

Discovered *i-rules*

February 2020

1 *i-rule*

This document presents *i-rules* to avoid useless mutants. To understand the concepts behind an *i-rule*, please refer to the article of Fernandes et al.([1]).

Next we present the meta-variables needed to understand the elements of a rule, then we show the rules that avoid equivalent mutants (e-rule), followed by those that avoid duplicate mutants (d-rule).

1.1 Meta-variables

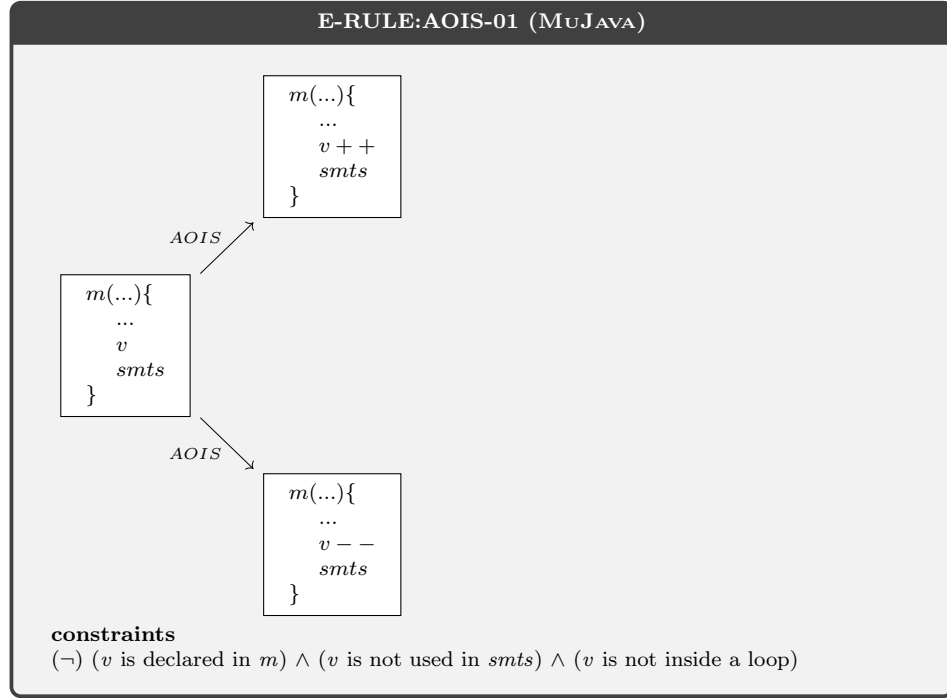
All meta-variables referred by the *i-rules* are depict at Table 1.

Table 1: Meta-variables referred by the *i-rules*.

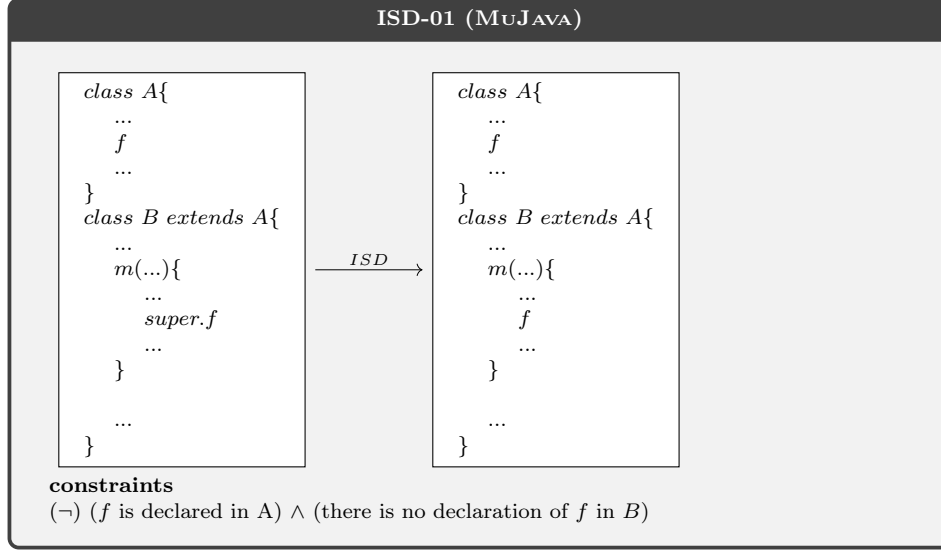
Meta-variables	Description
<i>op</i>	any binary or unary operator
<i>exp</i>	any expression
<i>smts</i>	any block of statements
<i>s</i>	any statement
<i>;</i>	an empty statement
<i>A, B</i>	class references
<i>m</i>	method references
<i>v</i>	any identifier of a local variable for a primitive integral type (byte, short, int, long)
<i>sv</i>	any identifier of a String or Array variable
<i>f</i>	any identifier of a field
<i>dv</i>	the default value of any specified type (i.e., int=0, boolean=false, char='u0000', double=0.0d, float=0.0f, long=0L, Object=null, etc.)

1.2 E-RULEs

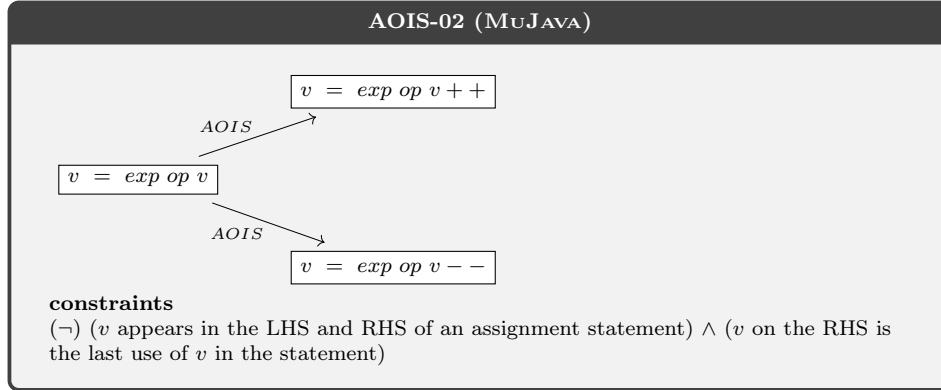
1.2.1 AOIS-01 (MuJAVa)



1.2.2 ISD-01 (MuJAVa)



1.2.3 AOIS-02 (MuJAVa)



1.2.4 ROR-01 (MAJOR)

ROR-01 (MAJOR)
<div style="border: 1px solid black; display: inline-block; padding: 5px; margin-bottom: 10px;"> $if(sv.length\ rop_1\ 0)\{\dots\}$ </div> \xrightarrow{ROR} <div style="border: 1px solid black; display: inline-block; padding: 5px; margin-bottom: 10px;"> $if(sv.length\ rop_2\ 0)\{\dots\}$ </div> <p>constraints</p> <p> $(\neg) (rop_1 \in \{<\} \wedge rop_2 \in \{!=\}) \vee$ $(rop_1 \in \{!=\} \wedge rop_2 \in \{<\}) \vee$ $(rop_1 \in \{>\} \wedge rop_2 \in \{!=\}) \vee$ $(rop_1 \in \{!=\} \wedge rop_2 \in \{>\}) \vee$ $(rop_1 \in \{==\} \wedge rop_2 \in \{<=\})$ </p>

- Obs 1. This E-RULE also works for Array type variables.
 Obs 2. This E-RULE also applies to MUJAVA (see [1.2.14](#)).

1.2.5 AOR-01 (MAJOR)

AOR-01 (MAJOR)
<div style="border: 1px solid black; display: inline-block; padding: 5px; margin-bottom: 10px;"> $new\ StringBuilder(exp_1\ aop_1\ exp_2)$ </div> \xrightarrow{AOR} <div style="border: 1px solid black; display: inline-block; padding: 5px; margin-bottom: 10px;"> $new\ StringBuilder(exp_1\ aop_2\ exp_2)$ </div> <p>constraints</p> <p>$(\neg) (op_2\ \text{does not turn the resulted value to negative})$</p>

- Obs 1. This E-RULE also applies to MUJAVA (see [1.2.27](#)).

