Gol and Games Laurent Regnier

Introduction Semantics

Abstract machines

Game semantics AJM and HO style Pointifixion

Go

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

A Geometry of Interaction and Game Semantics Tutorial

Laurent Regnier

Institut de Mathématiques de Luminy

Geometry of Computation 2006

・ロト ・ 戸 ・ ・ ヨ ・ ・ ヨ ・

э

Gol and Games

Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

Introduction

- Abstract operationnal semantics
- Abstract machines

Game semantics

- AJM and HO style
- Pointifixion

Geometry of interaction

• Interaction abstract machine

イロト イロト イヨト イヨト

э

- Execution formula
- Equationnal theory
- Lambda-calculus
- Execution paths



イロト イポト イヨト イヨト

э

Sac

paths



Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

Three logical levels:

- Formula: truth
- Proof: provability
- Cut elimination: coherence (subformula property)

イロト イポト イヨト イヨト

э



Introduction Semantics Abstract machines

Game semantics AJM and HO style

Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths Three logical levels:

- Formula: truth
- Proof: provability
- Cut elimination: coherence (subformula property)

イロト イポト イヨト イヨト

э



Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths Three logical levels:

- Formula: truth
- Proof: provability
- Cut elimination: coherence (subformula property)

イロト 不得 とうき とうとう

3

Curry-Howard

Gol and Games

Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths Model for sequential programming language, *e.g.*, (typed) lambda calculus: *use modularity (compositionality)*

イロト 不得 とうき とうとう

-

Sac

- Type: space
- Program: morphisms

Execution: introducing time in the model

Curry-Howard

Gol and Games

Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths Model for sequential programming language, *e.g.*, (typed) lambda calculus: *use modularity (compositionality)*

イロト 不得 とうき とうとう

3

Sac

- Type: space
- Program: morphisms

• Execution: introducing time in the model

Curry-Howard

Gol and Games

Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths Model for sequential programming language, *e.g.*, (typed) lambda calculus: *use modularity (compositionality)*

イロト 不得下 不良下 不良下

= nac

- Type: space
- Program: morphisms
- Execution: introducing time in the model

Semantics

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract

A bstract machines

Game semantics AJM and HO style

Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

Embedding syntax into more general structure

• Scott continuity: finitess of computation

イロト 不得 とうき とうとう

3

- Stability: (inverse) determinism
- Sequentiality: determinism

Semantics

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract

machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

Embedding syntax into more general structure

• Scott continuity: finitess of computation

・ロッ ・ 理 ・ ・ ヨ ・ ・ 日 ・

= nac

- Stability: (inverse) determinism
- Sequentiality: determinism

What about execution?

Syntaxes for execution

Gol and Games Laurent Regnier

Introduction Semantics

Abstract machines

Game semantics

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- Logic: cut elimination, *ie*, beta-reduction
- Programming language: abstract machines

イロト 不得 とうき とうとう

3

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract machines

Game

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths A machine for (weak) head linear reduction: $(\lambda \vec{x} x_i) \vec{u} \succ u_i$ $(\lambda \vec{x} vw) \vec{u} \succ (\lambda \vec{x} v) \vec{u} ((\lambda \vec{x} w) \vec{u})$

KAM: closures and stack

PAM: pointed sequences (hyper lazy KAM)

Execution = sequence of occurrences of variables

イロト イロト イヨト イヨト

-

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract machines

Game

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths A machine for (weak) head linear reduction: $(\lambda \vec{x} x_i) \vec{u} \succ u_i$ $(\lambda \vec{x} vw) \vec{u} \succ (\lambda \vec{x} v) \vec{u} ((\lambda \vec{x} w) \vec{u})$

• KAM: closures and stack

• PAM: pointed sequences (hyper lazy KAM)

Execution = sequence of occurrences of variables

イロト 不得下 イヨト イヨト

-

Sar

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract machines

```
Game
```

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths A machine for (weak) head linear reduction: $(\lambda \vec{x} x_i) \vec{u} \succ u_i$ $(\lambda \vec{x} vw) \vec{u} \succ (\lambda \vec{x} v) \vec{u} ((\lambda \vec{x} w) \vec{u})$

- KAM: closures and stack
- PAM: pointed sequences (hyper lazy KAM)
- Execution = sequence of occurrences of variables

イロト 不得 とうき とうとう

3

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract machines

```
Game
```

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths A machine for (weak) head linear reduction: $(\lambda \vec{x} x_i) \vec{u} \succ u_i$ $(\lambda \vec{x} vw) \vec{u} \succ (\lambda \vec{x} v) \vec{u} ((\lambda \vec{x} w) \vec{u})$

