<]$ *formula* $\Psi(X,Y)$ *such that*

1. *There is no C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is no SC-operator* G *such that* $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.

Proof of Indeterminacy

Theorem (Indeterminacy)

There exists an $MSO[<]$ *formula* $\Psi(X,Y)$ *such that*

1. *There is no C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is no SC-operator* G *such that* $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.

$\Psi(X,Y) := \exists t > 0.Y$ is not continuous at t and X is constant at $(0,t]$.

Proof of Indeterminacy

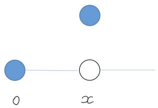
Theorem (Indeterminacy)

There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that

1. *There is no C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
2. *There is no SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

$\Psi(X, Y) := \exists t > 0.Y$ is not continuous at t and X is constant at $(0, t]$.

Define $\delta_x(t) := \begin{cases} 1, & \text{if } t = x \\ 0 & \text{else} \end{cases}$



Proof of Indeterminacy

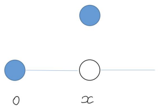
Theorem (Indeterminacy)

There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that

1. *There is no C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
2. *There is no SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

$\Psi(X, Y) := \exists t > 0. Y$ is not continuous at t and X is constant at $(0, t]$.

Define $\delta_x(t) := \begin{cases} 1, & \text{if } t = x \\ 0 & \text{else} \end{cases}$



If an C-operator F implements Ψ, then $Y_{\delta_2} := F(\delta_2)$ is not continuous at some $b \in (0, 2)$.

Proof of Indeterminacy

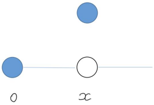
Theorem (Indeterminacy)

There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that

1. *There is no C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
2. *There is no SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

$\Psi(X, Y) := \exists t > 0. Y$ is not continuous at t and X is constant at $(0, t]$.

Define $\delta_x(t) := \begin{cases} 1, & \text{if } t = x \\ 0 & \text{else} \end{cases}$



If an C-operator F implements Ψ, then $Y_{\delta_2} := F(\delta_2)$ is not continuous at some $b \in (0, 2)$.

Let $b > 0$ be the minimal discontinuity of Y_{δ_2}.

Proof of Indeterminacy

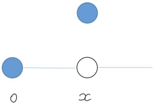
Theorem (Indeterminacy)

There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that

1. *There is no C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
2. *There is no SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

$\Psi(X, Y) := \exists t > 0. Y$ is not continuous at t and X is constant at $(0, t]$.

Define $\delta_x(t) := \begin{cases} 1, & \text{if } t = x \\ 0 & \text{else} \end{cases}$



If an C-operator F implements Ψ, then $Y_{\delta_2} := F(\delta_2)$ is not continuous at some $b \in (0, 2)$.
Let $b > 0$ be the minimal discontinuity of Y_{δ_2}.
$F(\delta_b)$ is not continuous at $c < b$. But δ_b and δ_2 are 0 on $[0, c]$, hence $F(\delta_b) = F(\delta_2)$ on $[0, c]$ - contradiction $1 = F(\delta_b)(c) \neq 0 = F(\delta_2)(c)$.

Proof of Indeterminacy -cont.

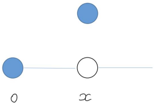


Figure: δ_x

Proof of Indeterminacy -cont.

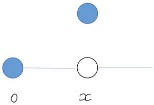


Figure: δ_x

$\Psi(X, Y) := \exists t > 0. Y$ jumps at t and X is constant at $(0, t]$

Proof of Indeterminacy -cont.

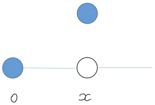


Figure: δ_x

$\Psi(X, Y) := \exists t > 0.Y$ jumps at t and X is constant at $(0, t]$
For no SC-operator G

$$\forall Y \neg \Psi(G(Y), Y).$$

Proof of Indeterminacy -cont.

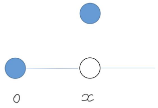


Figure: δ_x

$\Psi(X, Y) := \exists t > 0. Y$ jumps at t and X is constant at $(0, t]$
For no SC-operator G

$$\forall Y \neg \Psi(G(Y), Y).$$

$G(\delta_2)$ cannot be constant on $(0, b)$ for no $b > 2$.