1.2.6 JID-01 (MUJAVA)

JID-01 (MUJAVA)
<div style="border: 1px solid black; display: inline-block; padding: 5px; margin-bottom: 10px;"> $class\ A\{\$ $\quad f = dv;$ $\}$ </div> \xrightarrow{JID} <div style="border: 1px solid black; display: inline-block; padding: 5px; margin-bottom: 10px;"> $class\ A\{\$ $\quad f;$ $\}$ </div> <p>constraints</p> <p>$(\neg) \emptyset$</p>

1.2.7 PNC-01 (MuJAVa)

PNC-01 (MuJAVa)	
<div style="border: 1px solid black; display: inline-block; padding: 5px 10px;"><i>new A().f</i></div> \xrightarrow{PNC} <div style="border: 1px solid black; display: inline-block; padding: 5px 10px;"><i>new B().f</i></div>	<p>constraints $(\neg)(A \text{ extends } B) \wedge$ $(f \text{ exists only in } A) \wedge$ $(B \text{ constructor does not change } v) \wedge$ $(B \text{ constructor calls the same } C \text{ constructor})$</p>

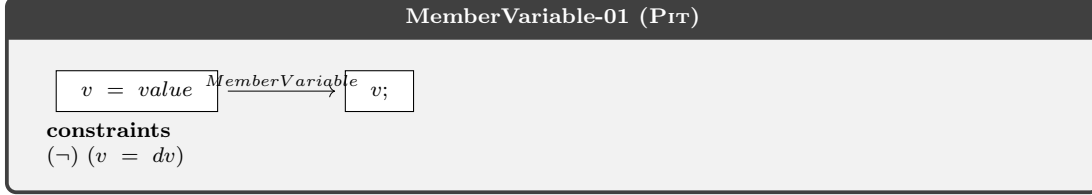
1.2.8 JSI-01 (MuJAVa)

JSI-01 (MuJAVa)	
<div style="border: 1px solid black; display: inline-block; padding: 5px 10px;"><i>type f = value;</i></div> \xrightarrow{JSI} <div style="border: 1px solid black; display: inline-block; padding: 5px 10px;"><i>static type f = value;</i></div>	<p>constraints $(\neg) (f \text{ is read-only})$</p>

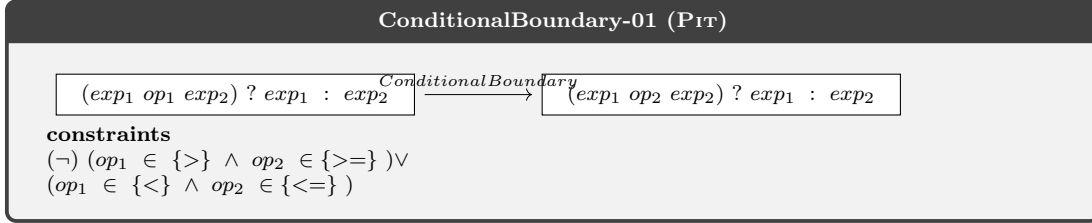
1.2.9 ROR-02 (MAJOR)

ROR-02 (MAJOR)	
<div style="border: 1px solid black; display: inline-block; padding: 10px;"> <i>if (v₁ op₁ v₂) {</i> <i style="padding-left: 20px;">return v₁;</i> <i>else {</i> <i style="padding-left: 20px;">return v₂;</i> <i>}</i> </div> \xrightarrow{ROR} <div style="border: 1px solid black; display: inline-block; padding: 10px;"> <i>if (v₁ op₂ v₂) {</i> <i style="padding-left: 20px;">return v₁;</i> <i>else {</i> <i style="padding-left: 20px;">return v₂;</i> <i>}</i> </div>	<p>constraints $(\neg) (op_1 \in \{>\} \wedge op_2 \in \{>=\}) \vee (op_1 \in \{<\} \wedge op_2 \in \{<=\})$</p>

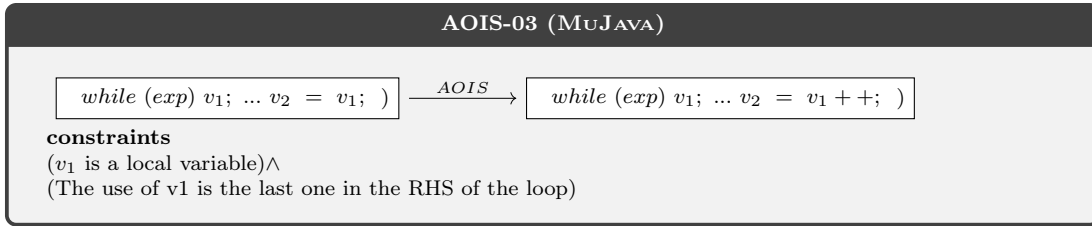
1.2.10 MemberVariable-01 (PIT)



1.2.11 ConditionalBoundary-01 (PIT)



1.2.12 AOIS-03 (MUJAVA)



1.2.13 AOIU-01 (MuJAVa)

AOIU-01 (MuJAVa)	
$\boxed{v1 \% = v2}$	$\xrightarrow{AOIU} \boxed{v1 \% = -v2}$
constraints $(\neg) \emptyset$	

1.2.14 ROR-01 (MuJAVa)

ROR-01 (MuJAVa)	
$\boxed{if(sv.length\ op_1\ 0)\{\dots\}}$	$\xrightarrow{ROR} \boxed{if(sv.length\ op_2\ 0)\{\dots\}}$
constraints $(\neg) (op_1 \in \{<\} \wedge op_2 \in \{!=\}) \vee$ $(op_1 \in \{!=\} \wedge op_2 \in \{<\}) \vee$ $(op_1 \in \{>\} \wedge op_2 \in \{!=\}) \vee$ $(op_1 \in \{!=\} \wedge op_2 \in \{>\}) \vee$ $(op_1 \in \{==\} \wedge op_2 \in \{<=\})$	

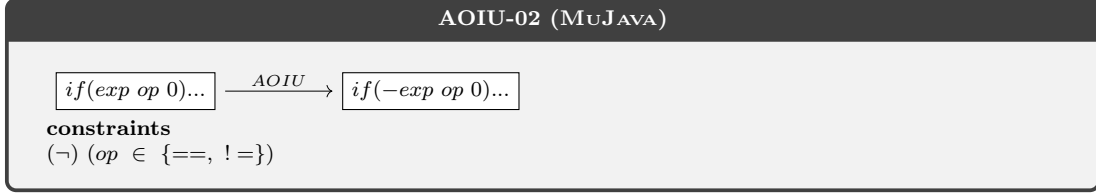
Obs 1. This E-RULE also works for Array type variables.

Obs 2. This E-RULE also applies to MuJAVa (see [1.2.4](#)).

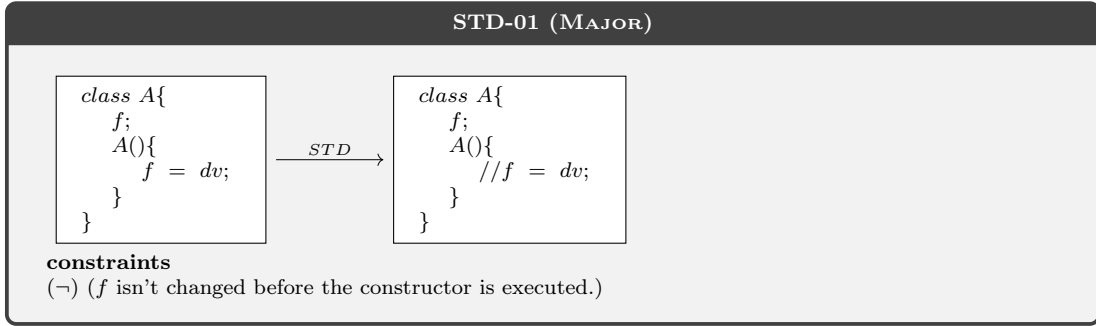
1.2.15 SOR-01 (MAJOR)

SOR-01 (MAJOR)	
$\boxed{v >> exp}$	$\xrightarrow{SOR} \boxed{v >>> exp}$
constraints $(\neg) (v > 0)$	

1.2.16 AOIU-02 (MuJAVa)

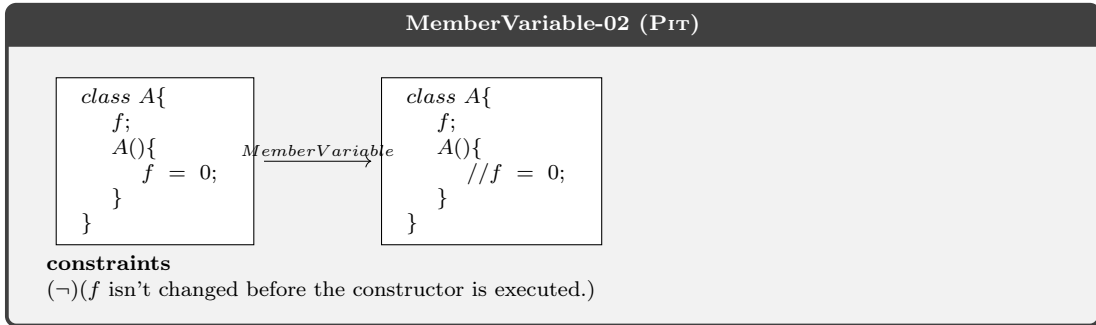


1.2.17 STD-01 (MAJOR)



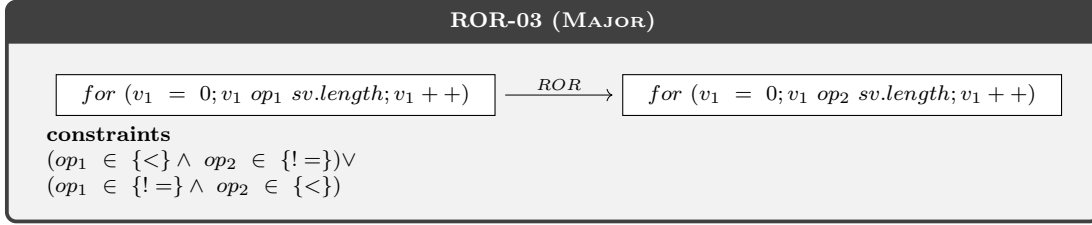
Obs 1. This E-RULE also applies to PIT (see 1.2.18).

