

```

class Main {
  public int main(int n) {
    int r;
    if (n == 0) { r = 1; }
    else { r = n * Main.main(n - 1); }
    return r;
  }
}

```

Above is the MiniJava program **factorial**. Below is a proof of calling it with $n = 3$ to obtain the value 6. Code from the original program is marked in **dark blue fixed-width font**. Variables, while not necessary to use when actually executing a program in SOS, are in **dark red font**. Please take note of the variables, as they help you show how the different parts of the proof interact.

Even if you do not understand this whole example, it is a demonstration of the ability to create a SOS framework to execute a program. Your interpreter and compiler will be performing the same steps that are outlined below.

$$\begin{aligned}
\Pi &\in \textit{Program} &= \text{Program}(\text{Class}(\text{Main}, -, [\text{m}])) \\
m &\in \textit{Method} &= \text{Method}(-, \text{main}, \{\text{Var}(-, \text{n})\}, \{\text{Var}(-, \text{r})\}, \text{if } \dots, \text{r}) \\
\sigma, \sigma' &\in \textit{State} \\
s, s_1, s_2 &\in \textit{Statement} \\
x_1, y_1 &\in \textit{Id} \\
v_1 &\in \textit{Value}
\end{aligned}$$

Rule	Instance	Notes
by METHOD-CALL :	<code>Main.main(3)</code> , $\emptyset, \Pi \Downarrow \text{IntV}(6)$	$m = \text{Method}(-, \text{main}, \{\text{Var}(-, \text{x}_1 = \text{n})\}, \{\text{Var}(-, \text{y}_1 = \text{r})\}, \text{if } \dots, \text{r})$ $m \in \text{Main} \in \Pi, \sigma_{f(3)} = \{x_1 \mapsto v_1, y_1 \mapsto \text{NullV}\} = \{n \mapsto \text{IntV}(3), r \mapsto \text{NullV}\}$
by INT :	<code>3</code> , $\emptyset, \Pi \Downarrow (v_1 = \text{IntV}(3))$	
by IF-FALSE :	(see below)	

continued	<code>if (n == 0) _ else</code> $s_2, \sigma_{f(3)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(3), r \mapsto \text{IntV}(6)\}$	$s_2 = \{ \text{r} = \dots; \}$
by EQUALS-INT :	<code>n == 0</code> , $\{n \mapsto \text{IntV}(3), \dots\}, - \Downarrow \text{BoolV}(\text{false})$	
by VAR :	<code>n</code> , $\{n \mapsto \text{IntV}(3), \dots\}, - \Downarrow \text{IntV}(3)$	$\{n \mapsto \text{IntV}(3), \dots\}(n) = \text{IntV}(3)$
by INT :	<code>0</code> , $-, - \Downarrow \text{IntV}(0)$	
by BLOCK :	<code>{ r = ...; }</code> , $\sigma_{f(3)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(3), r \mapsto \text{IntV}(6)\}$	$\sigma_{f(3)} = \{n \mapsto \text{IntV}(3), r \mapsto \text{NullV}\}$
by ASSIGN :	<code>r = n * Main.main(n - 1);</code> , $\sigma_{f(3)}, \Pi \Rightarrow (\sigma'_{f(3)} = \{n \mapsto \text{IntV}(3), r \mapsto \text{IntV}(6)\})$	$\sigma_{f(3)} = \{n \mapsto \text{IntV}(3), r \mapsto \text{NullV}\}, \sigma'_{f(3)} = \sigma_{f(3)}[\text{IntV}(6)/r]$
by MULTIPLICATION :	<code>n * Main.main(n - 1)</code> , $\{n \mapsto \text{IntV}(3), \dots\}, \Pi \Downarrow \text{IntV}(6)$	$6 = 3 \times 2$
by VAR :	<code>n</code> , $\{n \mapsto \text{IntV}(3), \dots\}, - \Downarrow \text{IntV}(3)$	
by METHOD-CALL :	<code>Main.main(n - 1)</code> , $\{n \mapsto \text{IntV}(3), \dots\}, \Pi \Downarrow \text{IntV}(2)$	$m = \text{Method}(-, \text{main}, \{\text{Var}(-, \text{x}_1 = \text{n})\}, \{\text{Var}(-, \text{y}_1 = \text{r})\}, \text{if } \dots, \text{r})$ $m \in \text{Main} \in \Pi, \sigma_{f(2)} = \{x_1 \mapsto v_1, y_1 \mapsto \text{NullV}\} = \{n \mapsto \text{IntV}(2), r \mapsto \text{NullV}\}$
by MINUS :	<code>n - 1</code> , $\{n \mapsto \text{IntV}(3), \dots\}, - \Downarrow (v_1 = \text{IntV}(2))$	$2 = 3 - 1$
by VAR :	<code>n</code> , $\{n \mapsto \text{IntV}(3), \dots\}, - \Downarrow \text{IntV}(3)$	$\{n \mapsto \text{IntV}(3), \dots\}(n) = \text{IntV}(3)$
by INT :	<code>1</code> , $-, - \Downarrow \text{IntV}(1)$	
by IF-FALSE :	(see next page)	

