On Interactively Computable Functions

Jérémie Cabessa
Joint work with Alessandro E.P. Villa

Laboratory of Mathematical Economics
University of Paris 2
France

24 June 2014

Outline

Introduction

Interactive Computation

Interactive Turing Machines

Interactively computable Functions

Conclusion

Outline

Introduction

Interactive Computation

Interactive Turing Machines

Interactively computable Functions

Conclusion

Outline

Introduction

Interactive Computation

Interactive Turing Machines

Interactively computable Functions

Conclusion

Outline

Introduction

Interactive Computation

Interactive Turing Machines

Interactively computable Functions

Conclusion

Outline

Introduction

Interactive Computation

Interactive Turing Machines

Interactively computable Functions

Conclusion

Introduction

- Classical computation was argued to no longer fully correspond to the current notions of computing in modern systems (closed box, amnesic,...).
- Interactive computation captures sequential interactivity between the system and its environment, as well as persistence of memory throughout the whole computational process.
- We provide a machine-based and a mathematical characterisation of the functions that are computable by any kind of deterministic interactive system.

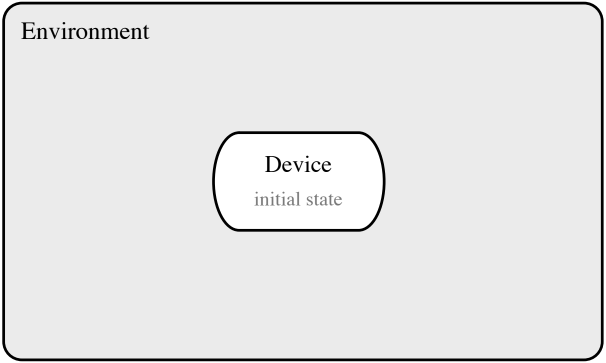
Introduction

- Classical computation was argued to no longer fully correspond to the current notions of computing in modern systems (closed box, amnesic,...).
- Interactive computation captures sequential interactivity between the system and its environment, as well as persistence of memory throughout the whole computational process.
- We provide a machine-based and a mathematical characterisation of the functions that are computable by any kind of deterministic interactive system.

Introduction

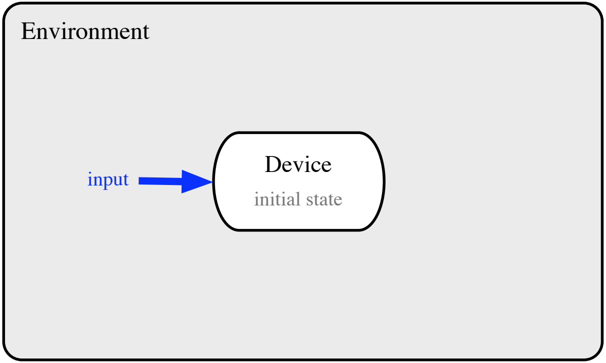
- Classical computation was argued to no longer fully correspond to the current notions of computing in modern systems (closed box, amnesic,...).
- Interactive computation captures sequential interactivity between the system and its environment, as well as persistence of memory throughout the whole computational process.
- We provide a machine-based and a mathematical characterisation of the functions that are computable by any kind of deterministic interactive system.

Classical Computation



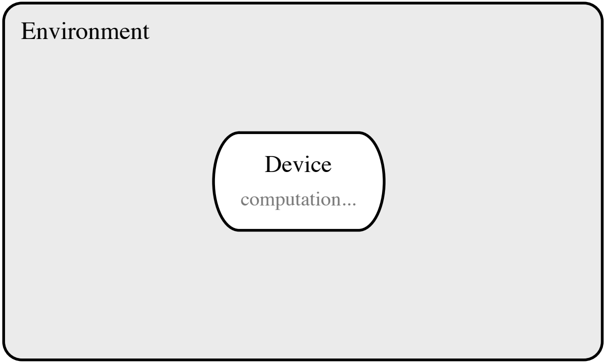
Closed-box and amnesic...

