

# IMP

Grigore Roşu (grosu@illinois.edu)

University of Illinois at Urbana-Champaign

## Abstract

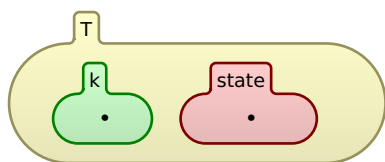
This is the  $\mathbb{K}$  semantic definition of the classic IMP language. IMP is considered a folklore language, without an official inventor, and has been used in many textbooks and papers, often with slight syntactic variations and often without being called IMP. It includes the most basic imperative language constructs, namely basic constructs for arithmetic and Boolean expressions, and variable assignment, conditional, while loop and sequential composition constructs for statements.

```
MODULE IMP-SYNTAX
IMPORTS #ID+#INT
SYNTAX AExp ::= #Id
           | #Int
           | AExp + AExp [strict]
           | AExp / AExp [strict]
SYNTAX BExp ::= #Bool
           | AExp <= AExp [strict]
           | not BExp [strict]
           | BExp and BExp [strict(1)]
SYNTAX Stmt ::= skip;
           | #Id = AExp ; [strict(2)]
           | Stmt Stmt
           | if BExp then Stmt else Stmt [strict(1)]
           | while BExp do Stmt
END MODULE
```

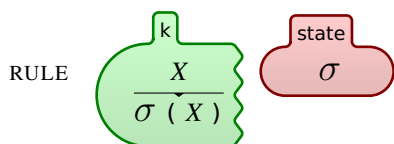
```
MODULE IMP-SEMANTICS
```

```
IMPORTS IMP-SYNTAX
```

```
CONFIGURATION:
```



```
SYNTAX KResult ::= #Int
                | #Bool
```



```
RULE  $I_1 + I_2 \Rightarrow I_1 +_{Int} I_2$ 
```

```
RULE  $I_1 / I_2 \Rightarrow I_1 \div_{Int} I_2$  when  $I_2 \neq_{Bool} 0$ 
```

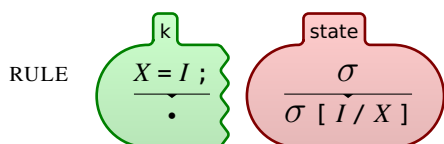
```
RULE  $I_1 <= I_2 \Rightarrow I_1 \leq_{Int} I_2$ 
```

```
RULE  $\text{not } T \Rightarrow \neg_{Bool} T$ 
```

```
RULE  $\text{true and } B \Rightarrow B$ 
```

```
RULE  $\text{false and } - \Rightarrow \text{false}$ 
```

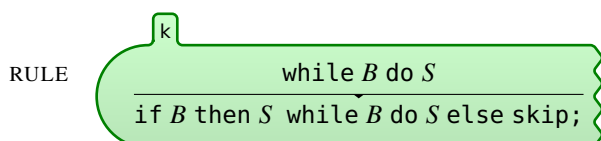
```
RULE  $\text{skip;} \Rightarrow \bullet$ 
```



```
RULE  $S_1 S_2 \Rightarrow S_1 \circ S_2$ 
```

```
RULE  $\text{if true then } S \text{ else } - \Rightarrow S$ 
```

```
RULE  $\text{if false then } - \text{ else } S \Rightarrow S$ 
```



[structural]

```
END MODULE
```