Ghica 1

Game semantics
denotational semantics

compare w/ operational sem.
(approximate, but not distorted)

Denotational Semantics
vs. Operational Semantics

op – spec. PL using rules

$t, c \rightarrow t', c'$
term config
info

symbolic step of computation

1) syntactic
2) self-contained

flexible
math - elementary

Den.

- need meta-language (semantic domain)
SD

$[\![-]\!] : PL \rightarrow SD$

well-understood math
universe
or better-understood

semantic function defined inductively
on syntax

$$[\![k\ t_0 \ldots t_i]\!] = f([\![t_0]\!], \ldots, [\![t_i]\!])$$

compositional:
meaning whole
specified as meaning of parts

Ex. Regular languages
operational sem. ~~syntactic~~ - Kleene algebras
semantic - FSA finite state automata

~~Denotational~~
payoff: reasoning about equivalence

Equivalence in PL

Oper. sem. $t \to t'$ (c stays same)

$t \equiv t'$

But is a congruence?

$\forall C[-] \qquad C[t] \overset{?}{\equiv} C[t']$
context
(doesn't have to be the case)

language could have crazy features
(reflection - inspect a term)
$\Downarrow$
equiv. will not be a congruence

compiler optimization relies on congruence

Den sem.

$[\![C[t]]\!] = [\![t]\!] \circ [\![C]\!]$ (composition in S.D. $\underset{\text{sem. domain}}{}$)

$[\![t]\!] = [\![t']\!]$ equivalence treated as equality in semantic domain

$\Leftrightarrow \forall C.\ [\![C[t]]\!] = [\![C[t']]\!]$

equality is always a congruence in SD

convenient to use "recipes" for models (category theory)

Den Sem - less obvious

Important to relate Op Sem and Den Sem.

1) Termination $t\Uparrow$ $t\Downarrow$

{ does not terminate $\leadsto$ terminates ; eval to constant

$[\![t]\!] = \bot$ $[\![t]\!] = \top$

closed program has one of two values in SD

2) Observational equivalence (not completely formal)

$t_0 \equiv t_1 \Leftrightarrow \forall C[-].\ C[t_0]\Downarrow$ iff $C[t_1]\Downarrow$

either both terminate
or both do not terminate

First key <u>Property 1</u> Soundness

$[\![t_0]\!] = [\![t_1]\!] \Rightarrow t_0 \equiv t_1$

bare minimum from a Den Sem.

<u>Prop 2</u> Adequacy

$[\![t]\!] = \top \Leftrightarrow t\Downarrow$

(terminates) (terminates in Op Sem)

rather hard to prove

<u>Prop 3</u> Definability (no garbage in Sem Dom)

$\forall \tau \in SD.\ \exists t \in PL.\ [\![t]\!] = \tau$

(den. sem. is translatable

in semantic unive, don't have meanings
that don't correspond to terms)

if all 3 props hold, then
Full Abstraction (close to completeness)

$$[\![t_0]\!] = [\![t_1]\!] \iff t_0 \equiv t_1$$

Theorem (really: recipe for Thm applies in all concrete langs)

prop 1 ; 2 ; 3 $\Rightarrow$ FA

Proof. [one direction is soundness]
Proof by contradiction
assume $[\![t_0]\!] \neq [\![t_1]\!]$

$$\exists^* \tau \in SD.\ [\![t_0]\!] \circ \tau = \top \neq \bot = [\![t_1]\!] \circ \tau$$

reason why it holds depends on choice of SD

applying definability

$$\Rightarrow \exists C.\ [\![C]\!] = \tau$$

$$[\![t_0]\!] \circ \tau = [\![t_0]\!] \circ [\![C]\!] = [\![C[t_0]]\!] = \top$$

$\Rightarrow$ (Prop 2) $\quad C[t_0]\Downarrow \qquad C[t_1]\Uparrow$

Game Semantics (den. sem.)

development came from
open problem: definability of PCF

difficult problem, PCF basic Func. ~~Lang~~

"parallel or" in den. sem not in op sem

in between eager and lazy

when one is true, cancel the other

game sem came out of effort
solved this problem

paradigm shift of den. sem.

function in PL → function in SD

change:

function as process

sequences of interactions
between term & context

(calls & returns)

full abstraction results

"game" not as in game theory

combinatorial games, like chess
whoever makes final move wins

can think of it as an <u>interaction</u>
(as opp to "game")

- Two Protagonists < P (proponent), O (opponent)

(historical names)
meaningless labels

- "moves" actions (for interaction)

	O	P
Q		
A		

question / answer
(fn calls / returns)

Ex: $0 : \mathbb{N}$

q^O

0^P (answer)

constant 0

call-by-name languages
(game sem deeply influenced by evaluation)
[call-by-name easier]