1.2.18 MemberVariable-02 (PIT)



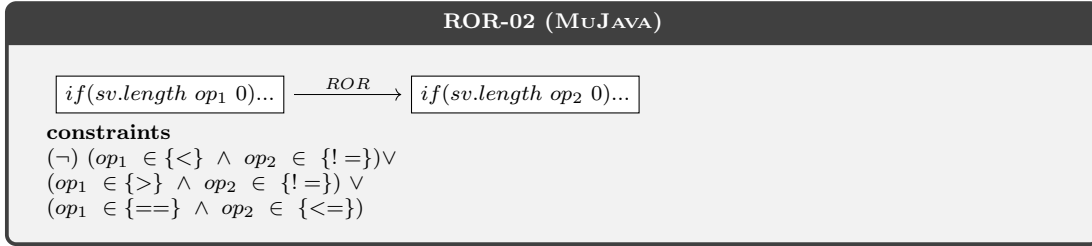
Obs 1. This E-RULE also applies to MAJOR (see 1.2.17).

1.2.19 ROR-03 (MAJOR)



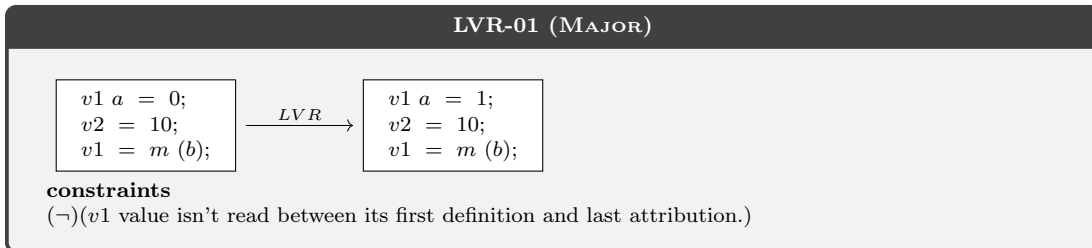
Obs 1. This E-RULE also applies to MuJAVA (see 1.2.20).

1.2.20 ROR-02 (MuJAVA)



Obs 1. This E-RULE also applies to MAJOR (see 1.2.19).

1.2.21 LVR-01 (MAJOR)



Obs 1. This E-RULE also applies to PIT (see 1.2.22).

1.2.22 InlineConstant-01 (PIT)

InlineConstant-01 (PIT)

$$\begin{array}{l} v_1 \ a = 0; \\ v_2 = 10; \\ v_1 = m(b); \end{array}$$

$\xrightarrow{\text{InlineConstant}}$

$$\begin{array}{l} v_1 \ a = 1; \\ v_2 = 10; \\ v_1 = m(b); \end{array}$$

constraints
 $(\neg)(v_1 \text{ value isn't read between its first definition and last attribution.})$

Obs 1. This E-RULE also applies to MAJOR (see 1.2.21).

1.2.23 AOIS-04 (MUJAV)

AOIS-04 (MUJAV)

$$v_1 := v_2; \dots v_2 := exp$$

$\xrightarrow{\text{AOIS}}$

$$v_1 := v_2 - -; \dots v_2 := exp$$

constraints
 $(\neg)(v_2 \text{ isn't read between the first and last attribution}) \wedge$
 $(op \in \{++, --\})$

1.2.24 ROR-03 (MUJAV)

ROR-03 (MUJAV)

$$\text{if } (v_1 \ op_1 \ v_2) \ v_1 := v_2$$

$\xrightarrow{\text{ROR}}$

$$\text{if } (v_1 \ op_2 \ v_2) \ v_1 := v_2$$

constraints
 $(\neg)(op_1 \in \{<\} \wedge op_2 \in \{<=\}) \vee$
 $(op_1 \in \{>\} \wedge op_2 \in \{>=\})$

Obs 1. This E-RULE also applies to MAJOR and PIT (see 1.2.25 and ??).

1.2.25 ROR-04 (MAJOR)

ROR-04 (MAJOR)	
$\boxed{\text{if } (v_1 \text{ op}_1 v_2) \ v_1 := v_2}$	$\xrightarrow{ROR} \boxed{\text{if } (v_1 \text{ op}_2 v_2) \ v_1 := v_2}$
constraints $(\neg)(\text{op}_1 \in \{<\} \wedge \text{op}_2 \in \{<=\}) \vee$ $(\text{op}_1 \in \{>\} \wedge \text{op}_2 \in \{>=\})$	

Obs 1. This E-RULE also applies to MUJAV and PIT (see 1.2.24 and 1.2.25).

1.2.26 ConditionalBoundary-02 (PIT)

ConditionalBoundary-02 (PIT)	
$\boxed{\text{if } (v_1 \text{ op}_1 v_2) \ v_1 := v_2}$	$\xrightarrow{\text{ConditionalBoundary}} \boxed{\text{if } (v_1 \text{ op}_2 v_2) \ v_1 := v_2}$
constraints $(\neg)(\text{op}_1 \in \{<\} \wedge \text{op}_2 \in \{<=\}) \vee$ $(\text{op}_1 \in \{>\} \wedge \text{op}_2 \in \{>=\})$	

1.2.27 AORB-01 (MUJAV)

AORB-01 (MUJAV)	
$\boxed{\text{new StringBuilder}(\text{exp}_1 \text{ aop}_1 \text{ exp}_2)}$	$\xrightarrow{AORB} \boxed{\text{new StringBuilder}(\text{exp}_1 \text{ aop}_2 \text{ exp}_2)}$
constraints $(\neg) (\text{op}_2 \text{ does not turn the resulted value to negative})$	

Obs 1. This E-RULE also applies to MAJOR (see 1.2.5).

1.2.28 JSD-01 (MuJAVa)

JSD-01 (MuJAVa)	
$\boxed{\text{private static final type} := \text{value};}$	$\xrightarrow{JSD} \boxed{\text{private final type} := \text{value};}$
constraints $(\neg) \emptyset$	

1.2.29 ROR-05 (MuJAVa)

ROR-05 (MuJAVa)	
$\boxed{\text{if}(v \text{ op}_1 \text{ value}) \{ \dots \}}$	$\xrightarrow{ROR} \boxed{\text{if}(v \text{ op}_2 \text{ value}) \{ \dots \}}$
constraints $(\neg) (op_1 \in \{<\} \wedge op_2 \in \{<=\}) \vee (op_1 \in \{>\} \wedge op_2 \in \{>=\})$	

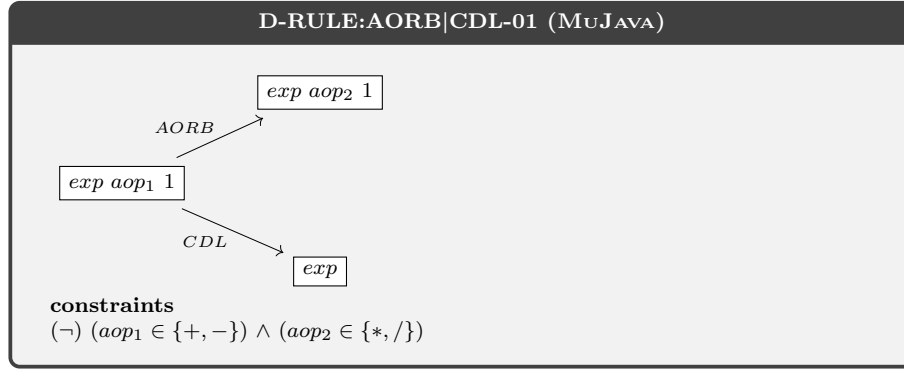
1.2.30 LVR-01 (MAJOR)

LVR-01 (MAJOR)	
$\boxed{v := \text{new CollectionType}(\text{value}_1);}$	$\xrightarrow{LVR} \boxed{v := \text{new CollectionType}(\text{value}_2);}$
constraints $(\neg) \text{ (There is no use of } v \text{ between the two definitions)}$	

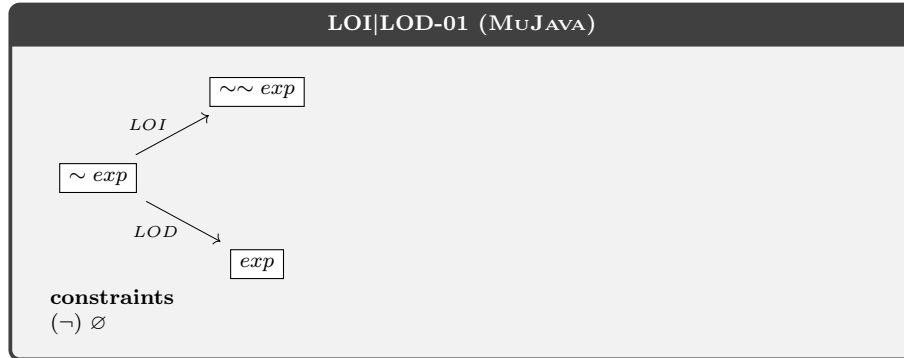
1.3 D-RULEs

...

