# Programski jezik PREV'26

(Boštjan Slivnik)

## 1 Leksikalna pravila

Programi v programskem jeziku PREV'26 so napisani v 7-bitnem ASCII naboru znakov.

1. Konstante:

   (a) *Celoštevilska konstanta* je neprazno zaporedje desetiških števk (od `0` do `9`) brez vodilnih ničel, pred katerim je lahko predznak (`+` ali `-`).

   (b) *Znakovna konstanta* je z enojnimi narekovaji (`'`) obdan izpisljiv znak (z ASCII kodo med vključno 32 in 126) z izjemo enojnega narekovaja (`'`) in leve poševnice (`\`). Enojni narekovaj in leva poševnica sta znotraj enojnih narekovajev uvedena z levo poševnico (zapisana torej kot `\'` in `\\`). Vsak znak je lahko zapisan tudi z dvomestno šestnajstiško ASCII kodo (šestnajstiške števke od `0` do `9` in od `A` do `F`), ki je uvedena s predpono `\x`.

   (c) *Konstanten niz* je z dvojnimi narekovaji (`"`) obdano (lahko tudi prazno) zaporedje izpisljivih znakov (z ASCII kodami med vključno 32 in 126) z izjemo dvojnega narekovaja (`"`) in leve poševnice (`\`). Dvojni narekovaj in leva poševnica sta znotraj dvojnih narekovajev uvedena z levo poševnico (zapisana torej kot `\"` in `\\`). Vsak znak je lahko zapisan tudi z dvomestno šestnajstiško ASCII kodo (šestnajstiške števke od `0` do `9` in od `A` do `F`), ki je uvedena s predpono `\x`.

2. *Simboli*:

   `. , : = + - * / % == != <= >= < > ( ) [ ] { } ^`

3. *Rezervirane besede*:

   `and as bool do char else end false fun if in int let nil none not or sizeof then true typ var void while`

4. *Imena*: Ime je neprazno zaporedje črk (od `A` do `Z` in od `a` do `z`), števk (od `0` do `9`) in podčrtajev (`_`), ki se ne začne s števko in ni rezervirana beseda.

5. *Komentarji:* Komentar se začne z dvema desnima poševnicama (`/`) in se razteza do konca vrstice (znak LF z ASCII kodo 10).

6. *Belo besedilo:* Presledek ter znaki HT (ASCII koda 9), LF (ASCII koda 10) in CR (ASCII koda 13).

Leksikalni simboli so določeni po pravilu najdaljšega ujemanja na prvem (skrajno levem) mestu.
Znak HT (ASCII koda 9) je širok (do) 8 znakov.

## 2 Sintaksna pravila

Sintaksna zgradba je opisana z množicami PROG, DEFN, TYPE in EXPR, ki vsebujejo vse sintaksno pravilne programe, definicije, tipe in izraze, zaporedoma. Pri tem $int$, $char$ in $string$ predstavljajo celoštevilske konstante, znakovne konstante in nize, zaporedoma, poleg tega pa velja še $bool = \{\texttt{true}, \texttt{false}\}$, $void = \{\texttt{none}\}$ in $ptr = \{\texttt{nil}\}$ ter $id \in \mathsf{ID}$, pri čemer je ID množica vseh imen.

**Programi:**

$$\frac{D_1, D_2, \ldots, D_n \in \mathsf{DEFN} \wedge n > 0}{D_1\ D_2\ \ldots\ D_n \in \mathsf{PROG}}\ (\textsc{syn}{:}1)$$

**Definicije:**

$$\frac{T \in \mathsf{TYPE}}{\texttt{typ}\ id\ \texttt{=}\ T \in \mathsf{DEFN}}\ (\text{SYN:2}) \qquad \frac{T \in \mathsf{TYPE}}{\texttt{var}\ id\ \texttt{:}\ T \in \mathsf{DEFN}}\ (\text{SYN:3})$$

$$\frac{T_1, T_2, \ldots, T_n,\ T \in \mathsf{TYPE} \wedge n \geq 0}{\texttt{fun}\ id\ \texttt{(}\ id_1\ \texttt{:}\ T_1\ \texttt{,}\ id_2\ \texttt{:}\ T_2\ \texttt{,}\ \ldots\ \texttt{,}\ id_n\ \texttt{:}\ T_n\ \texttt{)}\ \texttt{:}\ T \in \mathsf{DEFN}}\ (\text{SYN:4})$$

$$\frac{T_1, T_2, \ldots, T_n,\ T \in \mathsf{TYPE} \wedge n \geq 0 \qquad E_1, E_2, \ldots, E_m \in \mathsf{EXPR} \wedge m > 0}{\texttt{fun}\ id\ \texttt{(}\ id_1\ \texttt{:}\ T_1\ \texttt{,}\ id_2\ \texttt{:}\ T_2\ \texttt{,}\ \ldots\ \texttt{,}\ id_n\ \texttt{:}\ T_n\ \texttt{)}\ \texttt{:}\ T\ \texttt{=}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_m \in \mathsf{DEFN}}\ (\text{SYN:5})$$

**Tipi:**

$$\frac{T \in \{\texttt{int}, \texttt{char}, \texttt{bool}, \texttt{void}\}}{T \in \mathsf{TYPE}}\ (\text{SYN:6}) \qquad \frac{}{id \in \mathsf{TYPE}}\ (\text{SYN:7})$$

$$\frac{T \in \mathsf{TYPE}}{\texttt{[}\ int\ \texttt{]}\ T \in \mathsf{TYPE}}\ (\text{SYN:8}) \qquad \frac{T \in \mathsf{TYPE}}{\texttt{\^{}}\ T \in \mathsf{TYPE}}\ (\text{SYN:9})$$

$$\frac{T_1, T_2, \ldots, T_n \in \mathsf{TYPE} \wedge n > 0}{\texttt{(}\ id_1\ \texttt{:}\ T_1\ \texttt{,}\ id_2\ \texttt{:}\ T_2\ \texttt{,}\ \ldots\ \texttt{,}\ id_n\ \texttt{:}\ T_n\ \texttt{)} \in \mathsf{TYPE}}\ (\text{SYN:10})$$

$$\frac{T_1, T_2, \ldots, T_n \in \mathsf{TYPE} \wedge n > 0}{\texttt{\{}\ id_1\ \texttt{:}\ T_1\ \texttt{,}\ id_2\ \texttt{:}\ T_2\ \texttt{,}\ \ldots\ \texttt{,}\ id_n\ \texttt{:}\ T_n\ \texttt{\}} \in \mathsf{TYPE}}\ (\text{SYN:11})$$

$$\frac{T_1, T_2, \ldots, T_n,\ T \in \mathsf{TYPE} \wedge n \geq 0}{\texttt{(}\ \texttt{:}\ T_1\ \texttt{,}\ T_2\ \texttt{,}\ \ldots\ T_n\ \texttt{:}\ T\ \texttt{)} \in \mathsf{TYPE}}\ (\text{SYN:12})$$

$$\frac{T \in \mathsf{TYPE}}{\texttt{(}\ T\ \texttt{)} \in \mathsf{TYPE}}\ (\text{SYN:13})$$