$\lambda x.x : Nat \to Nat$

q^O

q^P

m^O

m^P

"play" are interactions
(set of all possible plays : strategy
(in one run) ↳ SD

Def: arena – where the games happen

rel bet moves

$\langle M, Q, O, I, \vdash \rangle$

- M is a set of <u>moves</u>
- $Q \subseteq M$ subset of questions
- $A = M \setminus Q$ subset of answers
- $O \subseteq M$ subset of opponent moves
- $P = M \setminus O$ " proponent "
- $I \subseteq O \cap Q$ initial moves
  distinguished subset
  designated moves start
  a computation / play
- $\vdash \subseteq Q \times M$ enabling

causal structure in moves

cannot have anser not corr. to Ques.

fn ask input only if you ask for result

$m \vdash n \Rightarrow m \in O$ iff $n \in P$

$m^{n?} \notin I$

Arena of natural numbers

Ex. $\mathbb{N}at = \langle 1 + \mathbb{N}$~~$\mathbb{N}$~~$, 1, 1, 1, 1 \times \mathbb{N} \rangle$

unit arena $I = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$

Composite arenas (product arena
arrow arena)

disjoint union across all components

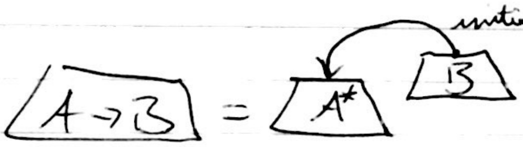
$A \times B = \langle M_A + M_B, Q_A + Q_B, O_A + O_B, I_A + I_B, \vdash_A + \vdash_B \rangle$

$\boxed{A \times B} = \boxed{A}\,\boxed{B}$ (set side-by-side)

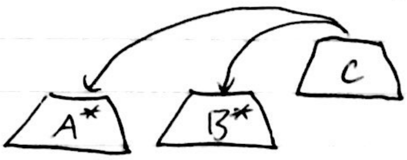
polarity switch

$A \to B = \langle M_A + M_B, Q_A + Q_B, \widetilde{P_A} + O_B, \text{inr}(I_B), \vdash_A + \vdash_B \cup \text{inr}(I_B) \times \text{inl}(I_A) \rangle$

initial here is initial there



Theorem. $A \times B \to C \cong A \to (B \to C)$



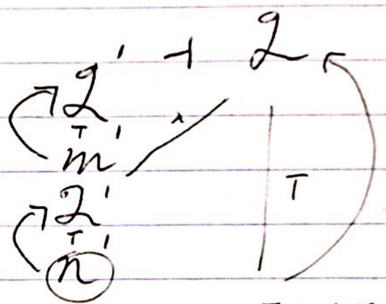
get this structure in both cases

only iso not equiv
bec + is assoc only up to iso
(tags are different)

Thm $I \times A \cong A \times I \cong A$

one side is empty but still tagging everything

$\boxed{\text{Plays}}$ $\lambda x. x+x : Nat' \to Nat$



allowed to happen (enabled)

$p = m+n$

need to know which returns corr to which calls

play = sequence + pointers

prime ' - syntactic tag

$q \; q' \; m' \; q' \; n' \; p$

move assoc w/ name of (pointers)

$q\,a\langle b\rangle \,.\, q'\,b\langle c\rangle \,.\, m'\,c \,.\, q'\,b\langle d\rangle \,.\, n'\,d \,.\, p\,b$

introduces name

points

Def (Play)

A play is a pointer seq ("justified sequence") in arena A s.t.

- $p' \cdot m a \sqsubseteq p$ (prefix)

then $\exists q \in Q_A$ s.t. $q b\langle a \rangle \in p'$

s.t. $q \vdash_A m$

- If $q^{a\langle b \rangle} \sqsubseteq p$ then $q \in I_A$

$\uparrow$ name is basically garbage

gives basic causal structure of computation

Def (Strategy) Our sem. domain to give meaning to terms

A strat $\sigma : A$ is a set of plays s.t.

$\forall p \in \sigma$

- $p' \sqsubseteq p$ then $p' \in \sigma$ (at $q^{b\langle a \rangle}$ level of granularity)

- $p \cdot m \in Play_A$, $m \in O_A$ then $p \cdot m \in \sigma$

opponent cannot do arbitrary move
can do legal moves

on names (inf. set)

- $\forall$ permutation $\Pi : A \to A$

  $\Pi \cdot p \in \sigma$ ↔ equivariance condition (bijection on names applying to all name)

  $\Pi \cdot \varepsilon = \varepsilon$

  $\Pi \cdot p \cdot m\, a\langle b\rangle = (\Pi p) \cdot m \cdot \Pi(a)\langle \Pi(b)\rangle$

  $\Pi \cdot (p \cdot ma) = (\Pi \cdot p) \cdot m\, \Pi(a)$

- names are not part of behavior
- generalization of α-equivalence

names are always fresh