Scribe: Siddharth Alexander, Sharad Goel Lecturer: Radu Rugina

1 Syntax of IMP

We start by describing the syntax of IMP (imperative language). IMP is made up of 5 parts:

1. **Constants**: integers $\mathbb{Z}$ (often denoted by $n \in \mathbb{Z}$) and truth values $\mathbb{T} = \{\mathbf{true}, \mathbf{false}\}$ (often denoted by $t \in \mathbb{T}$).

2. **Variables**: locations **Loc** (often denoted by $X \in \mathbf{Loc}$).

3. **Arithmetic Expressions**: **Aexp** (often denoted by $a \in \mathbf{Aexp}$). **Aexp** is defined inductively as follows:
$$a := n \;\mid\; X \;\mid\; a_0 + a_1 \;\mid\; a_0 - a_1 \;\mid\; a_0 * a_1$$

4. **Boolean Expressions**: **Bexp** (often denoted by $b \in \mathit{Bexp}$). **Bexp** is defined inductively as follows:
$$b := \mathbf{true} \;\mid\; \mathbf{false} \;\mid\; a_0 = a_1 \;\mid\; a_0 < a_1 \;\mid\; \neg b \;\mid\; b_0 \wedge b_1 \;\mid\; b_0 \vee b_1$$

5. **Commands**: **Com** (often denoted by $c \in \mathbf{Com}$. Defined inductively as follows:
$$c := \mathbf{skip} \;\mid\; c_0; c_1 \;\mid\; X := a \;\mid\; \mathbf{if}\, b\, \mathbf{then}\, c_0\, \mathbf{else}\, c_1 \;\mid\; \mathbf{while}\, b\, \mathbf{do}\, c$$

Note: This is a sequential, deterministic, barely Turing-complete language.

Example: (of a program in IMP that computes GCD(X,Y) if interpreted appropriately)
$\mathbf{while}\, \neg X = Y\, \mathbf{do}\, \mathbf{if}\, (X < Y)\, \mathbf{then}\, T := X; X := Y; Y := T\, \mathbf{else}\, \mathbf{skip}; X := X - Y$

The statement $X := X - Y$ could be part of the **else** branch of the **if** statement,or follow the **if** statement as part of the body of the **while** statement, or follow the **while** statement. This shows that IMP is ambiguous with respect to the grouping of statements. However, we will not deal with this issue and instead put in parentheses when disambiguation is necessary.

2 Large Step Operational Semantics for IMP

Operational Semantics can be categorized into large step semantics (multiple state transitions in a single step) and small step semantics (individual state transitions). Note that large step semantics trades expressiveness for ease of use.

States Σ are defined as the set of functions $\mathbf{Loc} \to \mathbb{Z}$.

- The operational semantics for commands, $\to_c$, is a subset of $(\mathbf{Com}\times\Sigma) \times \Sigma$. We will use the notation $\langle \mathrm{c}, \sigma\rangle \to_c \sigma'$ for $(c, \sigma, \sigma') \in \to_c$. Defining $\to_c$ as a subset of $(\mathbf{Com}\times\Sigma) \times \Sigma$ (as opposed to a subset of the functions $\mathbf{Com}\times\Sigma \to \Sigma$), we may express non-determinism.

- The operational semantics for arithmetic expressions, $\to_a$, is a subset of $(\mathbf{Aexp}\times\Sigma) \times \mathbb{Z}$. We will use the notation $\langle a, \sigma\rangle \to_a n$ for $(a, \sigma, n) \in \to_a$.

- The operational semantics for boolean expressions, $\to_b$, is a subset of $(\mathbf{Bexp}\times\Sigma) \times T$. We will use the notation $\langle b, \sigma\rangle \to_b t$ for $(b, \sigma, t) \in \to_b$.

Note that the operational semantics for arithmetic and boolean expressions cannot change the state - they do not have side-effects. Henceforth we will drop the subscripts from the operational semantics notation, and use $\to$ instead of $\to_a$, $\to_b$ and $\to_c$. Which relation is intended will be clear from context.

2.1 Semantics for Expressions in IMP

1. **Arithmetic Expressions**:

$$\frac{}{\langle n, \sigma \rangle \to n}$$

$$\frac{}{\langle X, \sigma \rangle \to \sigma(X)}$$

$$\frac{\langle a_0, \sigma \rangle \to n_0 \quad \langle a_1, \sigma \rangle \to n_1}{\langle a_0 \ op \ a_1, \sigma \rangle \to n} \quad \text{where } n = n_0 \ op \ n_1 \text{ and } op \in \{+, -, *\}.$$

2. **Boolean Expressions**:

$$\frac{}{\langle \mathbf{true}, \sigma \rangle \to \mathbf{true}} \qquad \frac{}{\langle \mathbf{false}, \sigma \rangle \to \mathbf{false}}$$

$$\frac{\langle a_0, \sigma \rangle \to n_0 \quad \langle a_1, \sigma \rangle \to n_1}{a_0 \ op \ a_1, \sigma \rangle \to t} \quad \text{where } t = n_0 \ op \ n_1 \text{ and } op \in \{=, <\}.$$

$$\frac{b, \sigma \rangle \to t'}{\langle \neg b, \sigma \rangle \to t} \quad \text{where } t = \neg t'.$$

$$\frac{\langle b_0, \sigma \rangle \to t_0 \quad \langle b_1, \sigma \rangle \to t_1}{\langle b_0 \ op \ b_1, \sigma \rangle \to t} \quad \text{where } t = t_0 \ op \ t_1 \text{ and } op \in \{\wedge, \vee\}.$$