**Izrazi:**

$$\frac{E \in \{int, bool, char, string, void, ptr\}}{E \in \mathsf{EXPR}}\ (\text{SYN:14}) \qquad \frac{}{id \in \mathsf{EXPR}}\ (\text{SYN:15})$$

$$\frac{E \in \mathsf{EXPR} \quad op \in \{\texttt{not}, \texttt{+}, \texttt{-}\}}{op\ E \in \mathsf{EXPR}}\ (\text{SYN:16})$$

$$\frac{E_1, E_2 \in \mathsf{EXPR} \quad op \in \{\texttt{=}, \texttt{and}, \texttt{or}, \texttt{+}, \texttt{-}, \texttt{*}, \texttt{/}, \texttt{\%}, \texttt{==}, \texttt{!=}, \texttt{<=}, \texttt{>=}, \texttt{<}, \texttt{>}\}}{E_1\ op\ E_2 \in \mathsf{EXPR}}\ (\text{SYN:17})$$

$$\frac{E_1, E_2 \in \mathsf{EXPR}}{E_1 \texttt{ [ } E_2 \texttt{ ]} \in \mathsf{EXPR}}\ (\textsc{syn}{:}18) \qquad \frac{E \in \mathsf{EXPR}}{\texttt{\^{}}\ E, E\ \texttt{\^{}} \in \mathsf{EXPR}}\ (\textsc{syn}{:}19) \qquad \frac{E \in \mathsf{EXPR}}{E\ \texttt{.}\ id \in \mathsf{EXPR}}\ (\textsc{syn}{:}20)$$

$$\frac{E \in \mathsf{EXPR} \quad T \in \mathsf{TYPE}}{E\ \texttt{as}\ T \in \mathsf{EXPR}}\ (\textsc{syn}{:}21) \qquad \frac{T \in \mathsf{TYPE}}{\texttt{sizeof}\ T \in \mathsf{EXPR}}\ (\textsc{syn}{:}22)$$

$$\frac{E,\ E_1, E_2, \ldots, E_n \in \mathsf{EXPR} \wedge n \geq 0}{E\ \texttt{(}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_n\ \texttt{)} \in \mathsf{EXPR}}\ (\textsc{syn}{:}23)$$

$$\frac{E,\ E_1, E_2, \ldots, E_n \in \mathsf{EXPR} \wedge n > 0}{\texttt{if}\ E\ \texttt{then}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_n\ \texttt{end} \in \mathsf{EXPR}}\ (\textsc{syn}{:}24)$$

$$\frac{E,\ E_1, E_2, \ldots, E_n,\ E'_1, E'_2, \ldots, E'_{n'} \in \mathsf{EXPR} \wedge n, n' > 0}{\texttt{if}\ E\ \texttt{then}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_n\ \texttt{else}\ E'_1\ \texttt{,}\ E'_2\ \texttt{,}\ \ldots\ \texttt{,}\ E'_{n'}\ \texttt{end} \in \mathsf{EXPR}}\ (\textsc{syn}{:}25)$$

$$\frac{E,\ E_1, E_2, \ldots, E_n \in \mathsf{EXPR} \wedge n > 0}{\texttt{while}\ E\ \texttt{do}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_n\ \texttt{end} \in \mathsf{EXPR}}\ (\textsc{syn}{:}26)$$

$$\frac{D_1, D_2,\ \ldots\ D_n \in \mathsf{DEFN} \wedge n > 0 \quad E_1, E_2, \ldots, E_m \in \mathsf{EXPR} \wedge m > 0}{\texttt{let}\ D_1\ D_2\ \ldots\ D_n\ \texttt{in}\ \ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_m\ \texttt{end} \in \mathsf{EXPR}}\ (\textsc{syn}{:}27)$$

$$\frac{E_1, E_2, \ldots, E_n \in \mathsf{EXPR} \wedge n > 0}{\texttt{(}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_n\ \texttt{)} \in \mathsf{EXPR}}\ (\textsc{syn}{:}28)$$

Prioriteto operatorjev v izrazih določa naslednja tabela:

| | | |
|---|---|---|
| postfiksni operatorji | `(·,·,...,·) [·] ^ .` | NAJVIŠJA PRIORITETA |
| prefiksni operatorji | `not + - ^` | |
| multiplikativni operatorji | `* / %` | |
| aditivni operatorji | `+ -` | |
| primerjalni operatorji | `== != < > <= >=` | |
| logični operatorji | `and` | |
| logični operatorji | `or` | |
| pretvorba tipa | `as` | |
| prirejanje | `=` | NAJNIŽJA PRIORITETA |

Primerjalni operatorji in operator prirejanja so neasociativni, ostali dvomestni operatorji so levoasociativni.

# 3 Semantična pravila

## 3.1 Imena

Funkcija

$$[\![ \cdot ]\!]_{\mathsf{BINDS}} \colon \mathsf{ID} \longrightarrow \mathsf{DEFN}$$

preslika ime v definicijo v skadu s pravili, ki veljajo za imenske prostore in dosege.

**Imenski prostori.**

1. Globalni imenski prostor vsebuje vsa imena tipov, spremenljivk, funkcij in parametrov.
2. Vsak zapis definira svoj lastni imenski prostor, ki vsebuje komponente tega zapisa.

**Dosegi.**

1. Program ustvari nov doseg.
2. Izraz `let...in...end` ustvari nov doseg.
3. Definicija funkcije ustvari nov doseg, v katerem so parameteri in morebitno telo funkcije. Ime funkcije, tipi parametrov in tip rezultata niso del novo ustvarjenega dosega.

Vsako ime, ki je definirano v določenem dosegu,

1. je v tem dosegu lahko definirano le enkrat in
2. je (razen kjer je zasenčeno z definicijo v vgnezdenem dosegu) vidno v celotnem dosegu.

