Innocent Game Semantics via Intersection Type Assignment Systems

Pietro Di Gianantonio, Marina Lenisa

Dipartimento di Matematica e Informatica
UniversitÃ di Udine

CSL'13, Torino

Aims and outline

Aim of this work:
To define intersection type assignment systems that can describe the game semantics of programs.
To transport in game semantics a construction normally used in domain semantics.

Outline.

- Introduction: intersection type assignment systems, domain semantics, game semantics.
- An intersection type assignment system for game semantics.
- Explanation of the construction.

Intersection type assignment systems (ITAS)

- Proposed 30 years ago by Coppo - Dezani.
  Mainly used for the λ-calculus,
  to characterize properties of λ-terms,
  to describe filter models for the untyped λ-calculus.
- Used also for other programming languages, a general technique with a plethora of alternative formulations and possible applications.

Main idea

A set of typing rules,

$$x_1 : t_1, \dots x_n : t_n \vdash M : t.$$

- A term (program) M can have many types.
- A type, t, represents a property of a program, M.
- Types can describe a reach set of properties.
- The semantic of a (program), M, is completely described by the set of types (t) assignable to it.
- Type grammar

$$t := C \mid t_1 \rightarrow t_2 \mid t_1 \wedge t_2.$$

Some basic rules

$$\frac{\Gamma, x : u \vdash M : t}{\Gamma \vdash \lambda x.M : u \to t} \quad \text{(abs)}$$

$$\frac{\Gamma \vdash M : u \to t \quad \Gamma \vdash N : u}{\Gamma \vdash MN : t} \quad \text{(app)}$$

$$\frac{\Gamma \vdash M : t_1 \quad \Gamma \vdash M : t_2}{\Gamma \vdash M : t_1 \wedge t_2} \quad \text{(intersection)}$$

ITAS and domain semantics

There is a duality:

- Domain semantics associates to a program, M, its interpretation, $[\![M]\!]$: a point (function) in a suitable domain (ordered set, topological space).
- ITAS, through types, describes the properties enjoyed by $[\![M]\!]$.

Types correspond to open sets in the domain of interpretation.

The set of types associated to M completely describes $[\![M]\!]$.

Stone duality

- In [Coppo,Dezani,Honsell,Longo 82] the above correspondence is presented.
- In [Abramsky 91] the correspondence is presented in a more structured way, as a form of a Stone duality.
- Stone Duality. A point, in a topological space, is described by the set of open sets containing it.
  A continuous function between topological spaces is described by the inverse function on open sets.
  The open sets of a topological space (Cantor space) define an algebra (Boolean algebra). Inverse functions are algebra homomorphisms.

ITAS advantages

- Intersection types assignment systems give effective, finitary descriptions of the semantics of terms.
- Alternative view of the domain structure.
- Useful to reason on programs.

The problem considered in this work:
to define ITAS for innocent game semantics.

Game semantics

- Domain semantics: input - output behavior of a program, programs as functions.
- Game semantics: interaction of the program with the environment, seen as an exchange of basic moves.
  The semantics of a term is a strategy describing how the term interacts with the environment.

  Intensional semantics: there are two different strategies for addition, distinguished by the order of interrogation of the arguments.

  Several categories of games, here we consider innocent games [Hyland, Ong].

  Full definability exactly characterizes the strategies describing programs.

Running example: λ-calculus

A programming language to exemplify the ITAS for games semantics.
Simple but sufficiently rich.
A simply typed λ-calculus with a ground type $\mathcal{O}$ and two constants $\bot, \top : \mathcal{O}$, and a test function $\& : \mathcal{O} \to \mathcal{O} \to \mathcal{O}$

Domain types:

$$A ::= \mathcal{O} \mid A \to A$$

Terms:

$$M ::= \bot^{\mathcal{O}} \mid \top^{\mathcal{O}} \mid \&^{\mathcal{O}\to\mathcal{O}\to\mathcal{O}} \mid x^A \mid (\lambda x^A.M^B)^{A\to B} \mid M^{A\to B}N^A$$

Call by name reduction strategy.

Intersection types for the assignment system

Simplified grammar:

Types of ground domain

$$t^{\mathcal{O}} ::= \emptyset \mid \{q_i\} \mid \{a_i\} \mid \{q_i, a_j\} \qquad i, j \in N$$

we use indexed questions q_i and answers a_i,
Types on arrow domain:

$$t^{A\to B} ::= (t^A \wedge \ldots \wedge t^A) \to t^B$$

Game ITAS

$$\frac{i \in \mathbb{N}}{\Gamma_\emptyset \vdash \top : \{q_i, a_i\}} \qquad (\top)$$

$$\frac{i \in \mathbb{N}}{\Gamma_\emptyset \vdash \& : \{q_i\} \to \emptyset^{\mathcal{O}} \to \{q_i\}} \qquad (\&_1)$$

$$\frac{i, j \in \mathbb{N}}{\Gamma_\emptyset \vdash \& : \{q_i, a_j\} \to \{q_j\} \to \{q_i\}} \qquad (\&_2)$$

$$\frac{i, j, k \in \mathbb{N}}{\Gamma_\emptyset \vdash \& : \{q_i, a_j\} \to \{q_j, a_k\} \to \{q_i, a_k\}} \qquad (\&_3)$$