Proof of Indeterminacy -cont.

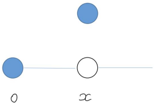


Figure: δ_x

$\Psi(X,Y) := \exists t > 0. Y$ jumps at t and X is constant at $(0,t]$
For no SC-operator G

$$\forall Y \neg \Psi(G(Y), Y).$$

$G(\delta_2)$ cannot be constant on $(0,b)$ for no $b > 2$. Hence, it jumps at some (minimal) $c \in (0,2]$.

Proof of Indeterminacy -cont.

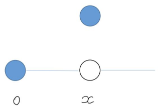


Figure: δ_x

$\Psi(X, Y) := \exists t > 0.Y$ jumps at t and X is constant at $(0, t]$
For no SC-operator G

$$\forall Y \neg \Psi(G(Y), Y).$$

$G(\delta_2)$ cannot be constant on $(0, b)$ for no $b > 2$. Hence, it jumps at some (minimal) $c \in (0, 2]$.

$G(\delta_{\frac{c}{2}})$ jumps at some $d \leq \frac{c}{2}$.

Proof of Indeterminacy -cont.

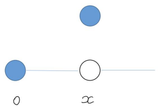


Figure: δ_x

$\Psi(X,Y) := \exists t > 0. Y$ jumps at t and X is constant at $(0,t]$
For no SC-operator G

$$\forall Y \neg \Psi(G(Y), Y).$$

$G(\delta_2)$ cannot be constant on $(0,b)$ for no $b > 2$. Hence, it jumps at some (minimal) $c \in (0,2]$.

$G(\delta_{\frac{c}{2}})$ jumps at some $d \leq \frac{c}{2}$.

δ_2 and $\delta_{\frac{c}{2}}$ coincide on $[0, \frac{c}{2})$,

Proof of Indeterminacy -cont.

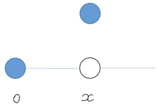


Figure: δ_x

$\Psi(X, Y) := \exists t > 0. Y$ jumps at t and X is constant at $(0, t]$
For no SC-operator G

$$\forall Y \neg \Psi(G(Y), Y).$$

$G(\delta_2)$ cannot be constant on $(0, b)$ for no $b > 2$. Hence, it jumps at some (minimal) $c \in (0, 2]$.

$G(\delta_{\frac{c}{2}})$ jumps at some $d \leq \frac{c}{2}$.

δ_2 and $\delta_{\frac{c}{2}}$ coincide on $[0, \frac{c}{2})$, however, $G(\delta_{\frac{c}{2}})$ differs from $G(\delta_2)$ at $d \leq \frac{c}{2}$. Contradicts that G is SC.

Dichotomy Fails

Theorem (Dichotomy Fails)

There exists an $MSO[<]$ *formula* $\Psi(X, Y)$ *such that both*

1. *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is a SC-operator* G *such that* $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.

Proof is elementary $\Psi \cdots$

Definability Fails

Theorem (Undefinability)

There exists $\Psi(X, Y)$ *such that*

1. *There is no MSO[<]-definable C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.

Definability Fails

Theorem (Undefinability)

There exists $\Psi(X, Y)$ *such that*

1. *There is no MSO[<]-definable C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.

A bijection $\rho : \mathcal{R}^{\geq 0} \to \mathcal{R}^{\geq 0}$ is an automorphism if it is order preserving.

Definability Fails

Theorem (Undefinability)

There exists $\Psi(X, Y)$ *such that*

1. *There is no MSO[<]-definable C-operator F such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
2. *There is a C-operator F such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.

A bijection $\rho : \mathcal{R}^{\geq 0} \to \mathcal{R}^{\geq 0}$ is an automorphism if it is order preserving.

Claim. If ρ is an automorphism, and $\Psi(P, Q)$ then $\Psi(\rho P, \rho Q)$.

Proof Undefinability

Consider a specification $\Psi(X, Y)$ that states:

- There is $t > 0$ such that Y is zero on $[0, t)$ and Y is one on $[t, \infty)$.

