Semantics of DSL

William Steingartner

william.steingartner@tuke.sk

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovakia

Semantics of programming languages

Spring 2022/2023

Theme 08 – Semantics of DSL

Domain-specific languages

- A domain-specific language is a programming language that is focused on a limited, specific problem domain through an appropriate abstraction and vocabulary.
- Domain-specific languages provide a way to bridge the gap between a solution to a problem expressed in terms of the domain and its computer-executable implementation.
- However, the development of such languages is still a complex and lengthy task.

Semantics of DSL

- DSLs are often implemented as an embedded language,
- semantics is expressed in the host (superior) language,
- very often operational semantics are used,
- observing the behavior of the program is not easy,
- semantic expression for DSL is often not formal enough.

The language

A language for describing robot movement in a grid.

Syntactic domains:

$n \in \mathbf{Num}$ – syntactic domain of numerals,
$S \in \mathbf{Statm}$ – syntactic domain of statements.

Production rule:

$$\begin{aligned} S ::= \quad & \mathbf{left} \mid \mathbf{right} \mid \mathbf{up} \mid \mathbf{down} \mid \\ & \mathbf{left}\ n \mid \mathbf{right}\ n \mid \mathbf{up}\ n \mid \mathbf{down}\ n \mid \\ & \mathbf{reset} \mid \mathbf{skip} \mid S; S. \end{aligned}$$

Semantics of the language

Semantic domains

- Numerals representing natural numbers express the number of steps the robot should take in a given direction.
- We express the semantics of numerals in the traditional way – using a simple semantic function:

$$\mathscr{N} : \mathbf{Num} \to \mathbf{N}_0.$$

- The current state is expressed by the coordinates where the robot is currently located. We assume a two-dimensional grid on which the robot moves, so the coordinates are in the form (x, y).
- We introduce a semantic domain $\mathbf{Point}$:

$$\mathbf{Point} = \mathbf{Z} \times \mathbf{Z}.$$

- We choose the initial state, i.e. the starting point:

$$p^* = (\mathbf{1}, \mathbf{1}).$$

Semantics of the language

A change in the robot's position is considered a state change. This change is expressed by new coordinates. For example, if the robot takes one step to the left, the horizontal coordinate is decremented.

$$p' = p\left(\pi_1(p) \ominus \mathbf{1}, \pi_2(p)\right),$$

where

- π_1 and π_2 are projections over the cartesian product:

$$\pi_1 : A \times B \to A,$$
$$\pi_2 : A \times B \to B,$$

- p is the starting position of the robot,
- p' is the new position where the robot has moved,
- $p = (x_0, y_0)$ expresses a specific point and its coordinates,
- forms $p' = p\left(f(x_0), y_0\right)$ or (alternatively for other coordinates) $p' = p\left(x_0, f(y_0)\right)$ represent an update of the given position, where some operation f is applied to the coordinates from position p,
- symbols $+$, $-$ represent operator notations, symbols $\oplus$, $\ominus$ (or $\otimes$) express real mathematical operations.

Denotational semantics

In the denotational semantics of the robot movement program, we are interested in the end of the path, i.e. the target coordinates of the robot.

We are currently not dealing with solid obstacles (e.g. walls). In the extension of the language, it would be appropriate to perform a further static analysis of the commands to check whether the robot e.g. does not attempt to cross walls (obstacles).

We define the denotation of commands using a semantic function

$$\mathscr{S}_{\mathrm{ds}} : \mathbf{Statm} \to (\mathbf{Point} \to \mathbf{Point}) \, .$$

Denotational semantics

$$\mathscr{S}_{\text{ds}}[\![\textbf{left}]\!](x,y) = (x \ominus \mathbf{1}, y) \qquad \mathscr{S}_{\text{ds}}[\![\textbf{left } n]\!](x,y) = (x \ominus \mathscr{N}[\![n]\!], y)$$

$$\mathscr{S}_{\text{ds}}[\![\textbf{right}]\!](x,y) = (x \oplus \mathbf{1}, y) \qquad \mathscr{S}_{\text{ds}}[\![\textbf{right } n]\!](x,y) = (x \oplus \mathscr{N}[\![n]\!], y)$$

$$\mathscr{S}_{\text{ds}}[\![\textbf{up}]\!](x,y) = (x, y \oplus \mathbf{1}) \qquad \mathscr{S}_{\text{ds}}[\![\textbf{up } n]\!](x,y) = (x, y \oplus \mathscr{N}[\![n]\!])$$

$$\mathscr{S}_{\text{ds}}[\![\textbf{down}]\!](x,y) = (x, y \ominus \mathbf{1}) \qquad \mathscr{S}_{\text{ds}}[\![\textbf{down } n]\!](x,y) = (x, y \ominus \mathscr{N}[\![n]\!])$$

