OPLSS22: Game Semantics - Lecture 2

Speaker: Pierre-Louis Curien

June 24, 2022

1 Pointers

In order to evaluate a strategy we need pointers to mark the number of player moves to opponent moves. The notation $\left[m, \overset{\text{i}}{\hookleftarrow}\right]$ is used for a P move, where i is a natural number recording how many O moves one must pass until meeting the justifier. For example, considering the player's strategy defined by the high-order type $(nat_{11} \to nat_1 \to nat_\epsilon)$:

$$h = \lambda f.\ case\ f(3)\,[4 \to 7, 6 \to 9]$$

and opponent's counter-strategy defined by

$$\lambda x.case\ x\,[0 \to 3, 3 \to 6]$$

and their associated Böhm Trees:

$$q_\epsilon q_1 \begin{cases} q_{11} 3_{11} \\ 4_1 7_\epsilon \\ 6_1 9_\epsilon \end{cases} \qquad\qquad q_1 q_{11} \begin{cases} 0_{11} 3_1 \\ 3_{11} 6_1 \end{cases}$$

we can insert explicit pointers as follows:

$$q_\epsilon [q_1, \overset{0}{\hookleftarrow}] \begin{cases} q_{11} [3_{11}, \overset{0}{\hookleftarrow}] \\ 4_1 [7_\epsilon, \overset{1}{\hookleftarrow}] \\ 6_1 [9_\epsilon, \overset{1}{\hookleftarrow}] \end{cases} \qquad\qquad q_1 [q_{11}, \overset{0}{\hookleftarrow}] \begin{cases} 0_{11} [3_1, \overset{1}{\hookleftarrow}] \\ 3_{11} [6_1, \overset{1}{\hookleftarrow}] \end{cases}$$

From the easy examples (with constants) the pointers can be inferred, blue colored is associated to Player and red to Opponent moves. As expected the first pointers for both strategies are set to 0; the pointers from answer move q_1 is the one corresponding to the binding λ and q_{11} is the one corresponding to the argument 3. The pointers marked with 1 are all to the last questions.

2 Game abstract machine for PCF Böhm trees

Game abstract machines operate over strategies and counter-strategies fig. 1. A player's step $2n'$ is determined by the opponent's last move $2n-1$ in the strategy. $2n'$ points to some earlier step p. The machine then prescribes where to play the next step $2n$ in the counter-strategy, namely considering the step p'.

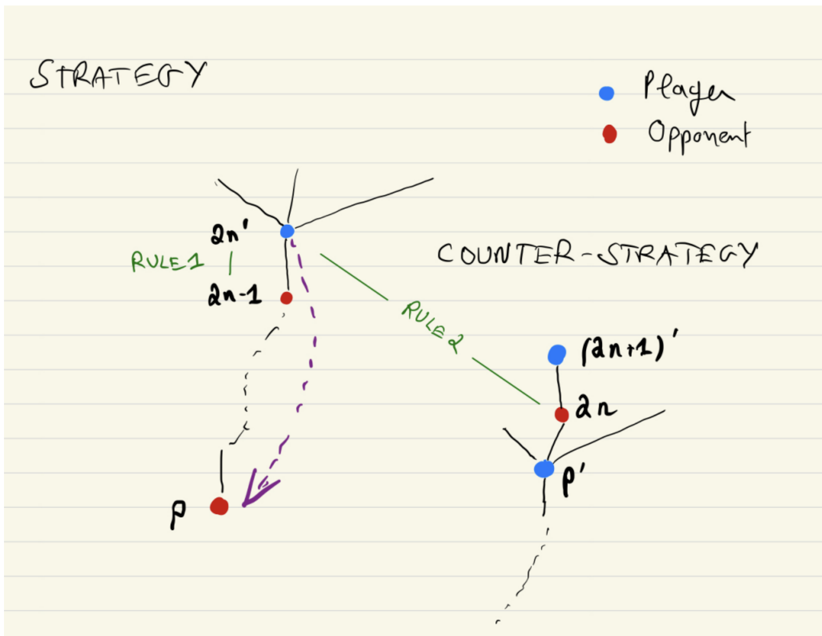

Figure 1: Tree illustrations for strategies and counter-strategies

2.1 Example

Each node of the Böhm tree is given in the form $q_n\left[q_m, \overset{p}{\hookleftarrow}\right]$. In response to the opponent's challenge q_n, the player answers q_m. We motivate game abstract machines for PCF Böhm trees via the example in fig. 2.