Proof Undefinability

Consider a specification $\Psi(X, Y)$ that states:

- There is $t > 0$ such that Y is zero on $[0, t)$ and Y is one on $[t, \infty)$.

This specification is easily formalized in $MSO[<]$.

Proof Undefinability

Consider a specification $\Psi(X, Y)$ that states:

- There is $t > 0$ such that Y is zero on $[0, t)$ and Y is one on $[t, \infty)$.

This specification is easily formalized in $MSO[<]$.

Claim. There is a C-operator that implements Ψ.

Proof Undefinability

Consider a specification $\Psi(X, Y)$ that states:

- There is $t > 0$ such that Y is zero on $[0, t)$ and Y is one on $[t, \infty)$.

This specification is easily formalized in $MSO[<]$.

Claim. There is a C-operator that implements Ψ.

Lemma. If $\Phi(X, Y)$ defines an operator and $\Phi(P, Q)$ then Q jumps at $t > 0$ only if P jumps at t.

Proof Undefinability

Consider a specification $\Psi(X, Y)$ that states:

- There is $t > 0$ such that Y is zero on $[0, t)$ and Y is one on $[t, \infty)$.

This specification is easily formalized in $MSO[<]$.

Claim. There is a C-operator that implements Ψ.

Lemma. If $\Phi(X, Y)$ defines an operator and $\Phi(P, Q)$ then Q jumps at $t > 0$ only if P jumps at t.

Hence, No $MSO[<]$-definable operator implements Ψ.

Indeed, take the input constant everywhere.

Proof of Lemma

Lemma. If $\Phi(X, Y)$ defines an operator and $\Phi(P, Q)$ then Q jumps at $t > 0$ only if P jumps at t.

Assume F is definable by $\Phi(X, Y)$. Assume that $\Phi(P, Q)$ and Q jumps at $t > 0$.
If P does not jump at t there are $t_1 < t < t_2$ such that P is constant in $[t_1, t_2]$, and t is the only jump of Q in $[t_1, t_2]$.

Proof of Lemma

Lemma. If $\Phi(X, Y)$ defines an operator and $\Phi(P, Q)$ then Q jumps at $t > 0$ only if P jumps at t.

Assume F is definable by $\Phi(X, Y)$. Assume that $\Phi(P, Q)$ and Q jumps at $t > 0$.
If P does not jump at t there are $t_1 < t < t_2$ such that P is constant in $[t_1, t_2]$, and t is the only jump of Q in $[t_1, t_2]$.

Take an automorphism ρ that is identity outside $[t_1.t_2]$ and moves t to $t' \neq t$, for $t' \in (t_1, t_2)$.

Proof of Lemma

Lemma. If $\Phi(X, Y)$ defines an operator and $\Phi(P, Q)$ then Q jumps at $t > 0$ only if P jumps at t.

Assume F is definable by $\Phi(X, Y)$. Assume that $\Phi(P, Q)$ and Q jumps at $t > 0$.
If P does not jump at t there are $t_1 < t < t_2$ such that P is constant in $[t_1, t_2]$, and t is the only jump of Q in $[t_1, t_2]$.

Take an automorphism ρ that is identity outside $[t_1.t_2]$ and moves t to $t' \neq t$, for $t' \in (t_1, t_2)$.

$\rho(P) = P$, $Q_1 := \rho(Q) \neq Q$, and $\Phi(P, Q)$ holds.

Proof of Lemma

Lemma. If $\Phi(X, Y)$ defines an operator and $\Phi(P, Q)$ then Q jumps at $t > 0$ only if P jumps at t.

Assume F is definable by $\Phi(X, Y)$. Assume that $\Phi(P, Q)$ and Q jumps at $t > 0$.
If P does not jump at t there are $t_1 < t < t_2$ such that P is constant in $[t_1, t_2]$, and t is the only jump of Q in $[t_1, t_2]$.

Take an automorphism ρ that is identity outside $[t_1.t_2]$ and moves t to $t' \neq t$, for $t' \in (t_1, t_2)$.