## 3.2 Sistem tipov

Množica

$$
\begin{aligned}
\mathcal{T} = {} & \{\mathbf{int}, \mathbf{char}, \mathbf{bool}, \mathbf{void}\} && \text{(atomarni tipi)} \\
& \cup \{\mathbf{ptr}(\tau) \mid \tau \in \mathcal{T}\} && \text{(kazalci)} \\
& \cup \{\mathbf{arr}(n \times \tau) \mid n > 0 \wedge \tau \in \mathcal{T}\} && \text{(tabele)} \\
& \cup \{\mathbf{struct}_{ids}(\overline{\tau}) \mid n > 0 \wedge ids \in \mathsf{ID}^n \wedge \overline{\tau} \in \mathcal{T}^n\} && \text{(strukture)} \\
& \cup \{\mathbf{union}_{ids}(\overline{\tau}) \mid n > 0 \wedge ids \in \mathsf{ID}^n \wedge \overline{\tau} \in \mathcal{T}^n\} && \text{(unije)} \\
& \cup \{\mathbf{fun}(\overline{\tau} \to \tau) \mid n \geq 0 \wedge \overline{\tau} \in \mathcal{T}^n \wedge\ \tau \in \mathcal{T}\} && \text{(funkcije)} \\
& \cup \{\mathbf{name}(id, \tau) \mid id \in \mathsf{ID} \wedge \tau \in \mathcal{T}\} && \text{(poimenovani tip)}
\end{aligned}
$$

vsebuje vse tipe. Pri tem velja naslednje:

1. Tip **int** predstavlja cela števila množice $\{-2^{63} \ldots 2^{63} - 1\}$, ki so predstavljena kot 64-bitna predznačena cela števila v dvojiškem komplementu.
2. Tip **char** predstavljat znake ASCII besede, ki so predstavljeni z ASCII kodami kot 8-bitna nepredznačena cela števila v obsegu $0 \ldots 127$ (8. bit je neuporabljen).
3. Tip **bool** predstavlja vrednosti $\mathrm{true}$ in $\mathrm{false}$, ki sta kot $1$ in $0$, zaporedoma, predstavljeni kot 8-bitni nepredznačeni celi števili (uporabljen je le 1. bit).
4. Kazalec hrani naslov, katerega širina je določena z arhitekturo, a ni širsi od 64 bitov.
5. Tabela je predstavljena kot zaporedje elementov brez polnila.
6. Struktura je predstavljena kot množica komponent zloženih ena za drugo z morebitnim polnilom.
7. Unija je predstavljena kot množica na istem mestu prekrivajočih se komponent.
8. Funkcija je predstavljena kot kazalec na začetni naslov kode funkcije.

Tipi (z izjemo tipa **void**), ki nimajo pomnilniške predstavitve, niso dovoljeni.

Strukturno ekvivalenco tipov opisuje relacija $(\equiv) \subset \mathcal{T} \times \mathcal{T}$, ki je opisana z naslednjimi pravili sklepanja:

$$
\frac{\tau_1 = \tau_2}{\tau_1 \equiv \tau_2}\ \text{EQU:1} \qquad \frac{\tau_1 \equiv \tau_2}{\mathbf{name}(id, \tau_1) \equiv \tau_2}\ \text{EQU:2} \qquad \frac{\tau_1 \equiv \tau_2}{\tau_1 \equiv \mathbf{name}(id, \tau_2)}\ \text{EQU:3}
$$

$$\frac{n_1 = n_2 \quad \tau_1 \equiv \tau_2}{\mathbf{arr}(n_1 \times \tau_1) \equiv \mathbf{arr}(n_2 \times \tau_2)}\ \text{EQU:4} \qquad \frac{\tau_1 \equiv \tau_2}{\mathbf{ptr}(\tau_1) \equiv \mathbf{ptr}(\tau_2)}\ \text{EQU:5}$$

$$\frac{\tau_i \equiv \tau_i'}{\mathbf{struct}_{id_1 id_2 \ldots id_n}(\tau_1\ \tau_2\ \ldots\ \tau_n) \equiv \mathbf{struct}_{id_1' id_2' \ldots id_n'}(\tau_1'\ \tau_2'\ \ldots\ \tau_n')}\ \text{EQU:6}$$

$$\frac{\tau_i \equiv \tau_i'}{\mathbf{union}_{id_1 id_2 \ldots id_n}(\tau_1\ \tau_2\ \ldots\ \tau_n) \equiv \mathbf{union}_{id_1' id_2' \ldots id_n'}(\tau_1'\ \tau_2'\ \ldots\ \tau_n')}\ \text{EQU:7}$$

$$\frac{\tau_i \equiv \tau_i' \quad \tau \equiv \tau'}{\mathbf{fun}(\tau_1\ \tau_2\ \ldots\ \tau_n \to \tau) \equiv \mathbf{fun}(\tau_1'\ \tau_2'\ \ldots\ \tau_n' \to \tau')}\ \text{EQU:8}$$

**Pravila.** Funkciji

$$[\![ \cdot ]\!]_{\mathsf{ISTYPE}} \colon \mathsf{TYPE} \longrightarrow \mathcal{T}$$

in

$$[\![ \cdot ]\!]_{\mathsf{OFTYPE}} \colon \mathsf{PROG} \cup \mathsf{DEFN} \cup \mathsf{EXPR} \longrightarrow \mathcal{T}$$

preslikata konstrukte jezika v tipe: prva opisuje konstrukcijo tipa, druga pa pripis tipa posameznemu konstruktu jezika. Funkciji

$$[\![ \cdot ]\!]_{\mathsf{ISCONST}} \colon \mathsf{EXPR} \longrightarrow \{\mathsf{true}, \mathsf{false}\}$$

in

$$[\![ \cdot ]\!]_{\mathsf{ISADDR}} \colon \mathsf{EXPR} \longrightarrow \{\mathsf{true}, \mathsf{false}\}$$

določata, kateri izrazi so konstantni in kateri izrazi opisujejo poleg vrednosti tudi naslov, na katerem je vrednost. Funkcija

$$[\![ \cdot ]\!]_{\mathsf{VALUE}} \colon \mathsf{EXPR} \longrightarrow \{-2^{63} \ldots 2^{63} - 1\}$$

preslika izraze v vrednosti (kjer je to mogoče in potrebno).