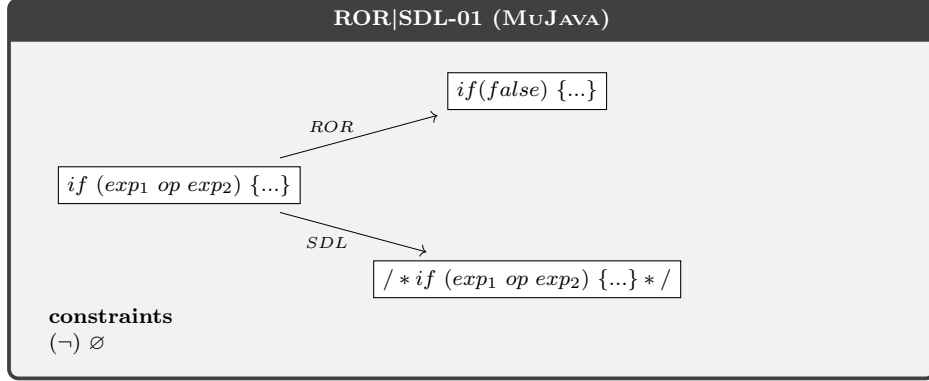
1.3.1 AORB|CDL-01 (MuJ_{AVA})



1.3.2 LOI|LOD-01 (MuJ_{AVA})

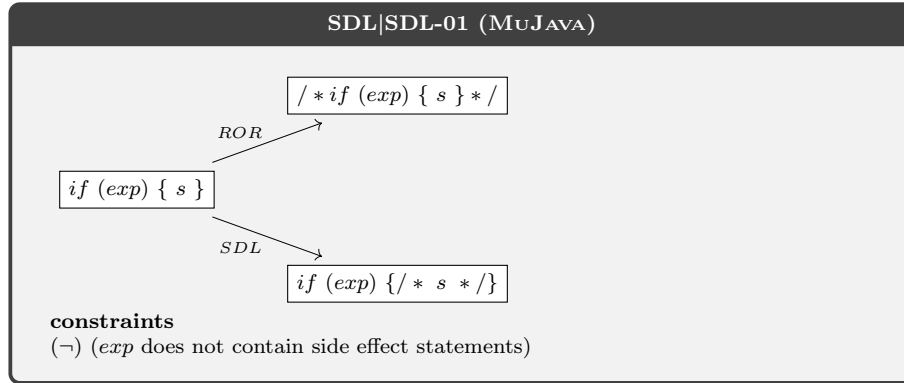


1.3.3 ROR|SDL-01 (MuJAVa)

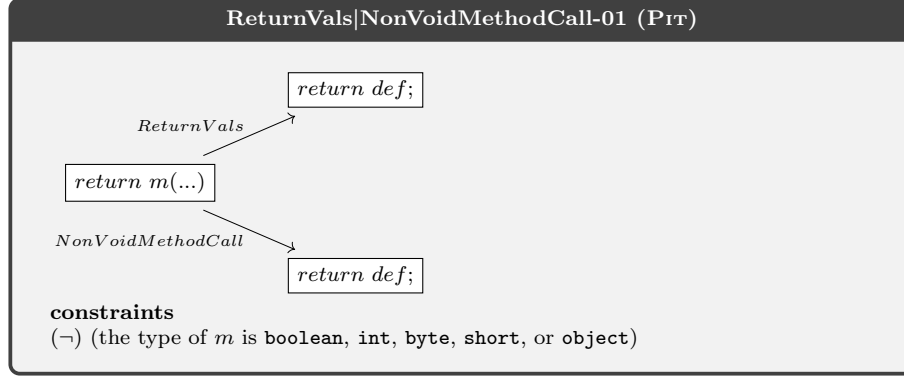


Obs 1. This D-RULE also applies to MAJOR (see ??).

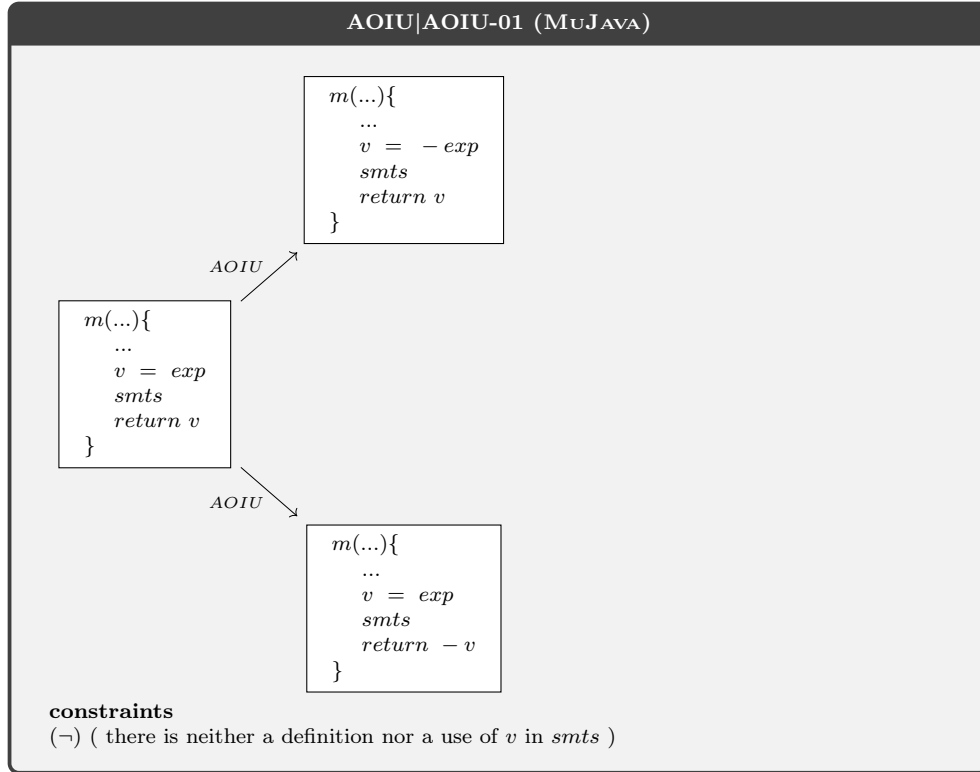
1.3.4 SDL|SDL-01 (MuJAVa)



1.3.5 ReturnVals|NonVoidMethodCall-01 (PIT)

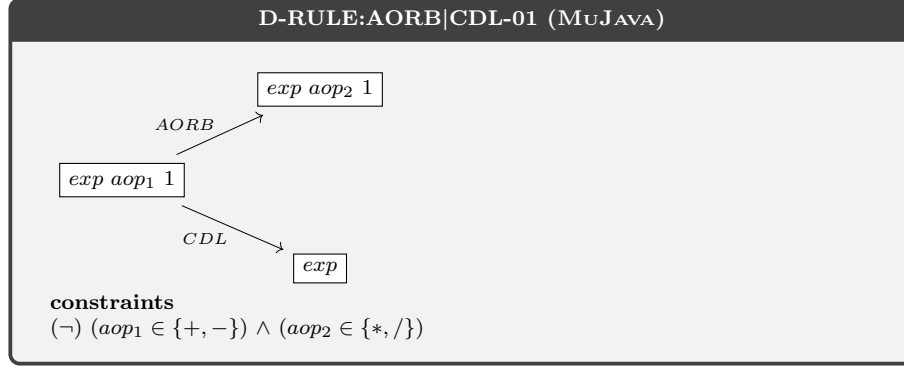


1.3.6 AOIU|AOIU-01 (MuJAVA)



Obs 1. This D-RULE works similar to LOI|LOI-01 (MuJAVA).

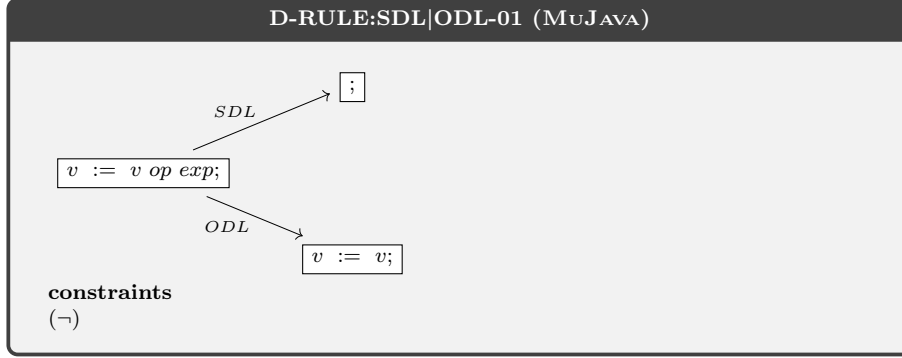
1.3.7 AORB|CDL-01 (MuJAVa)



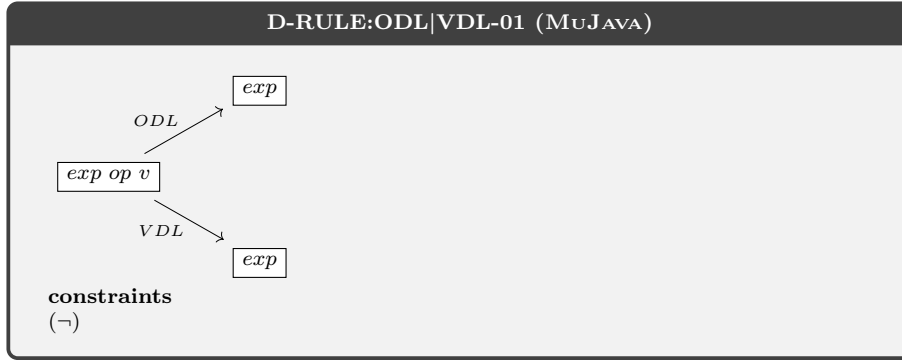
1.3.8 SDL|VDL-01 (MuJAVa)