Classical Computation



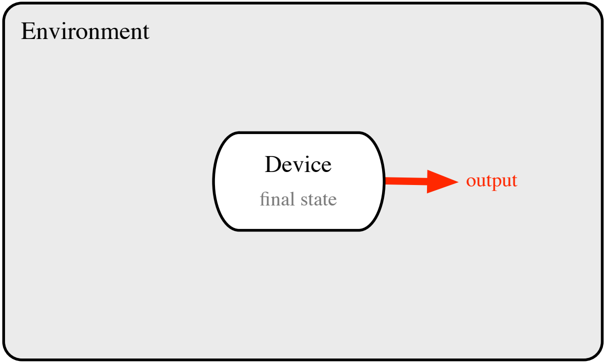
Closed-box and amnesic...

Classical Computation



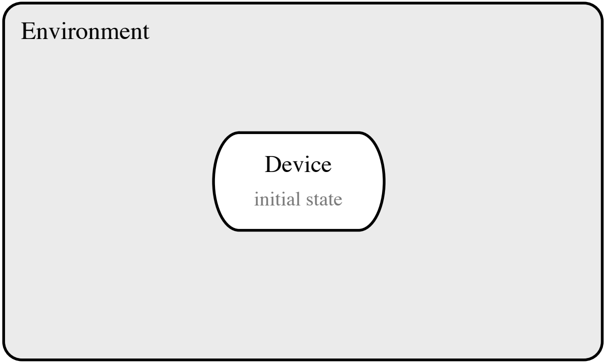
Closed-box and amnesic...

Classical Computation



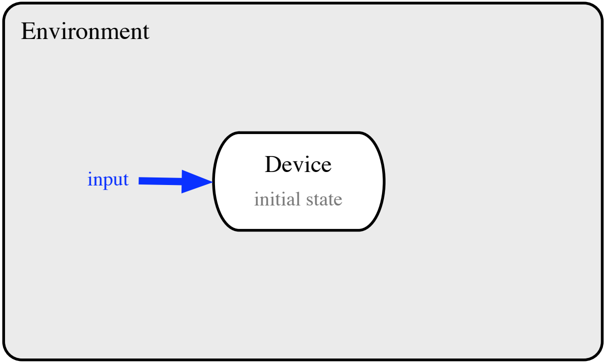
Closed-box and amnesic...

Classical Computation



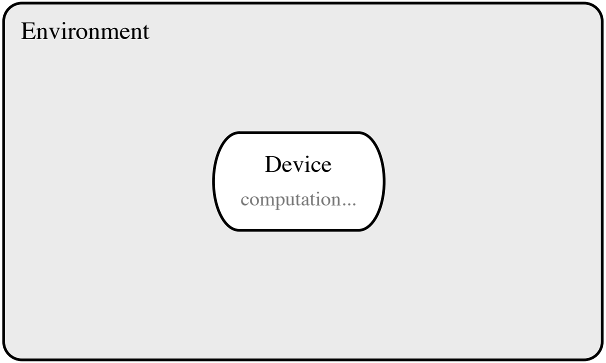
Closed-box and amnesic...

Classical Computation



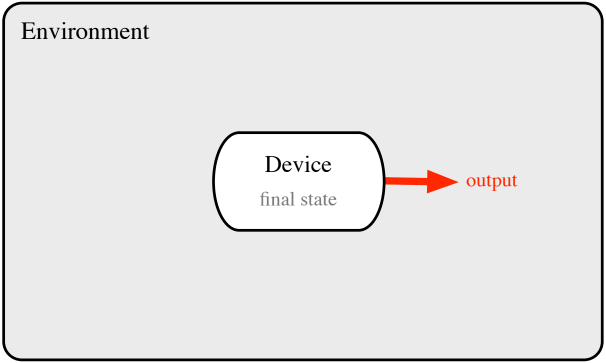
Closed-box and amnesic...

Classical Computation



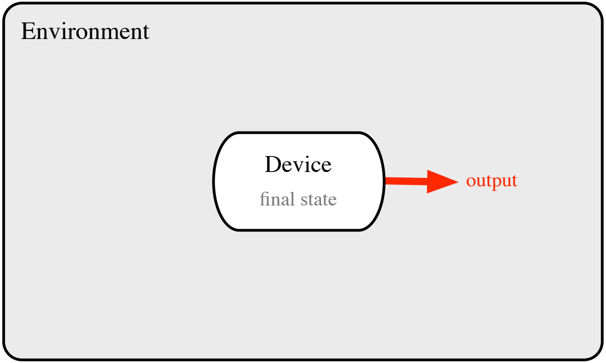
Closed-box and amnesic...