Rule	Instance	Notes
<i>continued</i>	if (n == 0) _ else $s_2, \sigma_{f(2)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(2), r \mapsto \text{IntV}(2)\}$	$s_2 = \{ \text{r} = \dots; \}$
by EQUALS-INT :	$n == 0, \{n \mapsto \text{IntV}(2), \dots\}, _ \Downarrow \text{BoolV}(\text{false})$	
by VAR :	$n, \{n \mapsto \text{IntV}(2), \dots\}, _ \Downarrow \text{IntV}(2)$	$\{n \mapsto \text{IntV}(2), \dots\}(n) = \text{IntV}(2)$
by INT :	$0, _, _ \Downarrow \text{IntV}(0)$	
by BLOCK :	$\{ r = \dots; \}, \sigma_{f(2)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(2), r \mapsto \text{IntV}(2)\}$	$\sigma_{f(2)} = \{n \mapsto \text{IntV}(2), r \mapsto \text{NullV}\}$
by ASSIGN :	$r = n * \text{Main.main}(n - 1);, \sigma_{f(2)}, \Pi \Rightarrow (\sigma'_{f(2)} = \{n \mapsto \text{IntV}(2), r \mapsto \text{IntV}(2)\})$	$\sigma_{f(2)} = \{n \mapsto \text{IntV}(2), r \mapsto \text{NullV}\}, \sigma'_{f(2)} = \sigma_{f(2)}[\text{IntV}(2)/r]$
by MULTIPLICATION :	$n * \text{Main.main}(n - 1), \{n \mapsto \text{IntV}(2), \dots\}, \Pi \Downarrow \text{IntV}(2)$	$2 = 2 \times 1$
by VAR :	$n, \{n \mapsto \text{IntV}(2), \dots\}, _ \Downarrow \text{IntV}(2)$	
by METHOD-CALL :	$\text{Main.main}(n - 1), \{n \mapsto \text{IntV}(2), \dots\}, \Pi \Downarrow \text{IntV}(1)$	$m = \text{Method}(_, \text{main}, \{\text{Var}(_, x_1 = n)\}, \{\text{Var}(_, y_1 = r)\}, \text{if } \dots, r),$ $m \in \text{Main} \in \Pi, \sigma_{f(1)} = \{x_1 \mapsto v_1, y_1 \mapsto \text{NullV}\} = \{n \mapsto \text{IntV}(1), r \mapsto \text{NullV}\}$
by MINUS :	$n - 1, \{n \mapsto \text{IntV}(2), \dots\}, _ \Downarrow (v_1 = \text{IntV}(1))$	$1 = 2 - 1$
by VAR :	$n, \{n \mapsto \text{IntV}(2), \dots\}, _ \Downarrow \text{IntV}(2)$	$\{n \mapsto \text{IntV}(2), \dots\}(n) = \text{IntV}(2)$
by INT :	$1, _, _ \Downarrow \text{IntV}(1)$	
by IF-FALSE :	(see below)	
<i>continued</i>	if (n == 0) _ else $s_2, \sigma_{f(1)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(1), r \mapsto \text{IntV}(1)\}$	$s_2 = \{ \text{r} = \dots; \}$
by EQUALS-INT :	$n == 0, \{n \mapsto \text{IntV}(1), \dots\}, _ \Downarrow \text{BoolV}(\text{false})$	
by VAR :	$n, \{n \mapsto \text{IntV}(1), \dots\}, _ \Downarrow \text{IntV}(1)$	$\{n \mapsto \text{IntV}(1), \dots\}(n) = \text{IntV}(1)$
by INT :	$0, _, _ \Downarrow \text{IntV}(0)$	