$$\mathscr{S}_{\text{ds}}[\![\textbf{reset}]\!]p = p^* \qquad \mathscr{S}_{\text{ds}}[\![\textbf{skip}]\!]p = p$$

$$\mathscr{S}_{\text{ds}}[\![S_1; S_2]\!] = \mathscr{S}_{\text{ds}}[\![S_2]\!] \circ \mathscr{S}_{\text{ds}}[\![S_1]\!]$$

$$\mathscr{S}_{\text{ds}}[\![S_1; S_2]\!]p = (\mathscr{S}_{\text{ds}}[\![S_2]\!] \circ \mathscr{S}_{\text{ds}}[\![S_1]\!])p$$

$$= \begin{cases} p', & \text{if there exists a state } p'' \text{ such, that } \mathscr{S}_{\text{ds}}[\![S_1]\!]p = p'' \\ & \text{and } \mathscr{S}_{\text{ds}}[\![S_2]\!]p'' = p', \\ \\ \bot, & \text{if } \mathscr{S}_{\text{ds}}[\![S_1]\!]p = \bot, \text{ or there exists a state } p'' \text{ such, that} \\ & \mathscr{S}_{\text{ds}}[\![S_1]\!]p = p'', \text{ but } \mathscr{S}_{\text{ds}}[\![S_2]\!]p'' = \bot. \end{cases}$$

Structural operational semantics

The structural operational semantics for our domain-specific language is defined as follows:

- we define a transition relation of structural operational semantics:

$$\langle S, p\rangle \Rightarrow \alpha,$$

  where α can be

  - either a new state p':

$$\langle S, p\rangle \Rightarrow p',$$

  - or a configuration $\langle S', p'\rangle$:

$$\langle S, p\rangle \Rightarrow \langle S', p'\rangle,$$

  where $S, S' \in \mathbf{Statm}, p, p' \in \mathbf{Point}$.

- Semantic function $\mathscr{S}_{\mathrm{os}} : \mathbf{Statm} \to (\mathbf{Point} \to \mathbf{Point})$ is defined as follows:

$$\mathscr{S}_{\mathrm{os}}[\![S]\!]p = \begin{cases} p', & \text{if } \langle S, p\rangle \Rightarrow^* p', \\ \perp, & \text{otherwise.} \end{cases}$$

Structural operational semantics

The structural operational semantics of language commands for robot control is defined as follows:

$$\langle \mathbf{left}, (x, y) \rangle \Rightarrow (x \ominus \mathbf{1}, y)$$

$$\langle \mathbf{left}\ 1, p \rangle \Rightarrow \langle \mathbf{left}, p \rangle$$

$$\langle \mathbf{left}\ n, p \rangle \Rightarrow \langle \mathbf{left};\ \mathbf{left}\ m, p \rangle,$$

while we assume that it holds

$$\mathscr{N}[\![n]\!] = \mathscr{N}[\![m]\!] \oplus \mathbf{1}.$$

The semantics of other commands for movement in a given direction is defined analogously.

Structural operational semantics

$$\langle \textbf{reset}, (x, y) \rangle \Rightarrow p^*$$

$$\langle \textbf{skip}, (x, y) \rangle \Rightarrow (x, y)$$

Two situations can occur for a sequence of statements:

1. $\langle S_1, (x, y) \rangle \Rightarrow \langle S_1', (x', y') \rangle$ (statement S_1 is not done in one step):

$$\langle S_1; S_2, (x, y) \rangle \Rightarrow \langle S_1'; S_2, (x', y') \rangle$$

2. $\langle S_1, (x, y) \rangle \Rightarrow (x', y')$ (statement S_1 is performed in one step):

$$\langle S_1; S_2, (x, y) \rangle \Rightarrow \langle S_2, (x', y') \rangle$$

Natural semantics

- we define a transition relation of natural semantics:

$$\langle S, p \rangle \to p',$$

- inference rule of natural semantics:

$$\frac{\langle S_1, s_1 \rangle \to s_1', \cdots, \langle S_n, s_n' \rangle \to s'}{\langle S, s \rangle \to s'}$$

- semantic function is defined as follows:

$$\mathscr{S}_{\mathrm{ns}} : \mathbf{Statm} \to (\mathbf{Point} \to \mathbf{Point}),$$

pričom

$$\mathscr{S}_{\mathrm{ns}}[\![S]\!]\ p = \begin{cases} p', & \text{if } \langle S, p \rangle \to p', \\ \perp, & \text{otherwise.} \end{cases}$$