Classical Computation



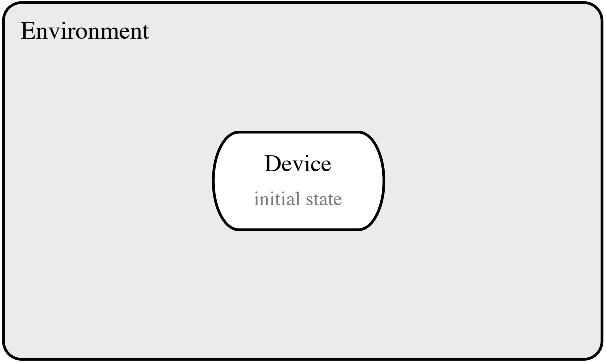
Closed-box and amnesic...

Classical Computation



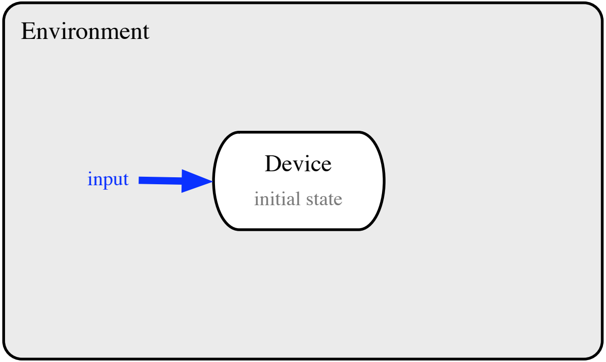
Closed-box and amnesic...

Interactive Computation



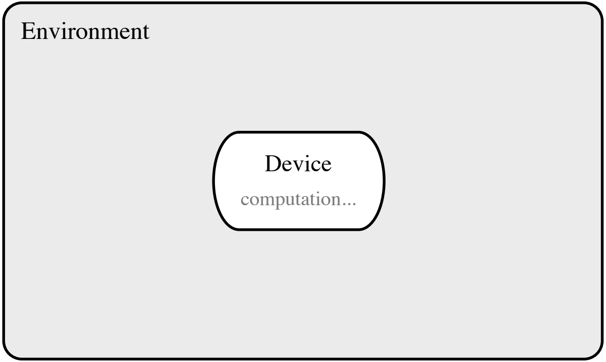
Sequentially interactive and memory active...

Interactive Computation



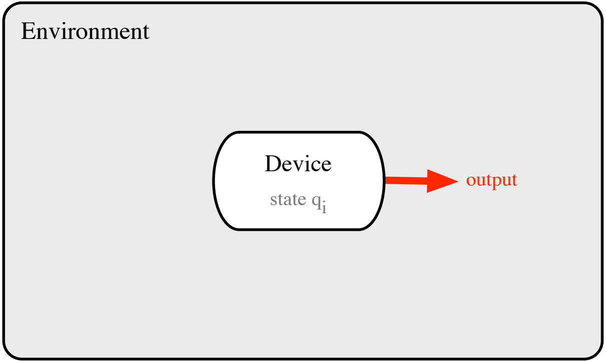
Sequentially interactive and memory active...

Interactive Computation



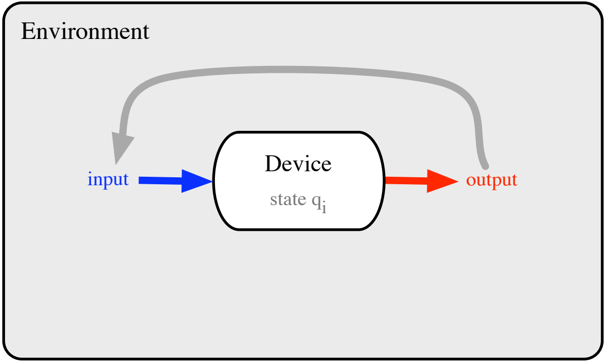
Sequentially interactive and memory active...

Interactive Computation



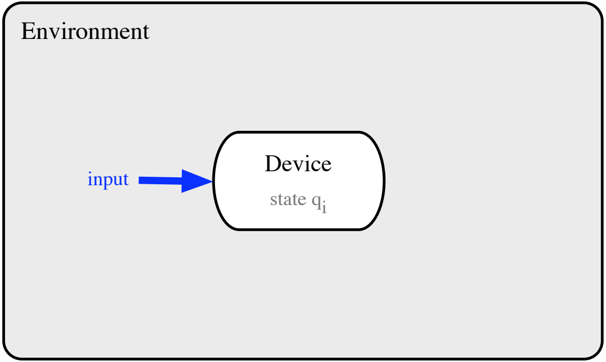
Sequentially interactive and memory active...