by BLOCK :	$\{ r = \dots; \}, \sigma_{f(1)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(1), r \mapsto \text{IntV}(1)\}$	$\sigma_{f(1)} = \{n \mapsto \text{IntV}(1), r \mapsto \text{NullV}\}$
by ASSIGN :	$r = n * \text{Main.main}(n - 1);, \sigma_{f(1)}, \Pi \Rightarrow (\sigma'_{f(1)} = \{n \mapsto \text{IntV}(1), r \mapsto \text{IntV}(1)\})$	$\sigma_{f(1)} = \{n \mapsto \text{IntV}(1), r \mapsto \text{NullV}\}, \sigma'_{f(1)} = \sigma_{f(1)}[\text{IntV}(1)/r]$
by MULTIPLICATION :	$n * \text{Main.main}(n - 1), \{n \mapsto \text{IntV}(1), \dots\}, \Pi \Downarrow \text{IntV}(1)$	$1 = 1 \times 1$
by VAR :	$n, \{n \mapsto \text{IntV}(1), \dots\}, _ \Downarrow \text{IntV}(1)$	
by METHOD-CALL :	$\text{Main.main}(n - 1), \{n \mapsto \text{IntV}(1), \dots\}, \Pi \Downarrow \text{IntV}(1)$	$m = \text{Method}(_, \text{main}, \{\text{Var}(_, x_1 = n)\}, \{\text{Var}(_, y_1 = r)\}, \text{if } \dots, r),$ $m \in \text{Main} \in \Pi, \sigma_{f(0)} = \{x_1 \mapsto v_1, y_1 \mapsto \text{NullV}\} = \{n \mapsto \text{IntV}(0), r \mapsto \text{NullV}\}$
by MINUS :	$n - 1, \{n \mapsto \text{IntV}(1), \dots\}, _ \Downarrow (v_1 = \text{IntV}(0))$	$0 = 1 - 1$
by VAR :	$n, \{n \mapsto \text{IntV}(1), \dots\}, _ \Downarrow \text{IntV}(1)$	$\{n \mapsto \text{IntV}(1), \dots\}(n) = \text{IntV}(1)$
by INT :	$1, _, _ \Downarrow \text{IntV}(1)$	
by IF-TRUE :	(see below)	
<i>continued</i>	if (n == 0) s_1 else $_, \sigma_{f(0)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(0), r \mapsto \text{IntV}(1)\}$	$s_1 = \{ \text{r} = 1; \}$
by EQUALS-INT :	$n == 0, \{n \mapsto \text{IntV}(1), \dots\}, _ \Downarrow \text{BoolV}(\text{true})$	
by VAR :	$n, \{n \mapsto \text{IntV}(0), \dots\}, _ \Downarrow \text{IntV}(0)$	$\{n \mapsto \text{IntV}(0), \dots\}(n) = \text{IntV}(0)$
by INT :	$0, _, _ \Downarrow \text{IntV}(0)$	
by BLOCK :	$\{ r = 1; \}, \sigma_{f(0)}, \Pi \Rightarrow \{n \mapsto \text{IntV}(0), r \mapsto \text{IntV}(1)\}$	$\sigma_{f(0)} = \{n \mapsto \text{IntV}(0), r \mapsto \text{NullV}\}$
by ASSIGN :	$r = 1;, \sigma_{f(0)}, \Pi \Rightarrow (\sigma'_{f(0)} = \{n \mapsto \text{IntV}(0), r \mapsto \text{IntV}(1)\})$	$\sigma_{f(0)} = \{n \mapsto \text{IntV}(0), r \mapsto \text{NullV}\}, \sigma'_{f(0)} = \sigma_{f(0)}[\text{IntV}(1)/r]$
by INT :	$1, _, _ \Downarrow \text{IntV}(1)$	