Natural semantics

$$\frac{}{\langle \mathbf{left}, p\rangle \to p\,(\pi_1(p) \ominus \mathbf{1}, \pi_2(p))}\ (\mathrm{left}_{\mathrm{ns}})$$

$$\frac{}{\langle \mathbf{right}, p\rangle \to p\,(\pi_1(p) \oplus \mathbf{1}, \pi_2(p))}\ (\mathrm{right}_{\mathrm{ns}})$$

$$\frac{}{\langle \mathbf{up}, p\rangle \to p\,(\pi_1(p), \pi_2(p) \oplus \mathbf{1})}\ (\mathrm{up}_{\mathrm{ns}})$$

$$\frac{}{\langle \mathbf{down}, p\rangle \to p\,(\pi_1(p), \pi_2(p) \ominus \mathbf{1})}\ (\mathrm{down}_{\mathrm{ns}})$$

Natural semantics

$$\frac{}{\langle \mathbf{reset}, p\rangle \to p^*} \ (\text{reset}_{\text{ns}})$$

$$\frac{\langle S_1, s\rangle \to s'' \quad \langle S_2, s''\rangle \to s'}{\langle S_1; S_2, s\rangle \to s'} \ (\text{comp}_{\text{ns}})$$

$$\frac{}{\langle \mathbf{skip}, p\rangle \to p} \ (\text{skip}_{\text{ns}})$$

Extension of a language

We extend and modify the syntax of the language:

$$\begin{aligned} S ::= \quad & \mathbf{forward} \mid \mathbf{forward}\ n \mid \mathbf{turn\ left} \mid \mathbf{turn\ right} \\ & \mathbf{reset} \mid \mathbf{skip} \mid S; S \end{aligned}$$

- We will extend the configuration with another element that will represent the current routing. This value is expressed as an angle (in degrees), e.g. angle $\mathbf{0}$ means upward direction (*up*), $\mathbf{90}$ means right direction (*right*), $\mathbf{270}$ means downward direction (*down*), etc.
- In this model, the robot can only turn by an angle of $\mathbf{90}$ degrees.
- The relevant semantic field for angles (direction) is $\mathbf{Angle} = \mathbf{Z}$.
- The state of the robot is expressed by the pair $\langle p, \varphi \rangle$, for the states we introduce the semantic field of configurations

$$\mathbf{Config} = \mathbf{Point} \times \mathbf{Angle}.$$

Structural operational semantics – transition relation

A transition relation can take the following forms:

- for a statemet S not terminated in one step:

$$\langle S, p, \varphi \rangle \Rightarrow \langle S', p', \varphi' \rangle,$$

- for a statement S that is executed in one step:

$$\langle S, p, \varphi \rangle \Rightarrow \langle p', \varphi' \rangle,$$

  where $p, p' \in \mathbf{Point}$, $\varphi, \varphi' \in \mathbf{Angle}$.

- Semantic function $\mathscr{S}'_{\mathrm{os}} : \mathbf{Statm} \to (\mathbf{Config} \to \mathbf{Config})$ is defined as follows:

$$\mathscr{S}'_{\mathrm{os}}[\![S]\!]\langle p, \varphi \rangle = \begin{cases} \langle p', \varphi' \rangle, & \text{if } \langle S, \langle p, \varphi \rangle \rangle \Rightarrow^* \langle p', \varphi \rangle, \\ \perp, & \text{otherwise.} \end{cases}$$

Structural operational semantics – semantics of statements

The structural operational semantics for extended language commands is defined in the following way:

$$\langle \textbf{turn left}, \langle p, \varphi \rangle \rangle \Rightarrow \langle p, (\varphi \oplus \mathbf{270}) \mod \mathbf{360} \rangle$$

$$\langle \textbf{turn right}, \langle p, \varphi \rangle \rangle \Rightarrow \langle p, (\varphi \oplus \mathbf{90}) \mod \mathbf{360} \rangle$$

$$\langle \textbf{forward}, \langle (x, y), \varphi \rangle \rangle \Rightarrow \langle (x \oplus \sin \varphi, y \oplus \cos \varphi), \varphi \rangle$$

$$\langle \textbf{forward } n, \langle (x, y), \varphi \rangle \rangle \Rightarrow \langle \textbf{forward};\ \textbf{forward } m, \langle (x, y), \varphi \rangle \rangle$$

for $\mathscr{N}[\![n]\!] = \mathscr{N}[\![m]\!] \oplus \mathbf{1}$.

Since the angles expressing the direction are multiples of the angle $\mathbf{90}$, the values of the functions $\sin \varphi$ and $\cos \varphi$ are only $-\mathbf{1}$, $\mathbf{0}$, $\mathbf{1}$.