**Programi:**

$$\frac{\begin{array}{c}[\![\ D_i\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ \exists m\colon D_m = \texttt{fun main ( ) : int = } E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_e\end{array}}{[\![\ D_1\ D_2\ \ldots\ D_d\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void}}\ \text{TYP:1}$$

**Definicije:**

$$\frac{[\![\ id\ ]\!]_{\mathsf{BINDS}} = \texttt{typ}\ id\ \texttt{=}\ T \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau}{\begin{array}{c}[\![\ \texttt{typ}\ id\ \texttt{=}\ T\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![\ id\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{name}(id, \tau)\end{array}}\ \text{TYP:2}$$

$$\frac{[\![\ id\ ]\!]_{\mathsf{BINDS}} = \texttt{var}\ id\ \texttt{:}\ T \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau \quad \tau \not\equiv \mathbf{void}}{\begin{array}{c}[\![\ \texttt{var}\ id\ \texttt{:}\ T\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![\ id\ ]\!]_{\mathsf{OFTYPE}} = \tau \quad [\![\ id\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![\ id\ ]\!]_{\mathsf{ISADDR}} = \mathsf{true}\end{array}}\ \text{TYP:3}$$

$$\frac{\begin{array}{c}[\![\ id\ ]\!]_{\mathsf{BINDS}} = \texttt{fun}\ id\ \texttt{(}\ id_1 : T_1\texttt{,}\ id_2 : T_2\texttt{,}\ \ldots\ \texttt{,}\ id_p : T_p\texttt{)} : T\ \left[\ \texttt{=}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_e\ \right]^? \\ [\![\ T_i\ ]\!]_{\mathsf{ISTYPE}} = \tau_i \quad \tau_i \equiv \tau_i' \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau \quad \tau \equiv \tau' \\ \tau_i' \in \{\mathbf{int}, \mathbf{char}, \mathbf{bool}\} \cup \{\mathbf{ptr}(\tau), \mathbf{fun}(\bar{\tau} \to \tau) | \bar{\tau} \in \mathcal{T}^* \wedge \tau \in \mathcal{T}\} \\ \tau' \in \{\mathbf{int}, \mathbf{char}, \mathbf{bool}, \mathbf{void}\} \cup \{\mathbf{ptr}(\tau), \mathbf{fun}(\bar{\tau} \to \tau) | \bar{\tau} \in \mathcal{T}^* \wedge \tau \in \mathcal{T}\} \\ [\![\ E_i\ ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T} \quad [\![\ E_e\ ]\!]_{\mathsf{OFTYPE}} = \tau'' \quad \tau' \equiv \tau'' \end{array}}{\begin{array}{c}[\![\ \texttt{fun}\ id\ \texttt{(}\ id_1 : T_1\texttt{,}\ id_2 : T_2\texttt{,}\ \ldots\ \texttt{,}\ id_p : T_p\texttt{)} : T\ \left[\ \texttt{=}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_e\ \right]^?\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![\ id\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{fun}(\tau_1\, \tau_2\, \ldots\, \tau_n \to \tau) \quad [\![\ id\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![\ id\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false} \\ [\![\ id_i\ ]\!]_{\mathsf{BINDS}} = id_i : T_i \quad [\![\ id_i\ ]\!]_{\mathsf{OFTYPE}} = \tau_i \\ [\![\ id_i\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![\ id_i\ ]\!]_{\mathsf{ISADDR}} = \mathsf{true} \end{array}}\ \text{TYP:4}$$

**Tipi:**

$$\frac{}{[\![\ \texttt{int}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{int}}\ \text{TYP:5} \qquad \frac{}{[\![\ \texttt{char}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{char}}\ \text{TYP:6}$$

$$\frac{}{[\![\ \texttt{bool}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{bool}}\ \text{TYP:7} \qquad \frac{}{[\![\ \texttt{void}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{void}}\ \text{TYP:8}$$

$$\frac{[\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau \quad \tau \in \mathcal{T} \setminus \{\mathbf{void}\}}{[\![\ \texttt{\^{}}\ T\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{ptr}(\tau)}\ \text{TYP:9}$$

$$\frac{[\![\ int\ ]\!]_{\mathsf{VALUE}} = n \quad 0 < n < 2^{63} \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau \quad \tau \not\equiv \mathbf{void}}{[\![\ \texttt{[}\ int\ \texttt{]}\ T\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{arr}(n \times \tau)}\ \text{TYP:10}$$

$$\frac{\begin{array}{c}[\![\ T_i\ ]\!]_{\mathsf{ISTYPE}} = \tau_i \quad \tau_i \not\equiv \mathbf{void} \end{array}}{\begin{array}{c}[\![\ \texttt{(}\ id_1 : T_1\ \texttt{,}\ id_2 : T_2\ \texttt{,}\ \ldots\ \texttt{,}\ id_c : T_c\ \texttt{)}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{struct}_{id_1\, id_2\, \ldots\, id_c}(\tau_1\, \tau_2\, \ldots\, \tau_c) \\ [\![\ id_i\ ]\!]_{\mathsf{BINDS}} = id_i : T_i \quad [\![\ id_i\ ]\!]_{\mathsf{OFTYPE}} = \tau_i \\ [\![\ id_i\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![\ id_i\ ]\!]_{\mathsf{ISADDR}} = \mathsf{true} \end{array}}\ \text{TYP:11}$$

$$\frac{\begin{array}{c}[\![\ T_i\ ]\!]_{\mathsf{ISTYPE}} = \tau_i \quad \tau_i \not\equiv \mathbf{void} \end{array}}{\begin{array}{c}[\![\ \texttt{\{}\ id_1 : T_1\ \texttt{,}\ id_2 : T_2\ \texttt{,}\ \ldots\ \texttt{,}\ id_c : T_c\ \texttt{\}}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{union}_{id_1\, id_2\, \ldots\, id_c}(\tau_1\, \tau_2\, \ldots\, \tau_c) \\ [\![\ id_i\ ]\!]_{\mathsf{BINDS}} = id_i : T_i \quad [\![\ id_i\ ]\!]_{\mathsf{OFTYPE}} = \tau_i \\ [\![\ id_i\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![\ id_i\ ]\!]_{\mathsf{ISADDR}} = \mathsf{true} \end{array}}\ \text{TYP:12}$$