Interactive Computation



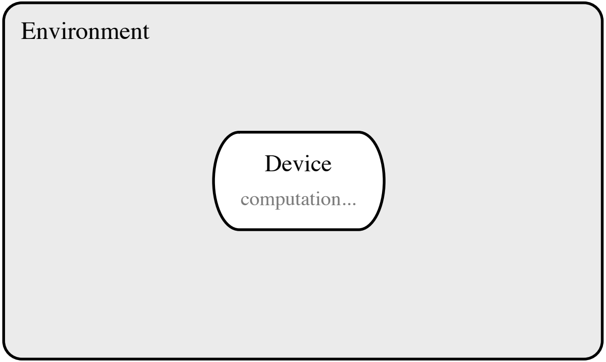
Sequentially interactive and memory active...

Interactive Computation



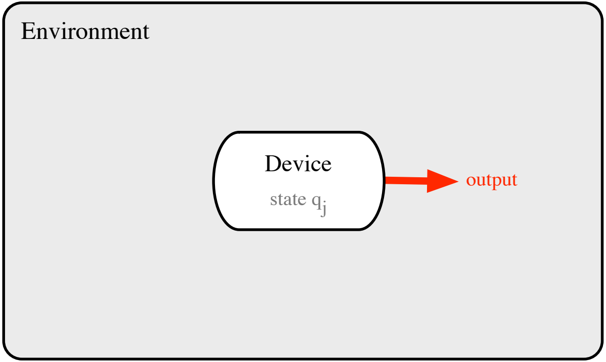
Sequentially interactive and memory active...

Interactive Computation



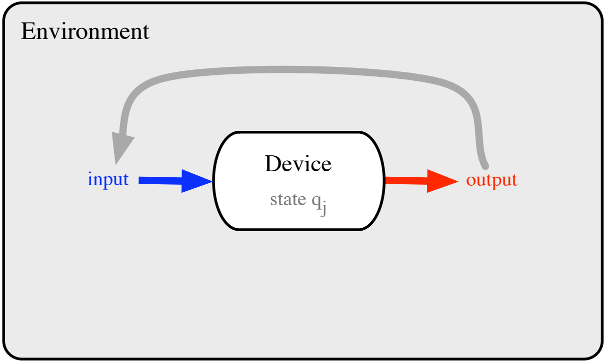
Sequentially interactive and memory active...

Interactive Computation



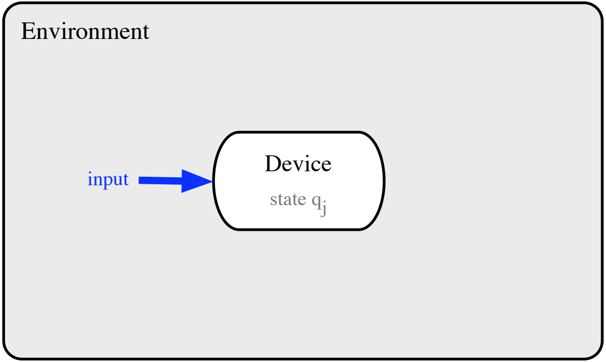
Sequentially interactive and memory active...

Interactive Computation



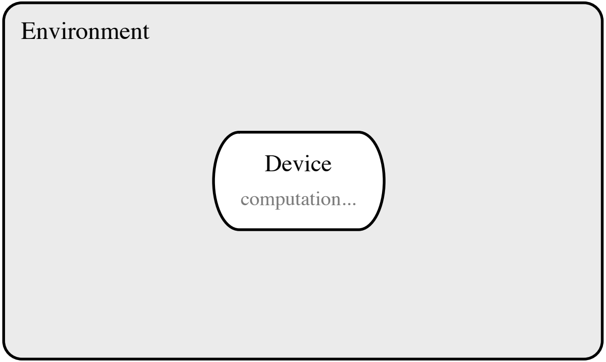
Sequentially interactive and memory active...