The first layer of the Böhm tree is $q_\epsilon\left[q_1, \overset{0}{\hookleftarrow}\right]$. This means that the opponent presents a challenge q_ϵ that the player will answer via q_1. To this end, we begin in the node $2'$. The opponent challenges with q_ϵ, and to address this challenge, the player makes move q_1. q_1 points to the Böhm tree with corresponding opponent strategy which is node $3'$. Transitioning to node $3'$, the opponent challenges with q_1. To this challenge, the player responds with move q_{11}. Continuing, the abstract machine proceeds to node $4'$, then $5'$, and lastly $6'$. In $6'$, the process terminates as the player has reached an answer to q_ϵ. In this example, q_ϵ is answered by 9, and the opponent move that enabled 9 is 6.

$$q_\epsilon \left[q_1, \overset{0}{\hookleftarrow}\right] \begin{cases} q_{11} \left[3_{11}, \overset{0}{\hookleftarrow}\right] \\ 4_1 \left[7_\epsilon, \overset{1}{\hookleftarrow}\right] \\ 6_1 \left[9_\epsilon, \overset{1}{\hookleftarrow}\right] \end{cases}$$

$$q_1 \left[q_1, \overset{0}{\hookleftarrow}\right] \begin{cases} 0_{11} \left[3_1, \overset{0}{\hookleftarrow}\right] \\ 3_{11} \left[6_1, \overset{1}{\hookleftarrow}\right] \end{cases} \qquad (\mathit{interaction})$$

$$2' : \langle q_\epsilon, 1\rangle \left[q_1, \overset{0}{\hookleftarrow}\right] \begin{cases} 4' : \langle q_{11}, 4\rangle \left[3_{11}, \overset{0}{\hookleftarrow}\right] \\ 6' : \langle 6_1, 5\rangle \left[9_\epsilon, \overset{1}{\hookleftarrow}\right] \end{cases}$$

$$3' : \langle q_1, 2\rangle \left[q_{11}, \overset{0}{\hookleftarrow}\right] \begin{cases} \\ 5' : \langle 3_{11}, 4\rangle \left[6_1, \overset{1}{\hookleftarrow}\right] \end{cases}$$

Figure 2: Example of game abstract machine for PCF Böhm trees

3 Kierstead example

Consider the terms

$$\begin{aligned} \mathit{Kierstead}_1 &= \lambda f.\mathrm{case} f(\lambda x.\mathrm{case} f(\lambda y.\mathrm{case} x)) \\ \mathit{Kierstead}_2 &= \lambda f.\mathrm{case} f(\lambda x.\mathrm{case} f(\lambda y.\mathrm{case} y)) \end{aligned}$$

where case$M$ is shorthand for case$M[T \to T, F \to F]$ for any M. The terms share the same type, namely

$$(\mathit{bool}_{111} \to \mathit{bool}_{11}) \to \mathit{bool}_1) \to \mathit{bool}_\epsilon$$

Note that the strategies for the two terms differ only in one pointer. The following diagram represents the strategies for each term: to see the strategy of $\mathit{Kierstead}_1$ let $a = 1$ to see the strategy of $\mathit{Kierstead}_1$ let $a = 0$. The discrepancy between the two is highlighted in red:

$$q_\epsilon[q_1, \overset{1}{\hookleftarrow}] \begin{cases} q_{11}[q_1, \overset{1}{\hookleftarrow}] \begin{cases} q_{11}[q_{111}, \overset{a}{\hookleftarrow}] \begin{cases} T_{111}[T_{11}, \overset{1}{\hookleftarrow}] \\ F_{111}[F_{11}, \overset{1}{\hookleftarrow}] \end{cases} \\ T_1[T_{11}, \overset{1}{\hookleftarrow}] \\ F_1[F_{11}, \overset{1}{\hookleftarrow}] \end{cases} \\ T_1[T_\epsilon, \overset{1}{\hookleftarrow}] \\ F_1[F_\epsilon, \overset{1}{\hookleftarrow}] \end{cases}$$

This example shows the importance of pointers; without their inclusion the two terms would have the same denotation.