1.3.9 SDL|ODL-01 (MuJAVa)



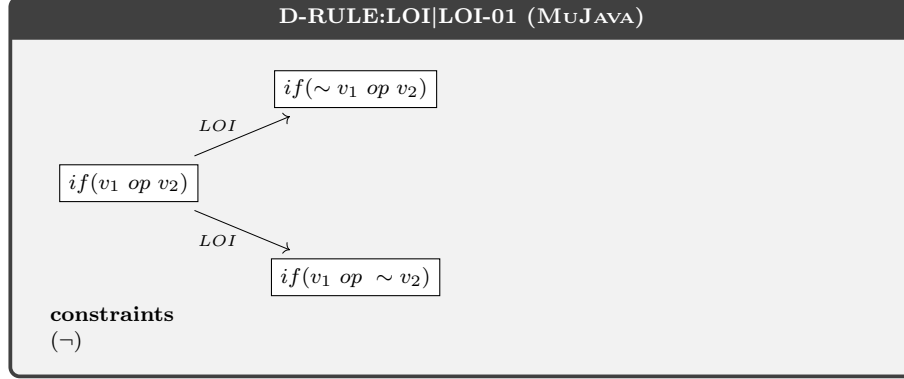
1.3.10 ODL|VDL-01 (MuJAVa)



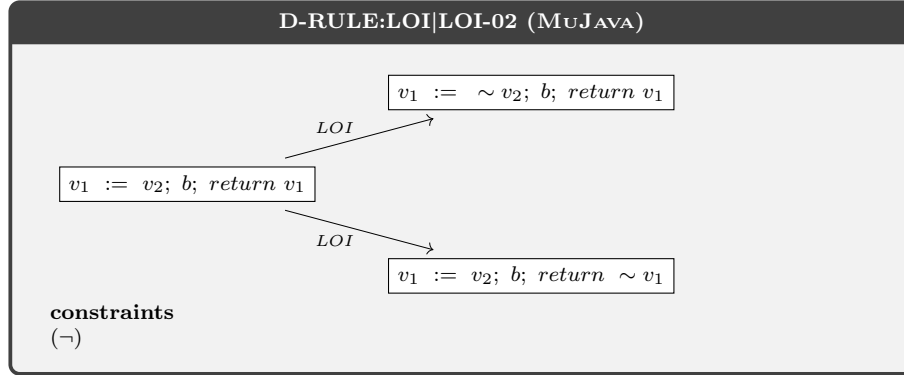
1.3.11 ODL|AODS-01 (MuJAVa)



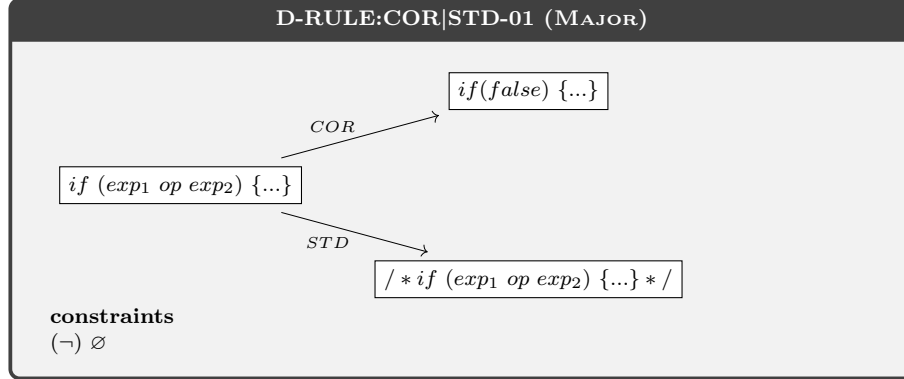
1.3.12 LOI|LOI-01 (MuJAVa)



1.3.13 LOI|LOI-02 (MuJAVa)

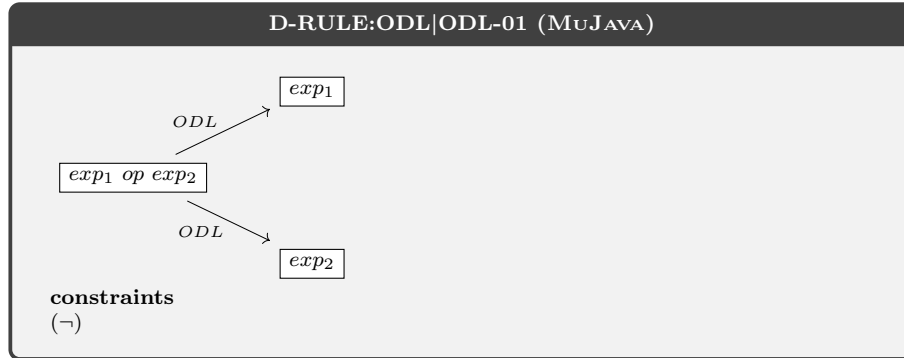


1.3.14 COR|STD-01 (MAJOR)

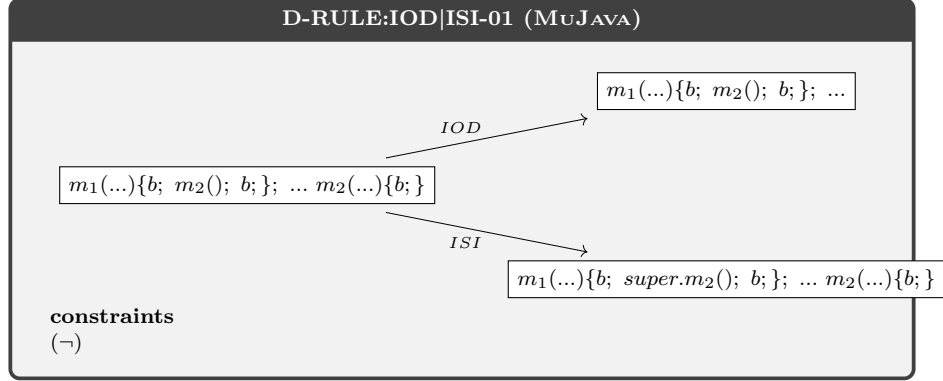


Obs 1. This D-RULE also applies to MUJAV (see ??).

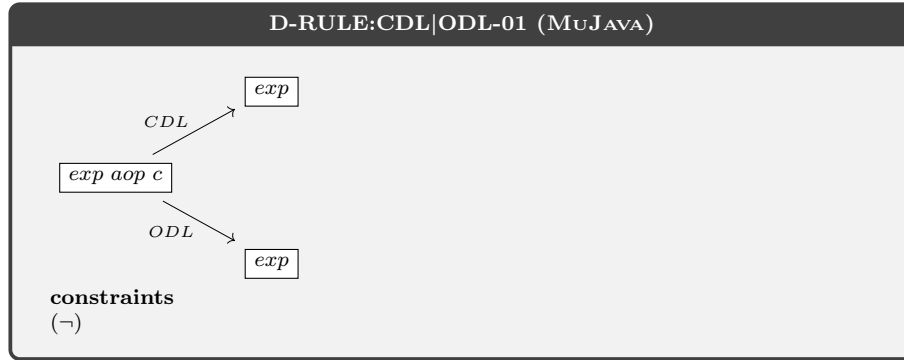
1.3.15 ODL|ODL-01 (MUJAV)



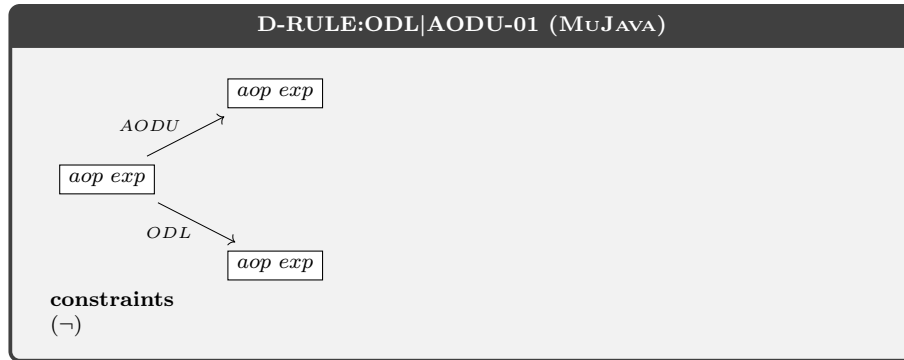
1.3.16 IOD|ISI-01 (MuJAVa)



1.3.17 CDL|ODL-01 (MuJAVa)



1.3.18 ODL|AODU-01 (MuJAVa)



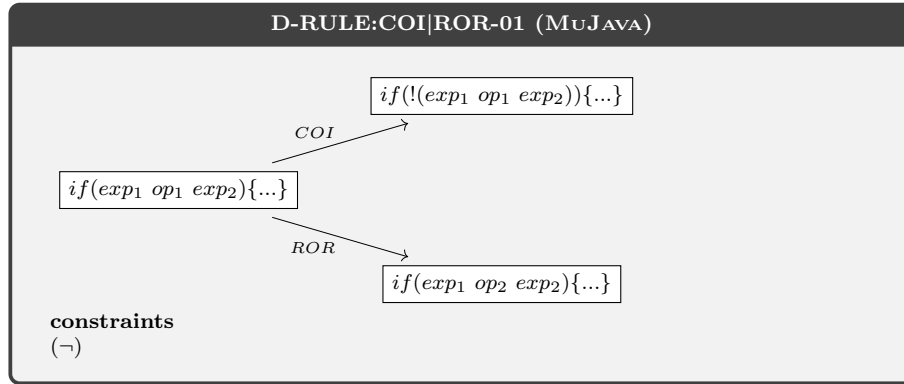
1.3.19 COD|ODL-01 (MuJAVa)