$$\frac{\begin{array}{c}[\![\ T_i\ ]\!]_{\mathsf{ISTYPE}} = \tau_i \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau \quad \tau_i \equiv \tau_i' \quad \tau \equiv \tau' \\ \tau_i' \in \{\mathbf{int}, \mathbf{char}, \mathbf{bool}\} \cup \{\mathbf{ptr}(\tau), \mathbf{fun}(\bar{\tau} \to \tau) | \bar{\tau} \in \mathcal{T}^* \wedge \tau \in \mathcal{T}\} \\ \tau' \in \{\mathbf{int}, \mathbf{char}, \mathbf{bool}, \mathbf{void}\} \cup \{\mathbf{ptr}(\tau), \mathbf{fun}(\bar{\tau} \to \tau) | \bar{\tau} \in \mathcal{T}^* \wedge \tau \in \mathcal{T}\} \end{array}}{[\![\ \texttt{(}\ :T_1\ \texttt{,}\ T_2\ \texttt{,}\ \ldots\ \texttt{,}\ T_p : T\ \texttt{)}\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{fun}(\tau_1 \tau_2 \ldots \tau_p \to \tau)}\ \text{TYP:13}$$

**Izrazi:**

$$\frac{}{\begin{array}{c}[\![\ int\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{int} \\ [\![\ int\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ int\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:14} \qquad \frac{}{\begin{array}{c}[\![\ char\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{char} \\ [\![\ char\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ char\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:15}$$

$$\frac{}{\begin{array}{c}[\![\ \texttt{true}\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{bool} \\ [\![\ \texttt{true}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ \texttt{true}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:16} \qquad \frac{}{\begin{array}{c}[\![\ \texttt{false}\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{bool} \\ [\![\ \texttt{false}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ \texttt{false}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:17}$$

$$\frac{}{\begin{array}{c}[\![\ string\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{ptr}(\mathbf{char}) \\ [\![\ string\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ string\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:18} \qquad \frac{}{\begin{array}{c}[\![\ \texttt{nil}\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{ptr}(\mathbf{void}) \\ [\![\ \texttt{nil}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ \texttt{nil}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:19}$$

$$\frac{}{\begin{array}{c}[\![\ \texttt{none}\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![\ \texttt{none}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![\ \texttt{none}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:20}$$

$$\frac{[\![\ E\ ]\!]_{\mathsf{OFTYPE}} = \tau \quad \tau \equiv \mathbf{int} \quad \mathrm{op} \in \{\texttt{+}, \texttt{-}\}}{[\![\ \mathrm{op}\ E\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{int} \quad [\![\ \mathrm{op}\ E\ ]\!]_{\mathsf{ISCONST}} = [\![\ E\ ]\!]_{\mathsf{ISCONST}} \quad [\![\ \mathrm{op}\ E\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}}\ \text{TYP:21}$$

$$\frac{[\![\ E\ ]\!]_{\mathsf{OFTYPE}} = \tau \quad \tau \equiv \mathbf{bool} \quad \mathrm{op} \in \{\texttt{not}\}}{[\![\ \mathrm{op}\ E\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{bool} \quad [\![\ \mathrm{op}\ E\ ]\!]_{\mathsf{ISCONST}} = [\![\ E\ ]\!]_{\mathsf{ISCONST}} \quad [\![\ \mathrm{op}\ E\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}}\ \text{TYP:22}$$

$$\frac{\begin{array}{c}[\![\ E_1\ ]\!]_{\mathsf{OFTYPE}} = \tau_1 \quad [\![\ E_2\ ]\!]_{\mathsf{OFTYPE}} = \tau_2 \quad \mathrm{op} \in \{\texttt{and}, \texttt{or}\} \\ \tau_1 \equiv \mathbf{bool} \quad \tau_2 \equiv \mathbf{bool}\end{array}}{\begin{array}{c}[\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{bool} \\ [\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{ISCONST}} = [\![\ E_1\ ]\!]_{\mathsf{ISCONST}} \wedge [\![\ E_2\ ]\!]_{\mathsf{ISCONST}} \quad [\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:23}$$

$$\frac{\begin{array}{c}[\![\ E_1\ ]\!]_{\mathsf{OFTYPE}} = \tau_1 \quad [\![\ E_2\ ]\!]_{\mathsf{OFTYPE}} = \tau_2 \quad \mathrm{op} \in \{\texttt{*}, \texttt{/}, \texttt{\%}, \texttt{+}, \texttt{-}\} \\ \tau_1 \equiv \mathbf{int} \quad \tau_2 \equiv \mathbf{int}\end{array}}{\begin{array}{c}[\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{int} \\ [\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{ISCONST}} = [\![\ E_1\ ]\!]_{\mathsf{ISCONST}} \wedge [\![\ E_2\ ]\!]_{\mathsf{ISCONST}} \quad [\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:24}$$

$$\frac{\begin{array}{c}[\![\ E_1\ ]\!]_{\mathsf{OFTYPE}} = \tau_1 \quad [\![\ E_2\ ]\!]_{\mathsf{OFTYPE}} = \tau_2 \quad \mathrm{op} \in \{\texttt{==}, \texttt{!=}, \texttt{<}, \texttt{>}, \texttt{<=}, \texttt{>=}\} \\ \tau_1 \equiv \tau_1' \quad \tau_2 \equiv \tau_2' \quad \tau_1 \equiv \tau_2 \\ \tau_1', \tau_2' \in \{\mathbf{int}, \mathbf{char}, \mathbf{bool}\} \cup \{\mathbf{ptr}(\tau), \mathbf{fun}(\bar{\tau} \to \tau) | \bar{\tau} \in \mathcal{T}^* \wedge \tau \in \mathcal{T}\}\end{array}}{\begin{array}{c}[\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{bool} \\ [\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{ISCONST}} = [\![\ E_1\ ]\!]_{\mathsf{ISCONST}} \wedge [\![\ E_2\ ]\!]_{\mathsf{ISCONST}} \quad [\![\ E_1\ \mathrm{op}\ E_2\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:25}$$

$$\frac{[\![ E_1 ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{arr}(n \times \tau_1) \quad [\![ E_2 ]\!]_{\mathsf{OFTYPE}} = \tau_2 \quad \tau_2 \equiv \mathbf{int} \quad [\![ E_1 ]\!]_{\mathsf{ISADDR}} = \mathsf{true}}{\begin{array}{c}[\![ E_1 \texttt{ [ } E_2 \texttt{ ] } ]\!]_{\mathsf{OFTYPE}} = \tau_1 \quad [\![ E_1 \texttt{ [ } E_2 \texttt{ ] } ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![ E_1 \texttt{ [ } E_2 \texttt{ ] } ]\!]_{\mathsf{ISADDR}} = \mathsf{true}\end{array}} \text{ TYP:26}$$