$\rho(P) = P$, $Q_1 := \rho(Q) \neq Q$, and $\Phi(P, Q)$ holds.
Hence $\Phi(P, Q_1)$ holds - contradicts that Φ defines an operator.

FV Signals and Timed ω-sequences

Representation of FV signals by ω sequence

Let $\widehat{t} := 0 = t_0 < t_1 < \cdots t_i < \cdots$ be an unbounded ω-sequence of reals and $\widehat{s} = (a_0, b_0)(a_1, b_1)\cdots$ be an ω string over $\Sigma \times \Sigma$.
The signal X represented by $(\widehat{t}, \widehat{s})$ is defined as follows:
$X(t_i) := a_i$ and $X(t) := b_i$ for $t \in (t_i, t_{i+1})$.

Representation of FV signals by ω sequence

Let $\widehat{t} := 0 = t_0 < t_1 < \cdots t_i < \cdots$ be an unbounded ω-sequence of reals and $\widehat{s} = (a_0, b_0)(a_1, b_1)\cdots$ be an ω string over $\Sigma \times \Sigma$.
The signal X represented by $(\widehat{t}, \widehat{s})$ is defined as follows:
$X(t_i) := a_i$ and $X(t) := b_i$ for $t \in (t_i, t_{i+1})$.

It is clear a signal is represented iff it is finitely variable signal.

Representation of FV signals by ω sequence

Let $\widehat{t} := 0 = t_0 < t_1 < \cdots t_i < \cdots$ be an unbounded ω-sequence of reals and $\widehat{s} = (a_0, b_0)(a_1, b_1)\cdots$ be an ω string over $\Sigma \times \Sigma$.
The signal X represented by $(\widehat{t}, \widehat{s})$ is defined as follows:
$X(t_i) := a_i$ and $X(t) := b_i$ for $t \in (t_i, t_{i+1})$.

It is clear a signal is represented iff it is finitely variable signal.

$(\widehat{t}, \widehat{s})$ is called a timed ω sequence; $\widehat{t}$ is its scale and $\widehat{s}$ its ω string.

Representation of FV signals by ω sequence

Let $\widehat{t} := 0 = t_0 < t_1 < \cdots t_i < \cdots$ be an unbounded ω-sequence of reals and $\widehat{s} = (a_0, b_0)(a_1, b_1)\cdots$ be an ω string over $\Sigma \times \Sigma$.
The signal X represented by $(\widehat{t}, \widehat{s})$ is defined as follows:
$X(t_i) := a_i$ and $X(t) := b_i$ for $t \in (t_i, t_{i+1})$.

It is clear a signal is represented iff it is finitely variable signal.

$(\widehat{t}, \widehat{s})$ is called a timed ω sequence; $\widehat{t}$ is its scale and $\widehat{s}$ its ω string.

Let L be an ω language over $\Sigma \times \Sigma$.
To L corresponds a set S of FV signals over Σ defined as $X \in S$ there is $\widehat{t}$ and there is $\widehat{s} \in L$ such that $(\widehat{s}, \widehat{t})$ represents X.

Representation of FV signals by ω sequence

Let $\widehat{t} := 0 = t_0 < t_1 < \cdots t_i < \cdots$ be an unbounded ω-sequence of reals and $\widehat{s} = (a_0, b_0)(a_1, b_1)\cdots$ be an ω string over $\Sigma \times \Sigma$.
The signal X represented by $(\widehat{t}, \widehat{s})$ is defined as follows:
$X(t_i) := a_i$ and $X(t) := b_i$ for $t \in (t_i, t_{i+1})$.

It is clear a signal is represented iff it is finitely variable signal.

$(\widehat{t}, \widehat{s})$ is called a timed ω sequence; $\widehat{t}$ is its scale and $\widehat{s}$ its ω string.

Let L be an ω language over $\Sigma \times \Sigma$.
To L corresponds a set S of FV signals over Σ defined as $X \in S$ there is $\widehat{t}$ and there is $\widehat{s} \in L$ such that $(\widehat{s}, \widehat{t})$ represents X.

We say that S is represented by L.