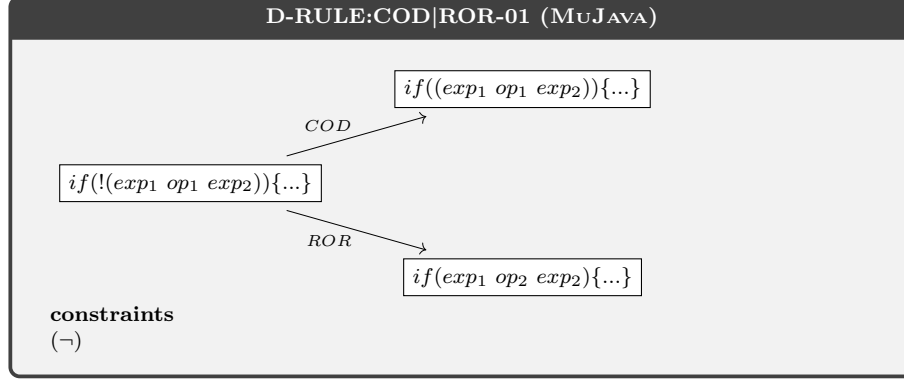
1.3.20 LOD|ODL-01 (MuJAVa)



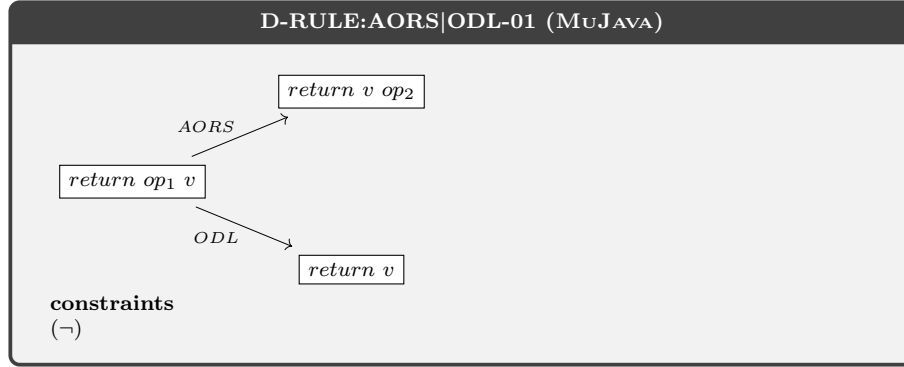
1.3.21 COI|ROR-01 (MuJAVa)



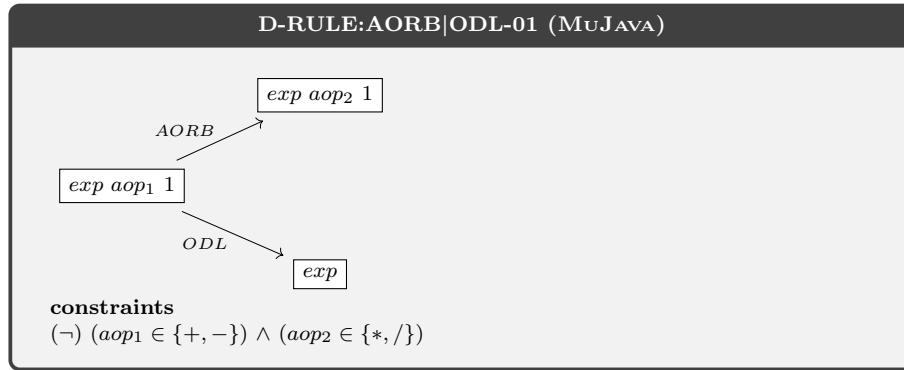
1.3.22 COD|ROR-01 (MuJAVa)



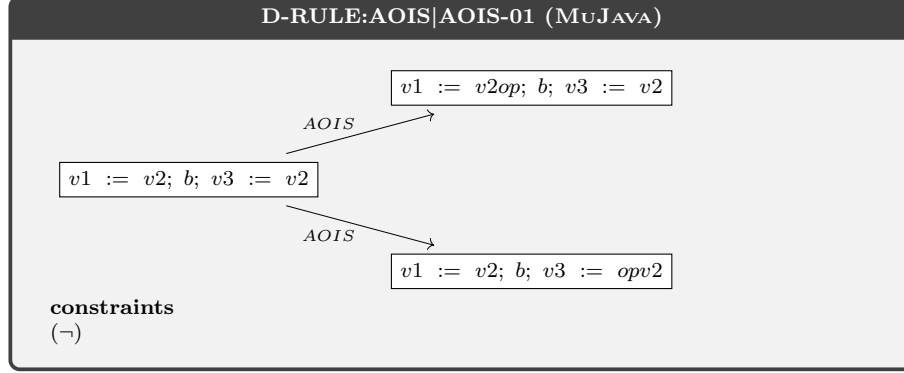
1.3.23 AORS|ODL-01 (MuJAVa)



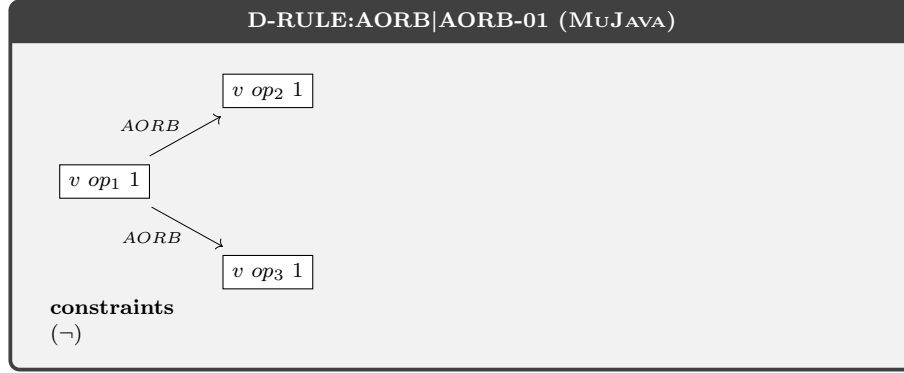
1.3.24 AORB|ODL-01 (MuJAVa)



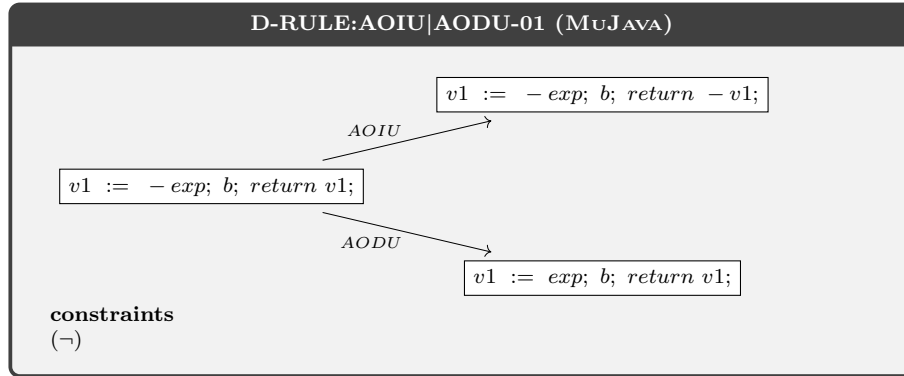
1.3.25 AOIS|AOIS-01 (MuJAVA)



1.3.26 AORB|AORB-01 (MuJAVA)



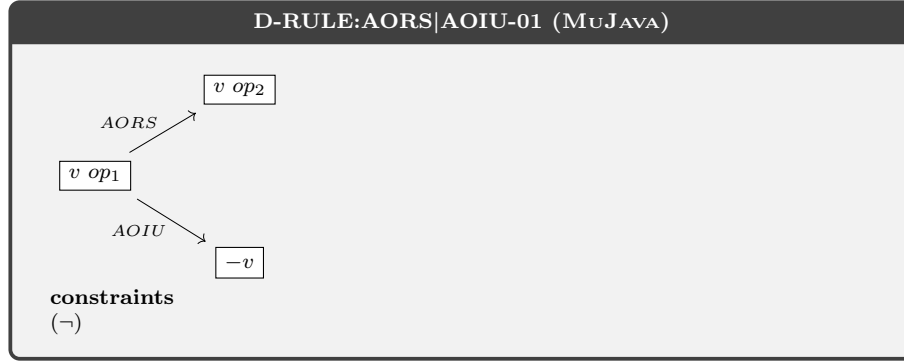
1.3.27 AOIU|AODU-01 (MuJAVA)



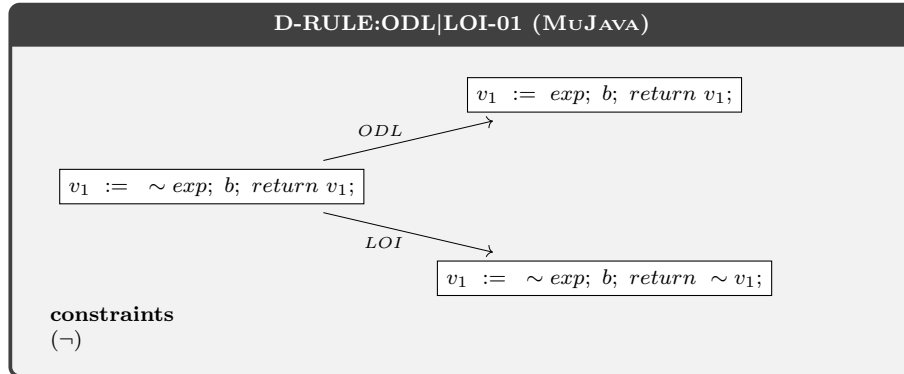
1.3.28 AORS|LOI-01 (MuJAVa)