$$\frac{[\![ E ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{ptr}(\tau) \quad [\![ E ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad \tau \not\equiv \mathbf{void}}{\begin{array}{c}[\![ E \texttt{ \^{} } ]\!]_{\mathsf{OFTYPE}} = \tau \quad [\![ E \texttt{ \^{} } ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![ E \texttt{ \^{} } ]\!]_{\mathsf{ISADDR}} = \mathsf{true}\end{array}} \text{ TYP:27}$$

$$\frac{[\![ E ]\!]_{\mathsf{OFTYPE}} = \tau \quad [\![ E ]\!]_{\mathsf{ISADDR}} = \mathsf{true} \quad \tau \not\equiv \mathbf{void}}{\begin{array}{c}[\![ \texttt{\^{}} E ]\!]_{\mathsf{OFTYPE}} = \mathbf{ptr}(\tau) \quad [\![ \texttt{\^{}} E ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![ \texttt{\^{}} E ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}} \text{ TYP:28}$$

$$\frac{[\![ E ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{struct}_{id_1 id_2 \ldots id_c}(\tau_1 \tau_2 \ldots \tau_c) \quad [\![ E ]\!]_{\mathsf{ISADDR}} = \mathsf{true} \quad id = id_i}{\begin{array}{c}[\![ E \texttt{ . } id ]\!]_{\mathsf{OFTYPE}} = \tau_i \quad [\![ E \texttt{ . } id ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![ E \texttt{ . } id ]\!]_{\mathsf{ISADDR}} = \mathsf{true}\end{array}} \text{ TYP:29}$$

$$\frac{[\![ E ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{union}_{id_1 id_2 \ldots id_c}(\tau_1 \tau_2 \ldots \tau_c) \quad [\![ E ]\!]_{\mathsf{ISADDR}} = \mathsf{true} \quad id = id_i}{\begin{array}{c}[\![ E \texttt{ . } id ]\!]_{\mathsf{OFTYPE}} = \tau_i \quad [\![ E \texttt{ . } id ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \quad [\![ E \texttt{ . } id ]\!]_{\mathsf{ISADDR}} = \mathsf{true}\end{array}} \text{ TYP:30}$$

$$\frac{[\![ E ]\!]_{\mathsf{OFTYPE}} = \mathbf{fun}(\tau_1 \tau_2 \ldots \tau_a \to \tau) \quad [\![ E_i ]\!]_{\mathsf{OFTYPE}} = \tau_i' \quad \tau_i' \equiv \tau_i}{\begin{array}{c}[\![ E \texttt{ ( } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_a \texttt{ ) } ]\!]_{\mathsf{OFTYPE}} = \tau \\ [\![ E \texttt{ ( } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_a \texttt{ ) } ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \\ [\![ E \texttt{ ( } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_a \texttt{ ) } ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}} \text{ TYP:31}$$

$$\frac{[\![ T ]\!]_{\mathsf{ISTYPE}} \not\equiv \mathbf{void}}{\begin{array}{c}[\![ \texttt{sizeof } T ]\!]_{\mathsf{OFTYPE}} = \mathbf{int} \\ [\![ \texttt{sizeof } T ]\!]_{\mathsf{ISCONST}} = \mathsf{true} \\ [\![ \texttt{sizeof } T ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}} \text{ TYP:32} \qquad \frac{[\![ E ]\!]_{\mathsf{OFTYPE}} \not\equiv \mathbf{void} \quad [\![ T ]\!]_{\mathsf{ISTYPE}} \not\equiv \mathbf{void}}{\begin{array}{c}[\![ E \texttt{ as } T ]\!]_{\mathsf{OFTYPE}} = [\![ T ]\!]_{\mathsf{ISTYPE}} \\ [\![ E \texttt{ as } T ]\!]_{\mathsf{ISCONST}} = [\![ E ]\!]_{\mathsf{ISCONST}} \\ [\![ E \texttt{ as } T ]\!]_{\mathsf{ISADDR}} = [\![ E ]\!]_{\mathsf{ISADDR}}\end{array}} \text{ TYP:33}$$

$$\frac{[\![ E_i ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T}}{\begin{array}{c}[\![ \texttt{ ( } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_e \texttt{ ) } ]\!]_{\mathsf{OFTYPE}} = [\![ E_e ]\!]_{\mathsf{OFTYPE}} \\ [\![ \texttt{ ( } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_e \texttt{ ) } ]\!]_{\mathsf{ISCONST}} = [\![ E_1 ]\!]_{\mathsf{ISCONST}} \wedge [\![ E_2 ]\!]_{\mathsf{ISCONST}} \wedge \ldots \wedge [\![ E_e ]\!]_{\mathsf{ISCONST}} \\ [\![ \texttt{ ( } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_e \texttt{ ) } ]\!]_{\mathsf{ISADDR}} = [\![ E_e ]\!]_{\mathsf{ISADDR}}\end{array}} \text{ TYP:34}$$

$$\frac{\begin{array}{c}[\![ E_1 ]\!]_{\mathsf{OFTYPE}} = \tau_1 \quad [\![ E_2 ]\!]_{\mathsf{OFTYPE}} = \tau_2 \quad \tau_1 \equiv \tau_2 \\ \tau_1, \tau_2 \in \{\mathbf{int}, \mathbf{char}, \mathbf{bool}\} \cup \{\mathbf{ptr}(\tau), \mathbf{fun}(\bar{\tau} \to \tau) | \bar{\tau} \in \mathcal{T}^* \wedge \tau \in \mathcal{T}\} \\ [\![ E_1 ]\!]_{\mathsf{ISADDR}} = \mathsf{true}\end{array}}{\begin{array}{c}[\![ E_1 \texttt{ = } E_2 ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![ E_1 \texttt{ = } E_2 ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \\ [\![ E_1 \texttt{ = } E_2 ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}} \text{ TYP:35}$$

$$\frac{[\![ E ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{bool} \quad [\![ E_i ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T}}{\begin{array}{c}[\![ \texttt{while } E \texttt{ do } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_e \texttt{ end } ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![ \texttt{while } E \texttt{ do } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_e \texttt{ end } ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \\ [\![ \texttt{while } E \texttt{ do } E_1 \texttt{ , } E_2 \texttt{ , } \ldots \texttt{ , } E_e \texttt{ end } ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}} \text{ TYP:36}$$