Structural operational semantics – semantics of statements

$$\langle \mathbf{reset}, \langle p, \varphi \rangle\rangle \Rightarrow \langle p^*, \mathbf{0} \rangle$$

$$\langle \mathbf{skip}, \langle p, \varphi \rangle\rangle \Rightarrow \langle p, \varphi \rangle$$

Two situations can occur for a sequence of commands:

1. $\langle S_1, \langle p, \varphi \rangle\rangle \Rightarrow \langle S_1', \langle p', \varphi' \rangle\rangle$ (statement S_1 is not done in one step):
$$\langle S_1; S_2, \langle p, \varphi \rangle\rangle \Rightarrow \langle S_1'; S_2, \langle p', \varphi' \rangle\rangle,$$
2. $\langle S_1, \langle p, \varphi \rangle\rangle \Rightarrow \langle p', \varphi' \rangle$ (statement S_1 is executed in one step):
$$\langle S_1; S_2, \langle p, \varphi \rangle\rangle \Rightarrow \langle S_2, \langle p', \varphi' \rangle\rangle.$$

Natural semantics – transition relation

- A transition session has the form
$$\langle S, \langle p, \varphi \rangle\rangle \rightarrow \langle p', \varphi' \rangle,$$
where $p, p' \in \mathbf{Point}$, $\varphi, \varphi' \in \mathbf{Angle}$.
- Semantic function $\mathscr{S}'_{\mathrm{ns}} : \mathbf{Statm} \rightarrow (\mathbf{Config} \rightarrow \mathbf{Config})$ has a form
$$\mathscr{S}'_{\mathrm{ns}}[\![S]\!]\langle p, \varphi \rangle = \begin{cases} \langle p', \varphi' \rangle, & \text{if } \langle S, \langle p, \varphi \rangle\rangle \rightarrow \langle p', \varphi \rangle, \\ \perp, & \text{otherwise.} \end{cases}$$

Natural semantics – semantics of statements

The natural semantics for extended language commands is defined as follows:

$$\frac{}{\langle \textbf{turn left}, p, \varphi\rangle \to \langle p, (\varphi \oplus \mathbf{270}) \bmod \mathbf{360}\rangle}\ (\text{turn-l}_{\text{ns}})$$

$$\frac{}{\langle \textbf{turn right}, p, \varphi\rangle \to \langle p, (\varphi \oplus \mathbf{90}) \bmod \mathbf{360}\rangle}\ (\text{turn-r}_{\text{ns}})$$

$$\frac{}{\langle \textbf{forward}, p, \varphi\rangle \to \langle p\left(\pi_1(p) \oplus \sin\varphi, \pi_2(p) \oplus \cos\varphi\right), \varphi\rangle}\ (\text{fw}_{\text{ns}})$$

Since the angles expressing the direction are multiples of the angle $\mathbf{90}$, the values of the functions $\sin\varphi$ and $\cos\varphi$ are only $-\mathbf{1}$, $\mathbf{0}$, $\mathbf{1}$.

Natural semantics – semantics of statements

$$\frac{}{\langle \mathbf{reset}, \langle p, \alpha \rangle \rangle \to \langle p^*, \mathbf{0} \rangle} \ (\text{e-reset}_{\text{ns}})$$

$$\frac{}{\langle \mathbf{skip}, \langle p, \alpha \rangle \rangle \to \langle p, \alpha \rangle} \ (\text{e-skip}_{\text{ns}})$$

$$\frac{\langle S_1, \langle p, \alpha \rangle \rangle \to \langle p'', \alpha'' \rangle \quad \langle S_2, \langle p'', \alpha'' \rangle \rangle \to \langle p', \alpha' \rangle}{\langle S_1; S_2, \langle p, \alpha \rangle \rangle \to \langle p', \alpha' \rangle} \ (\text{e-comp}_{\text{ns}})$$

The initial configuration is $\langle p^*, \mathbf{0} \rangle$, the coordinate values of p^* are given in advance.

Operational semantics of an extended language

Relationship between language versions

$$
\begin{array}{ll}
\varphi = \mathbf{0} & \langle \mathbf{forward}, \langle p, \varphi \rangle \rangle \equiv \langle \mathbf{up}, p \rangle \\
\varphi = \mathbf{90} & \langle \mathbf{forward}, \langle p, \varphi \rangle \rangle \equiv \langle \mathbf{right}, p \rangle \\
\varphi = \mathbf{180} & \langle \mathbf{forward}, \langle p, \varphi \rangle \rangle \equiv \langle \mathbf{down}, p \rangle \\
\varphi = \mathbf{270} & \langle \mathbf{forward}, \langle p, \varphi \rangle \rangle \equiv \langle \mathbf{left}, p \rangle
\end{array}
$$