Interactive Computation



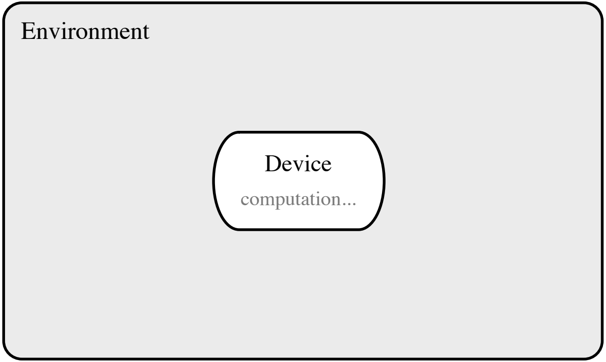
Sequentially interactive and memory active...

Interactive Computation



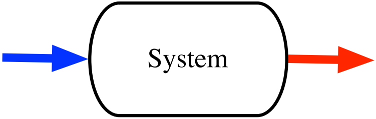
Sequentially interactive and memory active...

Interactive Computation



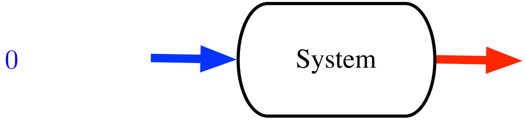
Sequentially interactive and memory active...

Interactive Scenario



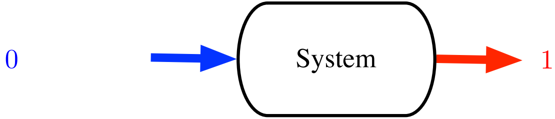
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



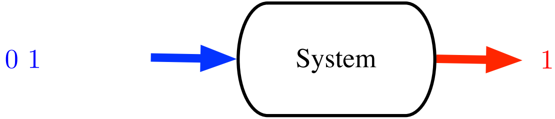
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



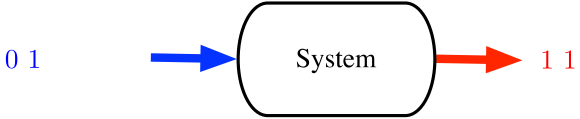
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



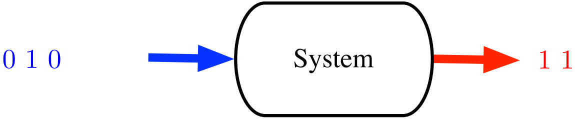
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



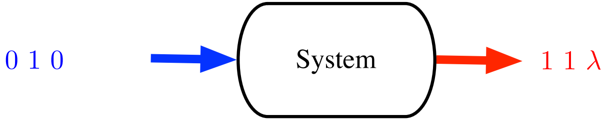
In this way, any deterministic interactive system S realises an *ω-translation* $\varphi_S : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



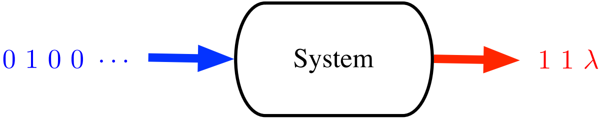
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



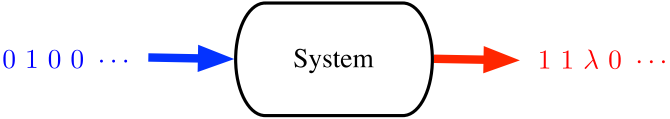
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



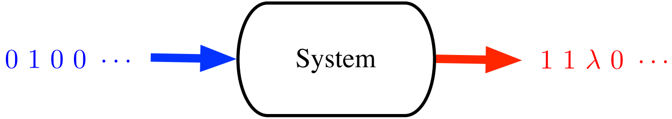
In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Scenario



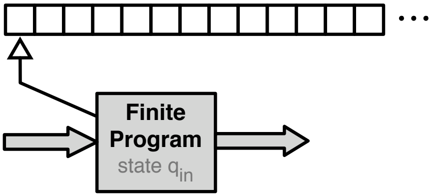
In this way, any deterministic interactive system S realises an *ω-translation* $\varphi_S : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$.