Observe also that the strategies diagrammed above have the *well-bracketing* property described in the proceeding section.

4 Well-bracketing

The well-bracketing condition states that moves must be answered in order. The arena that represents the interpretation of PCF only admits strategies that respect this condition. For example, the following strategy is well-bracketed:

$$q_\epsilon q_1 \begin{cases} 0_1 3_\epsilon \\ 1_1 4_\epsilon \end{cases}$$

On the other hand, the following strategy (which does not correspond to any PCF Böhm tree) does not satisfy the condition, because q_ϵ may be answered before q_1 or $q_1 1$.

$$q_\epsilon q_1 \begin{cases} q_{11} 0_\epsilon \\ q_{12} 1_\epsilon \\ n_1 (n+2)_\epsilon \end{cases}$$

5 Strong (or partial) evaluation

Strong evaluation resembles a lazy or stream-like loop of evaluation. An abstract machine starts at q_ϵ and for each λ present in a given term there exists an intermediate q_i in the corresponding Böhm tree that serves as a challenge that must be answered prior to q_ϵ.

Consider the term $(\lambda x.\lambda y.\texttt{case}\ \ y\,[6 \to \texttt{case}\ \ x\,[\dots], \dots])$ and a partial evaluation of this term to the input 4.

$$(\lambda x.\lambda y.\texttt{case}\ \ y\,[6 \mapsto \texttt{case}\ \ x\,[\dots], \dots])4 \ \to\ \lambda y.\texttt{case}\ \ y\,[\dots]$$

The Böhm tree for this partial evaluation is given in fig. 3. The input 4 eliminates the binding λx leaving λy, i.e. a partial evaluation. With this, we are able to answer the remaining challenge q_1 and return 4_1. Hence the Böhm tree reduces to the one given in fig. 4. There is still one layer remaining in the Böhm tree because there is still one λ left to evaluate. The opponent may then proceed in an analogous manner to eliminate λy.

6 Arenas

An arena A is defined by a set of moves M and an enabling relation. Each move has a polarity O (opponent move) or P (player move).

$$q_\epsilon q_2 \begin{cases} 6_2 q_1 \begin{cases} 4_1 8_\epsilon \\ 2_1 1_\epsilon \end{cases} \\ 3_2 q_1 \begin{cases} 4_1 5_\epsilon \\ 1_1 6_\epsilon \end{cases} \end{cases}$$

Figure 3: Example Böhm tree for partial evaluation

$$q_\epsilon q_2 \begin{cases} 6_2 8_\epsilon \\ 3_2 5_\epsilon \end{cases}$$

Figure 4: Example Böhm tree for partial evaluation against $q_1 4_1$

The enabling relation is composed of enabling rules of the form $m \vdash n$ (the move n is enabled by m) and by initial moves $\vdash m$ (m is an initial move).

The enabling relation has two restrictions:

1. Related moves must have opposing polarity.
2. Only opponent moves can be initial moves.

6.1 Product of arenas

The product of two arenas A and B (noted $A \times B$) has as moves the disjoint union of the moves of its components: for each move m in A (resp. n in B), there is a move m_1 (resp. n_2) in $A \times B$. The polarities of the moves are preserved.

Each enabling in the product arena corresponds to an enabling in one of its components:

$$\frac{\vdash_A m}{\vdash m_1} \qquad \frac{\vdash_B n}{\vdash n_2} \qquad \frac{m \vdash_A a}{m_1 \vdash a_1} \qquad \frac{n \vdash_B b}{n_2 \vdash b_2}$$

6.2 Function space of two arenas

The arena $A \to B$ of functions between arenas A and B. Like for the product of arenas, the move set of $A \to B$ is the disjoint union of the moves of A and B, with the only difference being that the polarity of the moves from A is reversed.

The only initial moves are the ones inherited from arena B. The initial moves of the A arena are now valid responses to initial moves from B (second rule below).

$$\frac{\vdash_B n}{\vdash n_2} \qquad \frac{\vdash_A m \quad \vdash_B n}{n_2 \vdash m_1} \qquad \frac{m \vdash_A a}{m_1 \vdash a_1} \qquad \frac{n \vdash_B b}{n_2 \vdash b_2}$$