$$\frac{[\![\ E\ ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{bool} \quad [\![\ E_i\ ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T}}{\begin{array}{c}[\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{end}\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{end}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \\ [\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{end}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:37}$$

$$\frac{[\![\ E\ ]\!]_{\mathsf{OFTYPE}} \equiv \mathbf{bool} \quad [\![\ E_i\ ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T} \quad [\![\ E'_i\ ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T}}{\begin{array}{c}[\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{else}\ E'_1\ ,\ E'_2\ ,\ \ldots\ ,\ E'_{e'}\ \texttt{end}\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \\ [\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{else}\ E'_1\ ,\ E'_2\ ,\ \ldots\ ,\ E'_{e'}\ \texttt{end}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \\ [\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{else}\ E'_1\ ,\ E'_2\ ,\ \ldots\ ,\ E'_{e'}\ \texttt{end}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:38}$$

$$\frac{[\![\ D_i\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{void} \quad [\![\ E_i\ ]\!]_{\mathsf{OFTYPE}} \in \mathcal{T} \quad [\![\ E_e\ ]\!]_{\mathsf{OFTYPE}} = \tau}{\begin{array}{c}[\![\ \texttt{let}\ D_1\ D_2\ \ldots\ D_d\ \texttt{in}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{end}\ ]\!]_{\mathsf{OFTYPE}} = \tau \\ [\![\ \texttt{let}\ D_1\ D_2\ \ldots\ D_d\ \texttt{in}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{end}\ ]\!]_{\mathsf{ISCONST}} = \mathsf{false} \\ [\![\ \texttt{let}\ D_1\ D_2\ \ldots\ D_d\ \texttt{in}\ E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_e\ \texttt{end}\ ]\!]_{\mathsf{ISADDR}} = \mathsf{false}\end{array}}\ \text{TYP:39}$$

## 3.3 Operacijska semantika

Operacijsko semantiko opišemo s funkcijami

$$[\![\ \cdot\ ]\!]_{\mathrm{ADDR}} : \mathsf{EXPR} \times \mathcal{M} \to \mathcal{I} \times \mathcal{M}$$
$$[\![\ \cdot\ ]\!]_{\mathrm{EXPR}} : \mathsf{EXPR} \times \mathcal{M} \to \mathcal{I} \times \mathcal{M}$$

Pri tem $\mathcal{I}$ predstavlja 64-bitna predznačena števila v dvojiškem komplementu, $\mathcal{M}$ pa stanja pomnilnika.

Funkcija addr vrne bodisi absolutni naslov statične spremenljivke ali niza bodisi odmik lokalne spremenljivke, parametra ali komponente zapisa. Funckija sizeof vrne velikost podatkovnega tipa. Funkcija val vrne celoštevilsko vrednost konstante.

**Naslovi.**

$$\frac{}{[\![\ \mathit{string}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle \mathrm{addr}(\mathit{string}), \mathrm{M} \rangle}\ \text{SEM:1}$$

$$\frac{\mathrm{addr}(\mathit{identifier}) = a}{[\![\ \mathit{identifier}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle a, \mathrm{M} \rangle}\ \text{SEM:2}$$

$$\frac{[\![\ E_1\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n_1, \mathrm{M}' \rangle \quad [\![\ E_2\ ]\!]^{\mathrm{M}'}_{\mathrm{EXPR}} = \langle n_2, \mathrm{M}'' \rangle \quad [\![\ E_1\ ]\!]_{\mathsf{OFTYPE}} = \mathbf{arr}(n \times \tau)}{[\![\ E_1 \texttt{[} E_2 \texttt{]}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n_1 + n_2 * \mathrm{sizeof}(\tau), \mathrm{M}'' \rangle}\ \text{SEM:3}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n_1, \mathrm{M}' \rangle}{[\![\ E\,.\,\mathit{identifier}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n_1 + \mathrm{addr}(\mathit{identifier}), \mathrm{M}' \rangle}\ \text{SEM:4}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}' \rangle}{[\![\ E\texttt{\^{}}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n, \mathrm{M}' \rangle}\ \text{SEM:5}$$

**Izrazi.**

$$\frac{}{[\![\ \texttt{nil}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle 0, \mathrm{M}\rangle}\ \text{SEM:6}$$

$$\frac{}{[\![\ \texttt{true}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle 1, \mathrm{M}\rangle}\ \text{SEM:7} \qquad \frac{}{[\![\ \texttt{false}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle 0, \mathrm{M}\rangle}\ \text{SEM:8}$$

$$\frac{}{[\![\ \mathit{char}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathrm{val}(\mathit{char}), \mathrm{M}\rangle}\ \text{SEM:9} \qquad \frac{}{[\![\ \mathit{int}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathrm{val}(\mathit{int}), \mathrm{M}\rangle}\ \text{SEM:10}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle \quad \mathrm{op} \in \{\texttt{not}, \texttt{+}, \texttt{-}\}}{[\![\ \mathrm{op}\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathrm{op}\ n, \mathrm{M}'\rangle}\ \text{SEM:11}$$

$$\frac{\begin{array}{c}[\![\ E_1\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}'\rangle \quad [\![\ E_2\ ]\!]^{\mathrm{M}'}_{\mathrm{EXPR}} = \langle n_2, \mathrm{M}''\rangle \\ \mathrm{op} \in \{\texttt{or}, \texttt{and}, \texttt{==}, \texttt{!=}, \texttt{<}, \texttt{>}, \texttt{<=}, \texttt{>=}, \texttt{+}, \texttt{-}, \texttt{*}, \texttt{/}, \texttt{\%}\}\end{array}}{[\![\ E_1\ \mathrm{op}\ E_2\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n_1\ \mathrm{op}\ n_2, \mathrm{M}''\rangle}\ \text{SEM:12}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n, \mathrm{M}'\rangle}{[\![\ \texttt{\^{}}E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle}\ \text{SEM:13} \qquad \frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle}{[\![\ E\texttt{\^{}}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathrm{M}'[n], \mathrm{M}'\rangle}\ \text{SEM:14}$$

$$\frac{\mathrm{addr}(\mathit{identifier}) = a}{[\![\ \mathit{identifier}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle M[a], \mathrm{M}\rangle}\ \text{SEM:15} \qquad \frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle}{[\![\ \texttt{(}E\texttt{)}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle}\ \text{SEM:16}$$