Interactive Scenario

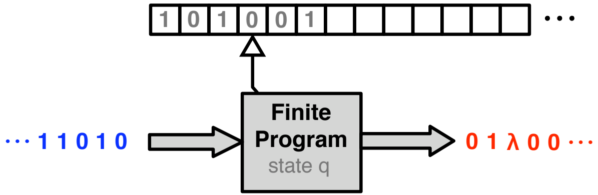


In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Interactive Turing machine (ITM)

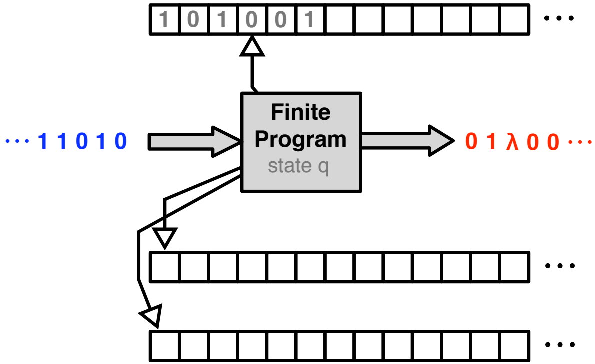


Interactive Turing machine (ITM)



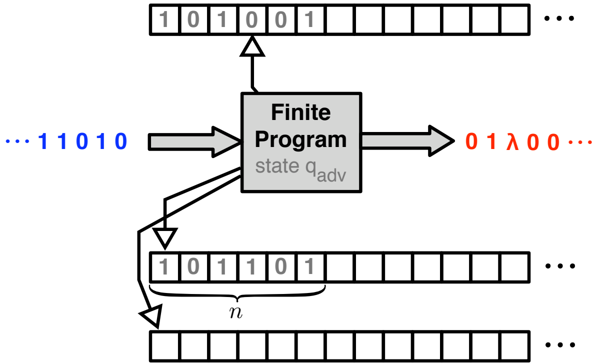
Interactive Turing machine with advice (ITM/A)

An ITM provided with additional advice input and output tapes, as well as an advice function $\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$



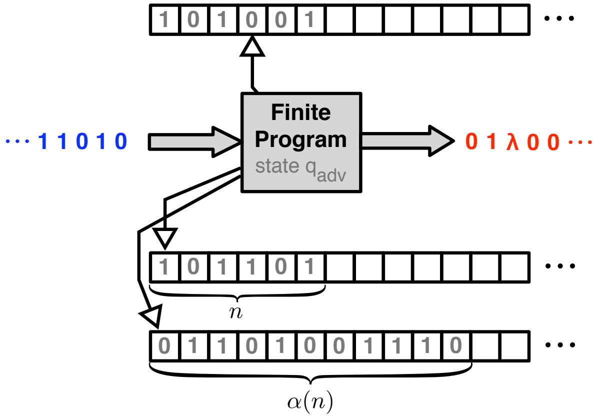
Interactive Turing machine with advice (ITM/A)

An ITM provided with additional advice input and output tapes, as well as an advice function $\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$



Interactive Turing machine with advice (ITM/A)

An ITM provided with additional advice input and output tapes, as well as an advice function $\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$



Continuous and Recursive Continuous ω-Translations

Any monotone function $f : \{0,1\}^* \to \{0,1\}^*$ induces *in the limit* an ω-translation $f_\omega : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by

$$f_\omega(x) = \lim_{i \geq 0} f(x[0{:}i])$$

Continuous and Recursive Continuous ω-Translations

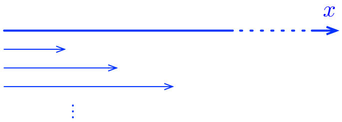
Any monotone function $f : \{0,1\}^* \to \{0,1\}^*$ induces *in the limit* an ω-translation $f_\omega : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by

$$f_\omega(x) = \lim_{i \geq 0} f(x[0{:}i])$$

Continuous and Recursive Continuous ω-Translations

Any monotone function $f : \{0,1\}^* \to \{0,1\}^*$ induces *in the limit* an ω-translation $f_\omega : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by

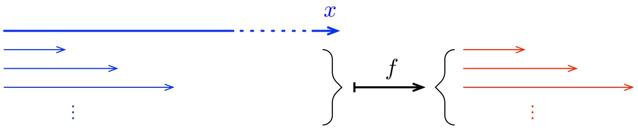
$$f_\omega(x) = \lim_{i \geq 0} f(x[0:i])$$



Continuous and Recursive Continuous ω-Translations

Any monotone function $f : \{0,1\}^* \to \{0,1\}^*$ induces *in the limit* an ω-translation $f_\omega : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by