- KAM: closures and stack
- PAM: pointed sequences (hyper lazy KAM)
- Execution = sequence of occurrences of variables

イロト 不得下 不良下 不良下

-

Sar

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution

Introduction

- Abstract operationnal semantics
- Abstract machines

2 Game semantics

- AJM and HO style
- Pointifixion

Geometry of interaction

• Interaction abstract machine

イロト イロト イヨト イヨト

э

- Execution formula
- Equationnal theory
- Lambda-calculus
- Execution paths

Game in mathematics

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract machines

Game semantics

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- Game theory (economics)
- Gentzen (coherence of arithmetics: sequent calculus proof = winning strategy)

▲ロト ▲帰ト ▲ヨト ▲ヨト - ヨー のく⊙

- Descriptive set theory (determination axioms)
- Program verification
- Game semantics

Game semantics

Gol and Games Laurent Regnier

Introduction

Semantics Abstract machines

Game semantics

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- Two players: Environment (0) and Program (P)
 - Execution = alternating sequence of moves (play)

イロト 不得 とうき とうとう

3

- Program = strategy
- Type = set of plays

AJM games

Gol and Games

Laurent Regnier

Introduction Semantics Abstract machines

Game semantics

AJM and HO style Pointifizion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths • Move = finite sequence of numbers (plus multiplicative information)

イロト 不得下 不良下 不良下

-

Sac

- Strategy = function on moves (memory freeness)
- Equivalence between strategies: renumbering

Theorem

AJM strategy of M = Gol of M



Gol and Games

Laurent Regnier

Introduction Semantics

Abstract machines

Game semantics

AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- Play = pointed sequence (à la PAM)
- Strategy = function on views (innocence)

イロト 不得 とうき とうとう

э

Sac

Theorem

Proof.

HO play = PAM run

Pointifixion

Gol and Games Laurent Regnier

Introduction

- Semantics Abstract machines
- Game semantics AJM and HO style Pointifixion

Go

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- AJM play \rightsquigarrow HO play: \vec{i}, \vec{i} points on \vec{i}
- AJM strategy (memory free) \rightsquigarrow HO strategy (innocent)

イロト イロト イヨト イヨト

3

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

Introduction

- Abstract operationnal semantics
- Abstract machines

Game semantics

- AJM and HO style
- Pointifixion

Geometry of interaction

• Interaction abstract machine

イロト イロト イヨト イヨト

э

- Execution formula
- Equationnal theory
- Lambda-calculus
- Execution paths

Gol and Games Laurent Regnier

Introduction Semantics

- Abstract machines
- Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

• A reversible abstract machine (IAM)

- An interpretation of programs/proofs by operators
- An algebraic characterization of execution paths
- A localization of beta-reduction (sharing graphs)
- A generalization of multiplicative experiments
- An interpretation into a traced monoidal category

イロト イポト イヨト イヨト

nac

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- A reversible abstract machine (IAM)
- An interpretation of programs/proofs by operators
- An algebraic characterization of execution paths
- A localization of beta-reduction (sharing graphs)
- A generalization of multiplicative experiments
- An interpretation into a traced monoidal category

イロト イポト イヨト イヨト

Gol and Games Laurent Regnier

Introduction Semantics

- Abstract machines
- Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- A reversible abstract machine (IAM)
- An interpretation of programs/proofs by operators
- An algebraic characterization of *execution paths*
- A localization of beta-reduction (sharing graphs)
- A generalization of multiplicative experiments
- An interpretation into a traced monoidal category

イロト イロト イヨト イヨト

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- A reversible abstract machine (IAM)
- An interpretation of programs/proofs by operators
- An algebraic characterization of *execution paths*
- A localization of beta-reduction (sharing graphs)
- A generalization of multiplicative *experiments*
- An interpretation into a traced monoidal category

イロト イロト イヨト イヨト

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- A reversible abstract machine (IAM)
- An interpretation of programs/proofs by operators
- An algebraic characterization of execution paths
- A localization of beta-reduction (sharing graphs)
- A generalization of multiplicative experiments
- An interpretation into a traced monoidal category

イロト 不得下 不良下 不良下

-

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution paths

- A reversible abstract machine (IAM)
- An interpretation of programs/proofs by operators
- An algebraic characterization of execution paths
- A localization of beta-reduction (sharing graphs)
- A generalization of multiplicative experiments
- An interpretation into a traced monoidal category

イロト 不得 トイヨト イヨト

1

The Interaction Abstract Machine

Gol and Games

Laurent Regnier

Introduction Semantics Abstract machines

- Game semantics
- AJM and HO style Pointifixion

Gol

ΙΑΜ

Execution formula Equationnal theory Lambdacalculus Execution paths

- Program = bideterministic (reversible) automaton
- State = (B, S) + location in the graph
 - $B = box \ stack \ of \ exponential \ signatures$
 - S = balanced stack of exponential signatures + multiplicative constants P and Q
 - exponential signature = binary tree with leaves in $\{\Box, R, S\}$