$$\frac{[\![\ E_1\texttt{[}E_2\texttt{]}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle a, M'\rangle}{[\![\ E_1\texttt{[}E_2\texttt{]}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle M'[a], \mathrm{M}'\rangle}\ \text{SEM:17} \qquad \frac{[\![\ E\texttt{.}\mathit{identifier}\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle a, M'\rangle}{[\![\ E\texttt{.}\mathit{identifier}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle M'[a], \mathrm{M}'\rangle}\ \text{SEM:18}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathit{fun}, \mathrm{M}_0\rangle \quad [\![\ E_1\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}_1\rangle\ \ldots\ [\![\ E_m\ ]\!]^{\mathrm{M}_{m-1}}_{\mathrm{EXPR}} = \langle n_m, \mathrm{M}_m\rangle}{[\![\ E\texttt{(}E_1\texttt{,}\ \ldots\ \texttt{,}E_m\texttt{)}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathit{fun}(n_1, \ldots, n_m), \mathrm{M}_m\rangle}\ \text{SEM:19}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau \quad \tau \not\equiv \mathbf{bool} \wedge \tau \not\equiv \mathbf{char}}{[\![\ E\ \texttt{as}\ T\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle}\ \text{SEM:20}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{bool}}{[\![\ E\ \texttt{as}\ T\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n \bmod 2, \mathrm{M}'\rangle}\ \text{SEM:21}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n, \mathrm{M}'\rangle \quad [\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \mathbf{char}}{[\![\ E\ \texttt{as}\ T\texttt{\}}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle n \bmod 256, \mathrm{M}'\rangle}\ \text{SEM:22}$$

$$\frac{[\![\ T\ ]\!]_{\mathsf{ISTYPE}} = \tau}{[\![\ \texttt{sizeof}\ T\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathrm{sizeof}(T), \mathrm{M} \rangle}\ \text{SEM:23}$$

$$\frac{[\![\ E_1\ ]\!]^{\mathrm{M}}_{\mathrm{ADDR}} = \langle n_1, \mathrm{M}' \rangle \quad [\![\ E_2\ ]\!]^{\mathrm{M}'}_{\mathrm{EXPR}} = \langle n_2, \mathrm{M}'' \rangle \quad \forall a\colon \mathrm{M}'''[a] = \begin{cases} n_2 & a = n_1 \\ M''[a] & \text{otherwise} \end{cases}}{[\![\ E_1 \texttt{ = } E_2\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle 0, \mathrm{M}''' \rangle}\ \text{SEM:24}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathbf{true}, \mathrm{M}_0 \rangle \quad [\![\ E_1\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}_1 \rangle \ \ldots\ [\![\ E_s\ ]\!]^{\mathrm{M}_{s-1}}_{\mathrm{EXPR}} = \langle n_s, \mathrm{M}_s \rangle}{[\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{end}\ ]\!]_{\mathrm{EXPR}} = \langle 0, \mathrm{M}_s \rangle}\ \text{SEM:25}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathbf{false}, \mathrm{M}_0 \rangle}{[\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{end}\ ]\!]_{\mathrm{EXPR}} = \langle 0, \mathrm{M}_0 \rangle}\ \text{SEM:26}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathbf{true}, \mathrm{M}_0 \rangle \quad [\![\ E_1\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}_1 \rangle \ \ldots\ [\![\ E_s\ ]\!]^{\mathrm{M}_{s-1}}_{\mathrm{EXPR}} = \langle n_s, \mathrm{M}_s \rangle}{[\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{else}\ E'_1\ \texttt{,}\ E'_2\ \texttt{,}\ \ldots\ \texttt{,}\ E'_{s'}\ \texttt{end}\ ]\!]_{\mathrm{EXPR}} = \langle 0, \mathrm{M}_s \rangle}\ \text{SEM:27}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathbf{false}, \mathrm{M}_0 \rangle \quad [\![\ E'_1\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}_1 \rangle \ \ldots\ [\![\ E'_{s'}\ ]\!]^{\mathrm{M}_{s'-1}}_{\mathrm{EXPR}} = \langle n_{s'}, \mathrm{M}_{s'} \rangle}{[\![\ \texttt{if}\ E\ \texttt{then}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{else}\ E'_1\ \texttt{,}\ E'_2\ \texttt{,}\ \ldots\ \texttt{,}\ E'_{s'}\ \texttt{end}\ ]\!]_{\mathrm{EXPR}} = \langle 0, \mathrm{M}_{s'} \rangle}\ \text{SEM:28}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathbf{true}, \mathrm{M}_0 \rangle \quad [\![\ E_1\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}_1 \rangle \ \ldots\ [\![\ E_s\ ]\!]^{\mathrm{M}_{s-1}}_{\mathrm{EXPR}} = \langle n_s, \mathrm{M}_s \rangle}{[\![\ \texttt{while}\ E\ \texttt{do}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{end}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = [\![\ \texttt{while}\ E\ \texttt{do}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{end}\ ]\!]^{\mathrm{M}_s}_{\mathrm{EXPR}}}\ \text{SEM:29}$$

$$\frac{[\![\ E\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle \mathbf{false}, \mathrm{M}_0 \rangle}{[\![\ \texttt{while}\ E\ \texttt{do}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{end}\ ]\!]^{\mathrm{M}}_{\mathrm{EXPR}} = \langle 0, \mathrm{M}_0 \rangle}\ \text{SEM:30}$$

$$\frac{[\![\ E_1\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_1, \mathrm{M}_1 \rangle \ \ldots\ [\![\ E_s\ ]\!]^{\mathrm{M}_{s-1}}_{\mathrm{EXPR}} = \langle n_s, \mathrm{M}_s \rangle}{[\![\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ ]\!]^{\mathrm{M}_0}_{\mathrm{EXPR}} = \langle n_s, \mathrm{M}_s \rangle}\ \text{SEM:31}$$

$$\frac{}{[\![\ \texttt{let}\ D_1\ D_2\ \ldots\ D_n\ \texttt{in}\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ \texttt{end}\ ]\!]^{M}_{\mathrm{EXPR}} = [\![\ E_1\ \texttt{,}\ E_2\ \texttt{,}\ \ldots\ \texttt{,}\ E_s\ ]\!]^{M}_{\mathrm{EXPR}}}\ \text{SEM:32}$$

Operacijska semantika vseh ostalih jezikovnih konstruktov je nedefinirana – prevajalnik jih lahko implementira na kakršenkoli način.