$MSO[<]$-definable FV signal Languages

Theorem

A FV signal language is $MSO[<]$ definable (over $(\mathcal{R}^{\geq 0}, <)$) iff it is represented by an $MSO[<]$ -definable ω languages.

Corollary

A FV signal language is $MSO[<]$ definable iff it is represented by an ω language accepted by a deterministic parity automaton.

Computability of Definable Synthesis

Theorem (Computability of Definable Synthesis)

Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether exists an $MSO[<]$-definable C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$ and if so, there is an algorithm that constructs an $MSO[<]$ formula that defines F.

Computability of Definable Synthesis

Theorem (Computability of Definable Synthesis)

Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether exists an $MSO[<]$-definable C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$ and if so, there is an algorithm that constructs an $MSO[<]$ formula that defines F.

A proof is similar to the Buchi - Landwerber proof for Discrete time. It relies on ω-Games over finite boards.

Computability of Definable Synthesis

Theorem (Computability of Definable Synthesis)

Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether exists an $MSO[<]$-definable C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$ and if so, there is an algorithm that constructs an $MSO[<]$ formula that defines F.

A proof is similar to the Buchi - Landwerber proof for Discrete time. It relies on ω-Games over finite boards.
There are some subtleties.
E.g. If Input player makes a move that does not make a jump in the corresponding signal, then the Output player is not allowed to make a move that creates a jump.

Decidability of Synthesis

Theorem (Decidability of Synthesis)

Given an $MSO[<]$ formula $\Psi(X,Y)$, it is decidable whether exists a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.

Decidability of Synthesis

Theorem (Decidability of Synthesis)

Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether exists a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.

Proof is based to a reduction to ω-games. However, even the alphabet of this games is uncountable.

Game

Round 0:

$\mathcal{I}$ set $t_0 := 0$. Then $\mathcal{I}$ chooses a_0. $\mathcal{O}$ chooses b_0. This defines X and Y on the interval $[0, t_0] = [0, 0]$.

Round n+1:

(A) $\mathcal{I}$ chooses a^d_{n+1} such that X will have this value for a while after t_n.
(B) $\mathcal{O}$ replies by suggesting an output signal Sig_{n+1} on the interval (t_n, ∞) under the condition that the input has the value a^d_{n+1} on all points of this interval.
(C) $\mathcal{I}$ either agrees and then games ends with the signals defined on $[0, \infty)$, or set $t_{n+1} > t_n$, agrees that on the points of (t_n, t_{n+1}) the input has value a^d_{n+1} and the output is the same as Sig_{n+1} on (t_n, t_{n+1}), and $\mathcal{I}$ will define a jump point at t_{n+1}.
(D) $\mathcal{I}$ chooses a value a_{n+1} for the input signal at t_{n+1}, and $\mathcal{O}$ replies by choosing b_{n+1} for the output at t_{n+1}. Now, input and output are defined on $[0, t_{n+1}]$ and a new round starts.

Winning Conditions

The winning condition for $\mathcal{O}$:

If $\lim_n t_n < \infty$ then $\mathcal{O}$ wins.

If $\lim_n t_n = \infty$ then the play has defined the input X and output Y on all points of $[0, \infty)$, and $\mathcal{O}$ wins iff $\Psi(X, Y)$ holds.

Winning Conditions

The winning condition for $\mathcal{O}$:

If $\lim_n t_n < \infty$ then $\mathcal{O}$ wins.
If $\lim_n t_n = \infty$ then the play has defined the input X and output Y on all points of $[0, \infty)$, and $\mathcal{O}$ wins iff $\Psi(X, Y)$ holds.

Theorem

Let Ψ be an $MSO[<]$ formula. Player $\mathcal{O}$ wins iff there is a C operators that implements Ψ.

Winning Conditions

The winning condition for $\mathcal{O}$:

If $\lim_n t_n < \infty$ then $\mathcal{O}$ wins.
If $\lim_n t_n = \infty$ then the play has defined the input X and output Y on all points of $[0, \infty)$, and $\mathcal{O}$ wins iff $\Psi(X, Y)$ holds.