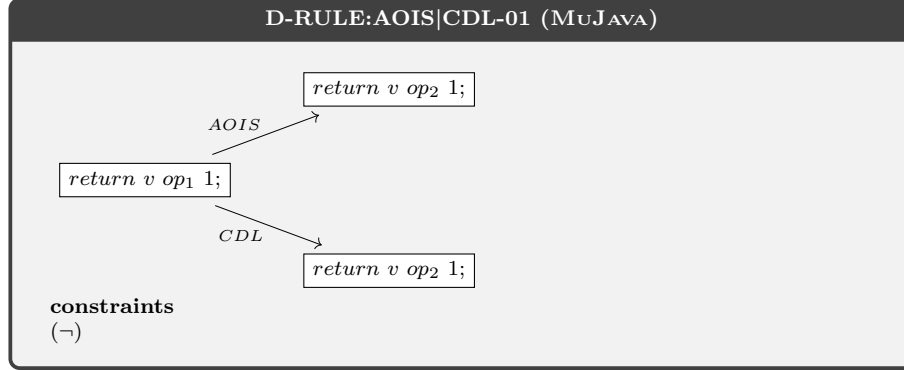
1.3.29 AORS|AOIU-01 (MuJAVa)



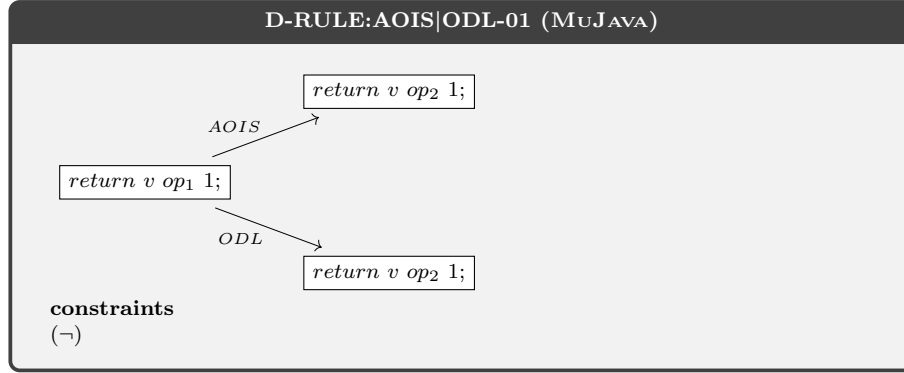
1.3.30 ODL|LOI-01 (MuJAVa)



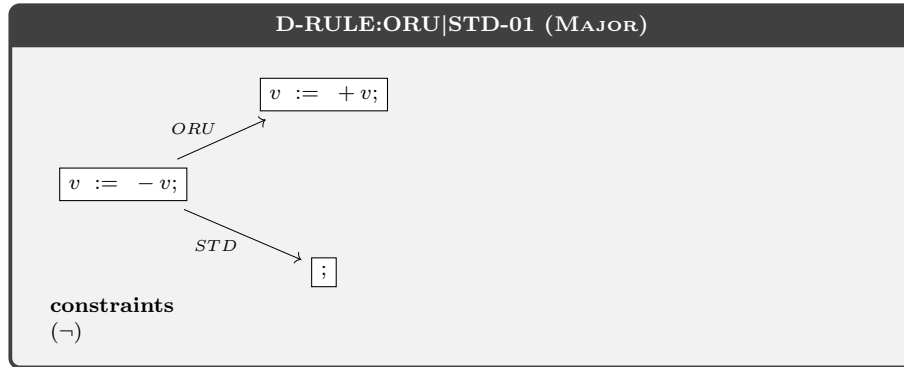
1.3.31 AOIS|CDL-01 (MuJAVa)



1.3.32 AOIS|ODL-01 (MuJAVa)



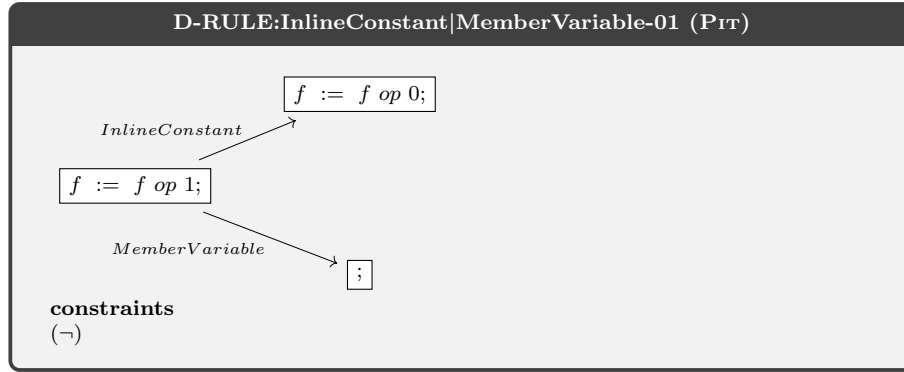
1.3.33 ORU|STD-01 (MAJOR)



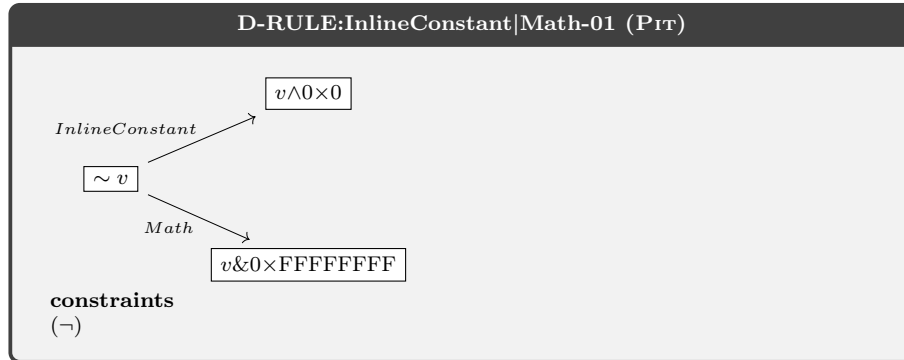
1.3.34 AOR|LVR-01 (MAJOR)



1.3.35 InlineConstant|MemberVariable-01 (PIT)

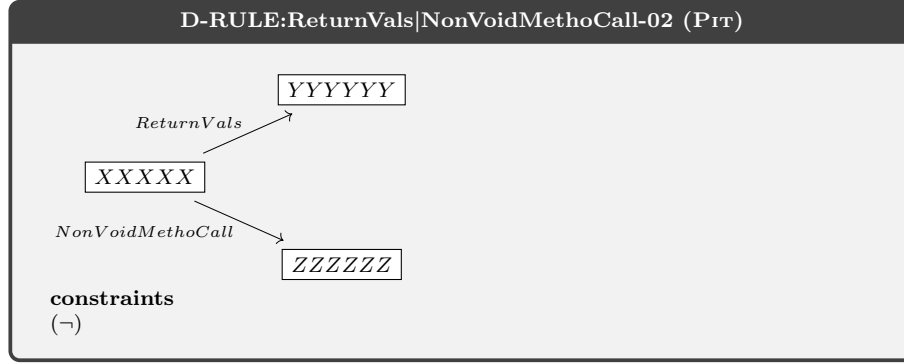


1.3.36 InlineConstant|Math-01 (PIT)

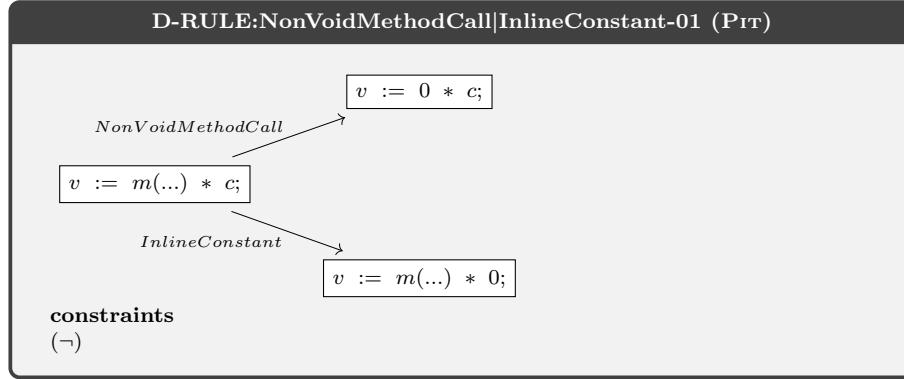


Obs 1. In byte code: $\sim v$ transform to $v \wedge 0 \times \text{FFFFFFFF}$

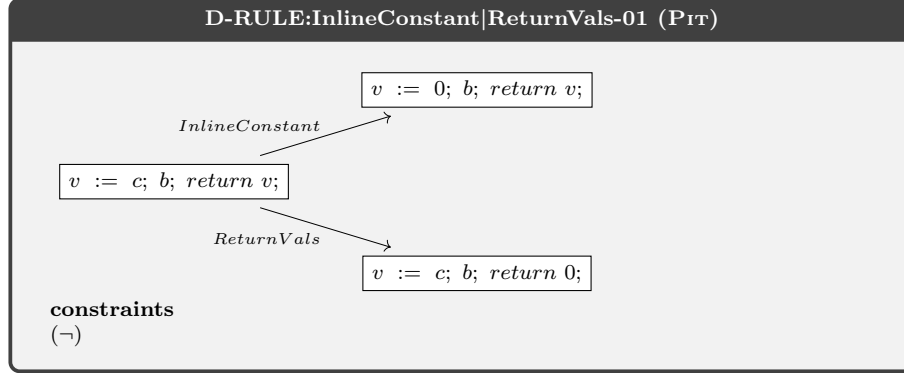
1.3.37 ReturnVals|NonVoidMethoCall-02 (PIT)