Game ITAS

$$\frac{}{\Gamma_\emptyset, x : t^A \vdash x : t^A} \quad \text{(var)}$$

$$\frac{\Gamma, x : u^A \vdash M : t^B}{\Gamma \vdash \lambda x\!:\!A.M : u^A \to t^B} \quad \text{(abs)}$$

$$\frac{\Gamma \vdash M : u_1^A \wedge \ldots \wedge u_n^A \to t^B \quad \Gamma_1 \vdash N : u_1^A \quad \ldots \quad \Gamma_n \vdash N : u_n^A}{\Gamma \wedge \Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash MN : t^B} \quad \text{(app)}$$

Structural rules

The above rules almost coincide with rules used in standard ITAS (associated to domain semantics).

Main difference.
The operator $\wedge$ defines a monoid operator (associative, commutative, with identity).

- Standard ITAS, domain semantics: idempotent $\wedge$ ($t \wedge t = t$).
- Present ITAS, game semantics: not idempotent $\wedge$.

ITAS with not idempotent $\wedge$ have been considered in the literature: [De Carvalho], [Neergaard, Mairson 04]
the complexity of type inference coincides with normalization.

Type interpretation

- ITAS for denotational semantics:
  A type describes a set of points in the graph of the interpretation function, a property, an open set.
- Present ITAS for games semantics:
  a type describes a multiset of moves,
  partitioned in several subsets.

For example the type $\{q_i, a_j\} \wedge \{q_j, a_k\} \multimap \{q_i, a_k\}$
describes the multiset:

$$\begin{array}{lll} \emptyset \multimap a & q \multimap \emptyset & q \multimap \emptyset \\ \emptyset \multimap q & a \multimap \emptyset & a \multimap \emptyset \end{array}$$

Moves with the same index lie in the same partition.

Timeless games

- A type describes a position:
  a multiset of moves contained in a play, together with the justification pointers.
  the multiset of moves that the program can exchange with the environment during a computation.
  In a position, the time order relation of moves is lost.
- What happens in the game model when the order relation is removed?
- The problem has been considered in the literature, [Bailot, Danos, ... 97], [Boudes 07], [Hyland, Schalk 99].
- The present ITAS works because there exists a time forgetful functor from the category of innocent games into a category of relational models [Boudes 07].

Indexes to recover the time order

The time order can be decomposed in two parts:

- Proponent-Opponent (defines the consecutive of any Proponent move): not important, it can be substituted by the justification relation: plays become trees of P-views.
- Opponent-Proponent (defines the consecutive of any Opponent move): important, its omission leads to the identification of 2 distinct strategies (Relational models are different from game models).

In ground domain types we add indexes on moves:

$$t^{\mathcal{O}} ::= \emptyset \mid \{q_i\} \mid \{a_i\} \mid \{q_i, a_j\} \qquad i, j \in N$$

to recover, from types (positions) the Opponent-Proponent order. moves in an Opponent-Proponent pair have the same index.

Other Game Notions

Games have a rich structure.

- Alternating sequences: Proponent - Opponent alternate.
- Pending questions: answer needs to respond to the last pending questions

This structure restricts the set of possible plays.
The corresponding restriction on types, (positions), is forced by modifying (complicating) the grammar of types.

$$t_P^{A\to B} ::= t_{Or}^{!A} \to t_P^B \mid t_{Op}^{!A} \to t_P^B$$
$$t_{Or}^{A\to B} ::= t_{Or}^{!A} \to t_{Or}^B$$
$$t_{Op}^{A\to B} ::= t_{Or}^{!A} \to t_{Op}^B \mid t_{Op}^{!A} \to t_{Op}^B \mid t_P^{!A} \to t_P^B$$

Connection with game semantics.

- Types correspond to partitioned position: multiset of moves with an equivalence relations.
- Partitioned positions describe trees of P-views.
- Given a term M, the set of P-views described by types t assignable to it, $\vdash M : t$
  defines the innocent strategy associated to M by game semantics.

Further Works

The final aim: to obtain a complete description of Game Semantics in terms of intersection types.
To define a real Stone duality.

We show how the game semantics interpretation of terms can be obtained through an ITAS.
However several aspects of Game Semantics are not captured by the ITAS presentation.

- Characterization of sets of types describing innocent strategies (full definability).
  Similarly to asynchronous games, [Mellies 06], where strategies are described by set of positions.
- Define the ITAS correspondences of different categorical constructions of games.

Thank you

Thanks for your attention.