Sar

• Transitions = partial transformations on (B, S)

Theorem

 $KAM \subset IAM$

Execution formula

Gol and Games Laurent Regnier

Introduction Semantics Abstract

A bst ract machines

Game semantics

AJM and HO style Pointifixion

Gol

IAM

Execution formula

Equationnal theory Lambdacalculus Execution paths • $M: A \text{ and } x: A \vdash N: B \text{ yields:}$

$$\pi = \begin{pmatrix} \pi_{AA} & 0 & 0 \\ 0 & \pi_{A^{\perp}A^{\perp}} & \pi_{A^{\perp}B} \\ 0 & \pi_{BA^{\perp}} & \pi_{BB} \end{pmatrix} \qquad \sigma = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

.

イロト 不得 トイヨト イヨト

= 900

Execution formula

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO

style Pointifixion

Gol

IAM

Execution formula

Equationnal theory Lambdacalculus Execution paths

•
$$M: A$$
 and $x: A \vdash N: B$ yields:

$$\pi = \begin{pmatrix} \pi_{AA} & 0 & 0 \\ 0 & \pi_{A^{\perp}A^{\perp}} & \pi_{A^{\perp}B} \\ 0 & \pi_{BA^{\perp}} & \pi_{BB} \end{pmatrix} \qquad \sigma = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

۲

$$(1-\sigma^2)\pi\sum_{k\geq 0}(\sigma\pi)^k(1-\sigma^2)$$

<□▶ <□▶ < □▶ < □▶ < □▶ = □ ○ ○ ○ ○

The Gol equationnal theory

Gol and Games Laurent

Regnier

Introduction

Semantics Abstract machines

Game semantics AJM and HO style

Pointifixion

Gol

IAM Execution formula

Equationnal theory

Lambdacalculus Execution paths

- Monoid with 0 generated by p, q, d, r, s, t
- Involution: $0^* = 0$, $1^* = 1$, $(uv)^* = v^*u^*$
- Morphism: !(0) = 0, !(1) = 1, !(u)!(v) = !(uv),
 !(u)* = !(u*)
- Annihilation equations: $x^*y = \delta_{xy}$ (x, y generators)

▲ロト ▲帰ト ▲ヨト ▲ヨト - ヨー のく⊙

- Commutation equations:
 - !(u)d = du
 - !(u)x = x!(u) for x = r, s
 - !(u)t = t!(!(u))

The theorem AB^*

Gol and Games Laurent Regnier

Introduction

- Semantics Abstract machines
- Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula

Equationnal theory

Lambdacalculus Execution paths \bullet Orientate equations \rightsquigarrow rewriting system

▲ロト ▲冊ト ▲ヨト ▲ヨト ヨー わえぐ

- Normal forms = 0 or AB^*
- Inverse semigroup structure

Models of the equationnal theory

Gol and Games Laurent Regnier

Introduction

- Semantics Abstract machines
- Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula

Equationnal theory

Lambdacalculus Execution paths

- Partial isometries on the Hilbert space
- \bullet Small models: partial injections on $\mathbb N$
- Partial transformations on an algebra of first order terms (clauses model, consistent semantics)

イロト 不得下 不良下 不良下

= nar

The Gol interpretation of lambda-calculus

Gol and Games Laurent Regnier

Introduction Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambda-

Lambdacalculus Execution

paths

Given M and n define the oriented graph $\mathcal{G}_n(M)$:

- Nodes: lambda and app, box nodes
- Edges: labelled with weight
- One exiting edge per free variable plus one entering edge for *M*.

イロト 不得 とうき とうとう

3

Sac

Gol of $M = \mathcal{G}_0(M)$

Variable case: $\mathcal{G}_n(x)$

Abstraction case: $\mathcal{G}_n(\lambda \times M)$



イロト イポト イヨト イヨト

Э

Application case: $\mathcal{G}_n(MN)$



▲ロト ▲課 ト ▲ 語 ト ▲ 語 ト → 語 → の Q (3)

Execution paths

Gol and Games

Laurent Regnier

Introduction

Semantics Abstract machines

Game semantics AJM and HO style Pointifixion

Gol

IAM Execution formula Equationnal theory Lambdacalculus Execution

paths

Definition

Execution paths = invariant of beta-reduction = virtual redexes

イロト イポト イヨト イヨト

Sac

Theorem

Execution paths = Regular paths = Legal paths

Corollary

Balanced execution paths = redex families