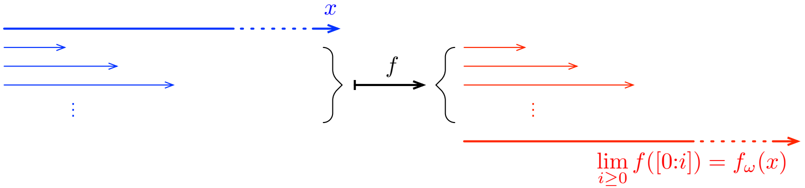
$$f_\omega(x) = \lim_{i \geq 0} f(x[0{:}i])$$



Continuous and Recursive Continuous ω-Translations

Any monotone function $f : \{0,1\}^* \to \{0,1\}^*$ induces *in the limit* an ω-translation $f_\omega : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by

$$f_\omega(x) = \lim_{i \geq 0} f(x[0{:}i])$$



Continuous and Recursive Continuous ω-Translations

- An ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ is called *continuous* if it is induced in the limit by some monotone function, i.e. if there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $\psi = f_\omega$.
- An ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ is called *recursive continuous* if it is induced in the limit by some monotone recursive function, i.e. if there exists some monotone and recursive function $f : \{0,1\}^* \to \{0,1\}^*$ such that $\psi = f_\omega$.

Continuous and Recursive Continuous ω-Translations

- An ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ is called *continuous* if it is induced in the limit by some monotone function, i.e. if there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $\psi = f_\omega$.
- An ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ is called *recursive continuous* if it is induced in the limit by some monotone recursive function, i.e. if there exists some monotone and recursive function $f : \{0,1\}^* \to \{0,1\}^*$ such that $\psi = f_\omega$.

ITMs and ITM/As Computable Functions

One has the following characterisation of the ω-translations computable by ITM and ITM/A.

Theorem 1

1. *An ω-translation is ITM-computable if an only if it is recursive continuous.*
2. *An ω-translation is ITM/A-computable if an only if it is continuous.*

ITMs and ITM/As Computable Functions

One has the following characterisation of the ω-translations computable by ITM and ITM/A.

Theorem 1

1. *An ω-translation is ITM-computable if an only if it is recursive continuous.*
2. *An ω-translation is ITM/A-computable if an only if it is continuous.*

ITMs and ITM/As Computable Functions

One has the following characterisation of the ω-translations computable by ITM and ITM/A.

Theorem 1

1. *An ω-translation is ITM-computable if an only if it is recursive continuous.*
2. *An ω-translation is ITM/A-computable if an only if it is continuous.*

Interactively Computable Functions

One has the following characterisation of all interactively computable ω-translations.

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Interactively Computable Functions

One has the following characterisation of all interactively computable ω-translations.

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\rightarrow)$"

- Let $\varphi_{\mathcal{S}}$ be some ω-translation computable by some interactive system $\mathcal{S}$.
- Consider $f : \{0,1\}^* \rightarrow \{0,1\}^*$ defined by $f(u) =$ finite word produced by $\mathcal{S}$ after exactly $|u|$ computational steps.
- Then f is monotone and one can show that $f_\omega = \varphi_{\mathcal{S}}$.
- Therefore $\varphi_{\mathcal{S}}$ is continuous.

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\rightarrow)$"

- Let $\varphi_{\mathcal{S}}$ be some ω-translation computable by some interactive system $\mathcal{S}$.
- Consider $f : \{0,1\}^* \rightarrow \{0,1\}^*$ defined by $f(u) =$ finite word produced by $\mathcal{S}$ after exactly $|u|$ computational steps.
- Then f is monotone and one can show that $f_\omega = \varphi_{\mathcal{S}}$.
- Therefore $\varphi_{\mathcal{S}}$ is continuous.

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\rightarrow)$"

- Let $\varphi_{\mathcal{S}}$ be some ω-translation computable by some interactive system $\mathcal{S}$.
- Consider $f : \{0,1\}^* \rightarrow \{0,1\}^*$ defined by $f(u) =$ finite word produced by $\mathcal{S}$ after exactly $|u|$ computational steps.
- Then f is monotone and one can show that $f_\omega = \varphi_{\mathcal{S}}$.
- Therefore $\varphi_{\mathcal{S}}$ is continuous.

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\rightarrow)$"

- Let $\varphi_{\mathcal{S}}$ be some ω-translation computable by some interactive system $\mathcal{S}$.
- Consider $f : \{0,1\}^* \rightarrow \{0,1\}^*$ defined by $f(u) =$ finite word produced by $\mathcal{S}$ after exactly $|u|$ computational steps.
- Then f is monotone and one can show that $f_\omega = \varphi_{\mathcal{S}}$.
- Therefore $\varphi_{\mathcal{S}}$ is continuous.

