

Proiectarea Semanticii Limbajului JavaScript-PLP

Secțiunea C - Documentație Tehnică

January 6, 2026

1 Modelul Semanticii Operaționale

Semantica limbajului este definită utilizând **Semantica Operațională Big-Step**. Deoarece în acest limbaj atribuirea este o expresie, evaluarea oricărei expresii poate modifica starea programului [cite: 730, 749-750].

1.1 Starea Programului (Environment)

Starea programului, notată cu σ , este o funcție care mapează identificatorii la valorile lor:

$$\sigma \in \text{Env} = \text{Var} \rightarrow \text{Value} \cup \{\perp\}$$

unde $\text{Value} = \{\text{Int}, \text{Float}, \text{Bool}, \text{Char}, \text{String}, \text{Unit}\}$ [cite: 729].

2 Semantica Expresiilor

Evaluarea unei expresii e în starea σ produce o valoare v și o stare nouă σ' (datorită posibilelor atribuiri):

$$\langle e, \sigma \rangle \Downarrow \langle v, \sigma' \rangle$$

2.1 Reguli pentru Expresii

- **Constante:** $\langle c, \sigma \rangle \Downarrow \langle c, \sigma \rangle$
- **Variabile:** $\frac{\sigma(x)=v}{\langle x, \sigma \rangle \Downarrow \langle v, \sigma \rangle}$
- **Atribuire:** $\frac{\langle e, \sigma \rangle \Downarrow \langle v, \sigma' \rangle}{\langle x=e, \sigma \rangle \Downarrow \langle v, \sigma'[x \leftarrow v] \rangle}$ (returnează valoarea și actualizează starea) [cite: 731, 750].
- **Operații Binare:** $\frac{\langle e_1, \sigma \rangle \Downarrow \langle v_1, \sigma'' \rangle \quad \langle e_2, \sigma'' \rangle \Downarrow \langle v_2, \sigma' \rangle \quad v=v_1 \oplus v_2}{\langle e_1 \oplus e_2, \sigma \rangle \Downarrow \langle v, \sigma' \rangle}$
Notă: e_2 se evaluează în starea σ'' produsă de e_1 . [cite: 730].
- **Operații Unare (NOT, Neg):** $\frac{\langle e, \sigma \rangle \Downarrow \langle v_1, \sigma' \rangle \quad v=\ominus v_1}{\langle \ominus e, \sigma \rangle \Downarrow \langle v, \sigma' \rangle}$ [cite: 764-765].

3 Semantica Instrucțiunilor

Instrucțiunile transformă starea σ . Introducem o stare specială $Exit(v)$ pentru a gestiona instrucțiunea `return` [cite: 733-734].

4 Semantica Instrucțiunilor (Actualizată cu Block Scoping)

4.1 Reguli de Control și Vizibilitate

- **Skip:** $\langle ;, \sigma \rangle \rightarrow \sigma$.
- **Declarație (Let):** $\frac{\langle e, \sigma \rangle \Downarrow \langle v, \sigma' \rangle}{\langle \text{let } x = e, \sigma \rangle \rightarrow \sigma' [x \leftarrow v]}$ sau $\langle \text{let } x, \sigma \rangle \rightarrow \sigma [x \leftarrow \text{Unit}]$.
- **Secvențiere:** $\frac{\langle s_1, \sigma \rangle \rightarrow \sigma'' \quad \sigma'' \neq \text{Exit} \quad \langle s_2, \sigma'' \rangle \rightarrow \sigma'}{\langle s_1; s_2, \sigma \rangle \rightarrow \sigma'}$
- **Bloc (Scoping):** $\frac{\langle ss, \sigma \rangle \rightarrow \sigma_{final}}{\langle \{ss\}, \sigma \rangle \rightarrow \sigma'}$
unde starea rezultată σ' elimină variabilele locale:

$$\sigma'(x) = \begin{cases} \sigma_{final}(x) & \text{dacă } x \in \text{dom}(\sigma) \\ \perp & \text{dacă } x \notin \text{dom}(\sigma) \end{cases}$$

- **Condițional (If-Else):**

$$\frac{\langle e, \sigma \rangle \Downarrow \langle \text{true}, \sigma'' \rangle \quad \langle s_1, \sigma'' \rangle \rightarrow \sigma'}{\langle \text{if } (e) \ s_1 \text{ else } s_2, \sigma \rangle \rightarrow \sigma'}$$

$$\frac{\langle e, \sigma \rangle \Downarrow \langle \text{false}, \sigma'' \rangle \quad \langle s_2, \sigma'' \rangle \rightarrow \sigma'}{\langle \text{if } (e) \ s_1 \text{ else } s_2, \sigma \rangle \rightarrow \sigma'}$$

- **Repetiție (While):**

$$\frac{\langle e, \sigma \rangle \Downarrow \langle \text{false}, \sigma' \rangle}{\langle \text{while } (e) \ s, \sigma \rangle \rightarrow \sigma'}$$

$$\frac{\langle e, \sigma \rangle \Downarrow \langle \text{true}, \sigma'' \rangle \quad \langle s, \sigma'' \rangle \rightarrow \sigma''' \quad \langle \text{while } (e) \ s, \sigma''' \rangle \rightarrow \sigma'}{\langle \text{while } (e) \ s, \sigma \rangle \rightarrow \sigma'}$$

5 Semantica Instrucțiunii Return

Instrucțiunea **return** oprește execuția imediată a blocului curent și propagă valoarea[cite: 734, 766]:

$$\frac{\langle e, \sigma \rangle \Downarrow \langle v, \sigma' \rangle}{\langle \text{return } e, \sigma \rangle \rightarrow \text{Exit}(v)}$$

Dacă un bloc întâlnește o stare $\text{Exit}(v)$, toate instrucțiunile următoare sunt sărite.

6 Sistemul de Tipizare

Limbaajul utilizează **Strong Typing la runtime**. Orice operație între tipuri incompatibile (ex: adunarea unui **Int** cu un **String**) va opri execuția cu o eroare de tip **RuntimeError**, fără a încerca conversia automată a valorilor.