Theorem

Let Ψ be an $MSO[<]$ formula. Player $\mathcal{O}$ wins iff there is a C operators that implements Ψ.

In the game each player has uncountable many possible move at each round $i > 0$. Our main technical results reduce this game to a game with finitely many moves at each round, and further reduce it to a parity game on a finite arena.

Simple Strategies

Let $\rho := (\widehat{t}, \widehat{s})$ be a timed sequence.
A timed sequence is simple if its untimed version $\widehat{s}$ is ultimately periodic and its time scale is uniform ($t_i := \Delta \times i$ for some Δ and all i).
A simple move - move that uses a simple sequence.

Simple Strategies

Let $\rho := (\widehat{t}, \widehat{s})$ be a timed sequence.
A timed sequence is simple if its untimed version $\widehat{s}$ is ultimately periodic and its time scale is uniform ($t_i := \Delta \times i$ for some Δ and all i).
A simple move - move that uses a simple sequence.

Lemma

$\mathcal{O}$ has a winning strategy iff there is a finite set U of ultimately periodic strings such that $\mathcal{O}$ has a winning strategy which uses only simple moves with strings from U.

Simple Strategies

Let $\rho := (\widehat{t}, \widehat{s})$ be a timed sequence.
A timed sequence is simple if its untimed version $\widehat{s}$ is ultimately periodic and its time scale is uniform ($t_i := \Delta \times i$ for some Δ and all i).
A simple move - move that uses a simple sequence.

Lemma

$\mathcal{O}$ has a winning strategy iff there is a finite set U of ultimately periodic strings such that $\mathcal{O}$ has a winning strategy which uses only simple moves with strings from U.
Moreover, at i-th move $\mathcal{O}$ uses the timescale $\frac{1}{2^i}$.

Simple Strategies

Let $\rho := (\widehat{t}, \widehat{s})$ be a timed sequence.
A timed sequence is simple if its untimed version $\widehat{s}$ is ultimately periodic and its time scale is uniform ($t_i := \Delta \times i$ for some Δ and all i).
A simple move - move that uses a simple sequence.

Lemma

$\mathcal{O}$ has a winning strategy iff there is a finite set U of ultimately periodic strings such that $\mathcal{O}$ has a winning strategy which uses only simple moves with strings from U.
Moreover, at i-th move $\mathcal{O}$ uses the timescale $\frac{1}{2^i}$.

This restricted games can be converted to memoryless restricted games over finite arenas.

Simple Strategies

Let $\rho := (\widehat{t}, \widehat{s})$ be a timed sequence.
A timed sequence is simple if its untimed version $\widehat{s}$ is ultimately periodic and its time scale is uniform ($t_i := \Delta \times i$ for some Δ and all i).
A simple move - move that uses a simple sequence.

Lemma

$\mathcal{O}$ has a winning strategy iff there is a finite set U of ultimately periodic strings such that $\mathcal{O}$ has a winning strategy which uses only simple moves with strings from U.
Moreover, at i-th move $\mathcal{O}$ uses the timescale $\frac{1}{2^i}$.

This restricted games can be converted to memoryless restricted games over finite arenas.

Hence, decidability.

Open Problems

Metrical Extensions - add +1 function.

Open Problems

Metrical Extensions - add +1 function.

Unfortunately, even $FO[<, +1]$ is undecidable.

Open Problems

Metrical Extensions - add +1 function.

Unfortunately, even $FO[<, +1]$ is undecidable.

Restricted metric properties.
"P will happen at distance at most 1 from t."
MITL logic.

Open Problems

Metrical Extensions - add +1 function.

Unfortunately, even $FO[<, +1]$ is undecidable.

Restricted metric properties.
"P will happen at distance at most 1 from t."
MITL logic.

Is the Church synthesis decidable for MITL specifications?

Open Problems

Metrical Extensions - add +1 function.

Unfortunately, even $FO[<, +1]$ is undecidable.

Restricted metric properties.
"P will happen at distance at most 1 from t."
MITL logic.

Is the Church synthesis decidable for MITL specifications?

THANK YOU