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\rightarrow)$"

- Let $\varphi_{\mathcal{S}}$ be some ω-translation computable by some interactive system $\mathcal{S}$.
- Consider $f : \{0,1\}^* \rightarrow \{0,1\}^*$ defined by $f(u) =$ finite word produced by $\mathcal{S}$ after exactly $|u|$ computational steps.
- Then f is monotone and one can show that $f_\omega = \varphi_{\mathcal{S}}$.
- Therefore $\varphi_{\mathcal{S}}$ is continuous.

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\leftarrow)$"

- Let ψ be some continuous ω-translation.
- Then there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$.
- One can show that there exists an ITM/A $\mathcal{M}$ such that $\varphi_{\mathcal{M}} = f_\omega = \psi$.
- Therefore ψ is ITM/A-computable, meaning that ψ is interactively computable. □

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\leftarrow)$"

- Let ψ be some continuous ω-translation.
- Then there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$.
- One can show that there exists an ITM/A $\mathcal{M}$ such that $\varphi_{\mathcal{M}} = f_\omega = \psi$.
- Therefore ψ is ITM/A-computable, meaning that ψ is interactively computable. $\square$

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\leftarrow)$"

- Let ψ be some continuous ω-translation.
- Then there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$.
- One can show that there exists an ITM/A $\mathcal{M}$ such that $\varphi_\mathcal{M} = f_\omega = \psi$.
- Therefore ψ is ITM/A-computable, meaning that ψ is interactively computable. $\square$

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\leftarrow)$"

- Let ψ be some continuous ω-translation.
- Then there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$.
- One can show that there exists an ITM/A $\mathcal{M}$ such that $\varphi_{\mathcal{M}} = f_\omega = \psi$.
- Therefore ψ is ITM/A-computable, meaning that ψ is interactively computable. □

Interactively Computable Functions

Theorem 2

An ω-translation is interactively computable if and only if it is continuous.

Proof: "$(\leftarrow)$"

- Let ψ be some continuous ω-translation.
- Then there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$.
- One can show that there exists an ITM/A $\mathcal{M}$ such that $\varphi_{\mathcal{M}} = f_\omega = \psi$.
- Therefore ψ is ITM/A-computable, meaning that ψ is interactively computable. $\square$

Universality of ITM/As

Putting together theorems 2 and 1 point (2), one obtains the universality of ITM/As for interactive computation.

Theorem 3

Let ψ be some interactively computable ω-translation. The following conditions are equivalent:

- *ψ is interactively computable;*
- *ψ is ITM/A-computable;*
- *ψ is continuous.*

Universality of ITM/As

Putting together theorems 2 and 1 point (2), one obtains the universality of ITM/As for interactive computation.

Theorem 3

Let ψ be some interactively computable ω-translation. The following conditions are equivalent:

- ψ *is interactively computable;*
- ψ *is ITM/A-computable;*
- ψ *is continuous.*

Conclusion

- We provided a machine-based and a mathematical characterisation of the interactively computable functions.
- We showed the model of an ITM/A is universal for interactive computation.
- One can show that evolving recurrent neural networks are also universal for interactive computation (ICANN 2014).
- For future work, we expect to study the characterisations of more complex interactive scenarios.

Conclusion

- We provided a machine-based and a mathematical characterisation of the interactively computable functions.
- We showed the model of an ITM/A is universal for interactive computation.
- One can show that evolving recurrent neural networks are also universal for interactive computation (ICANN 2014).
- For future work, we expect to study the characterisations of more complex interactive scenarios.

Conclusion

- We provided a machine-based and a mathematical characterisation of the interactively computable functions.
- We showed the model of an ITM/A is universal for interactive computation.
- One can show that evolving recurrent neural networks are also universal for interactive computation (ICANN 2014).
- For future work, we expect to study the characterisations of more complex interactive scenarios.

Conclusion

- We provided a machine-based and a mathematical characterisation of the interactively computable functions.
- We showed the model of an ITM/A is universal for interactive computation.
- One can show that evolving recurrent neural networks are also universal for interactive computation (ICANN 2014).
- For future work, we expect to study the characterisations of more complex interactive scenarios.