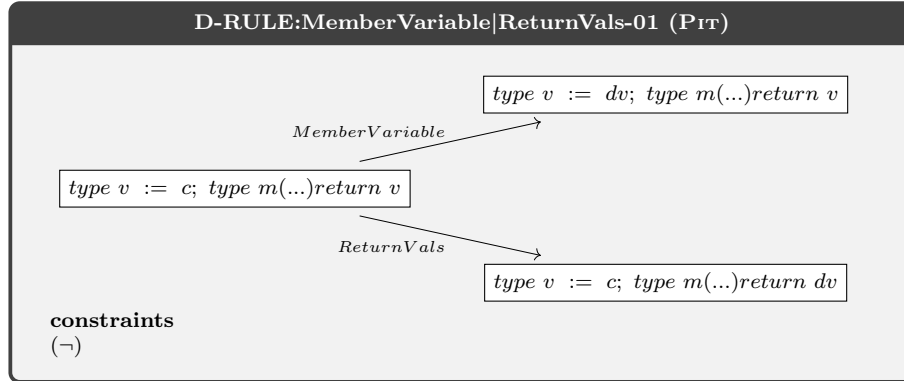
1.3.38 NonVoidMethodCall|InlineConstant-01 (PIT)



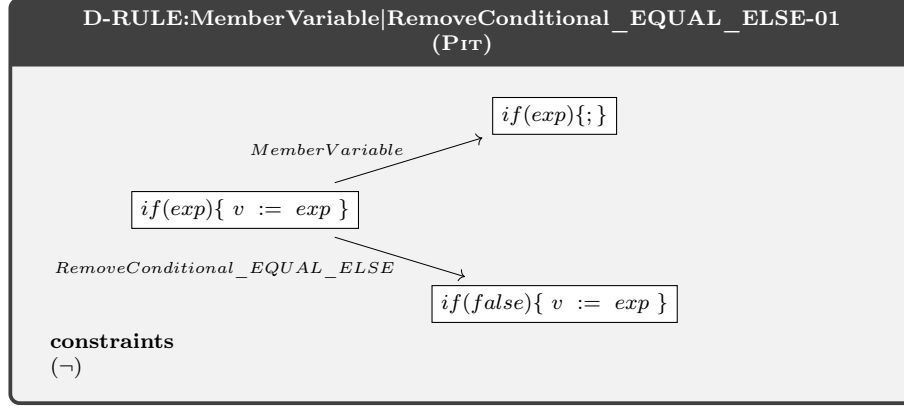
1.3.39 InlineConstant|ReturnVals-01 (PIT)



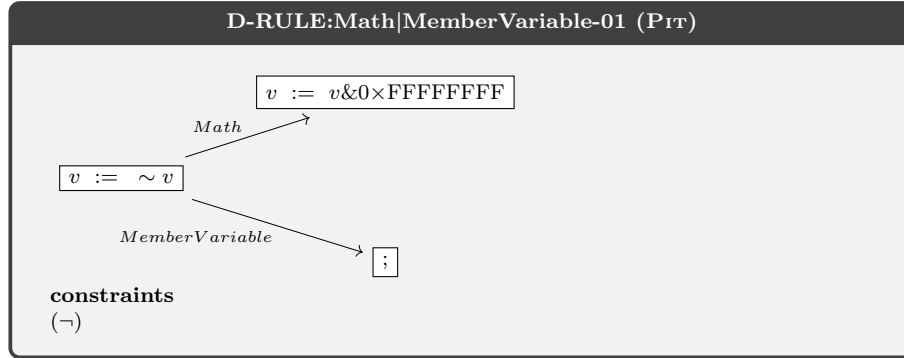
1.3.40 MemberVariable|ReturnVals-01 (PIT)



1.3.41 MemberVariable|RemoveConditional_EQUAL_ELSE-01 (PIT)

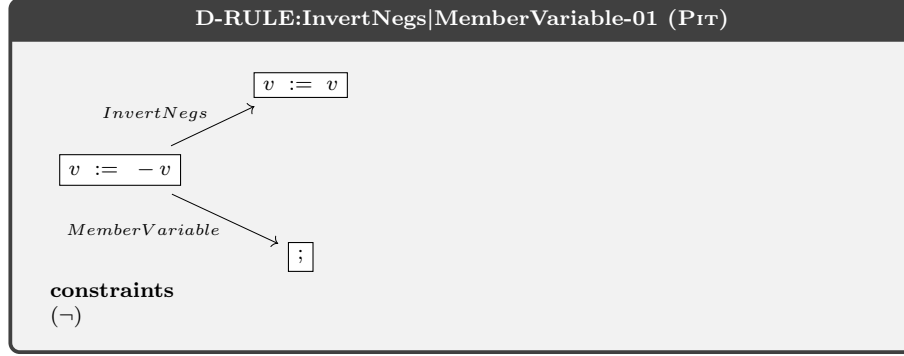


1.3.42 Math|MemberVariable-01 (PIT)

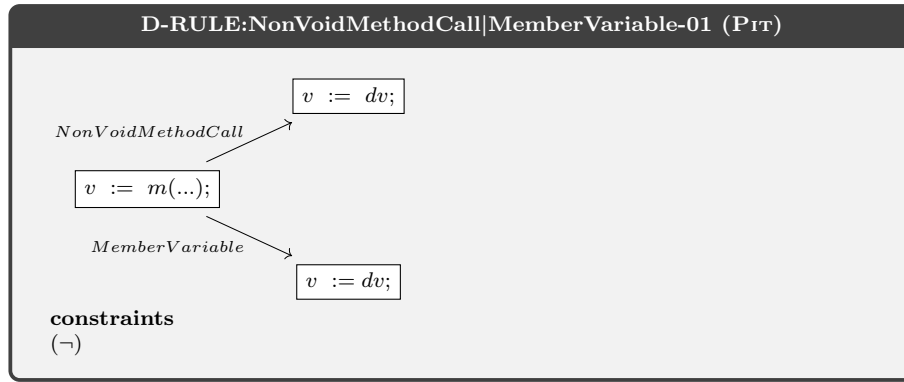


Obs 1. In byte code: $\sim v$ transform to $v \wedge 0 \times \text{FFFFFFFF}$

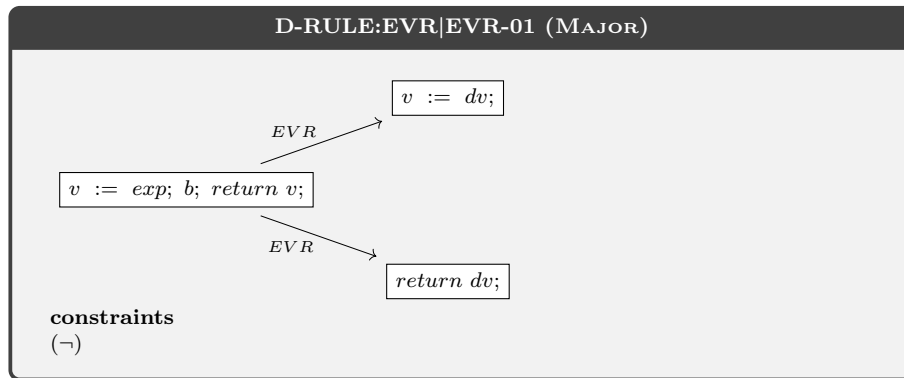
1.3.43 InvertNegs|MemberVariable-01 (PIT)



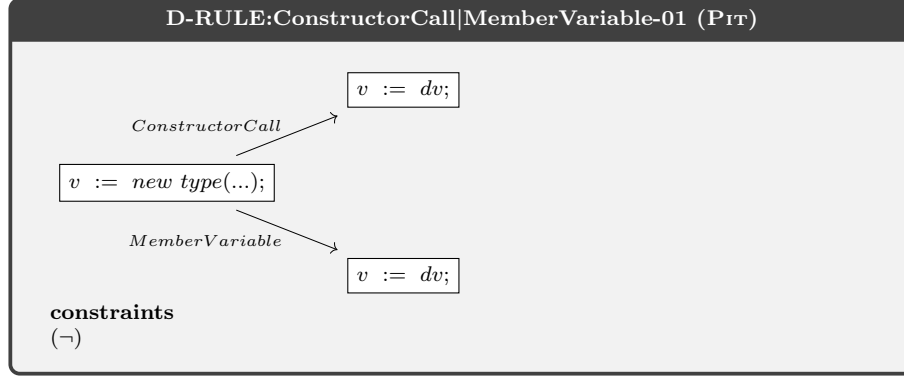
1.3.44 NonVoidMethodCall|MemberVariable-01 (PIT)



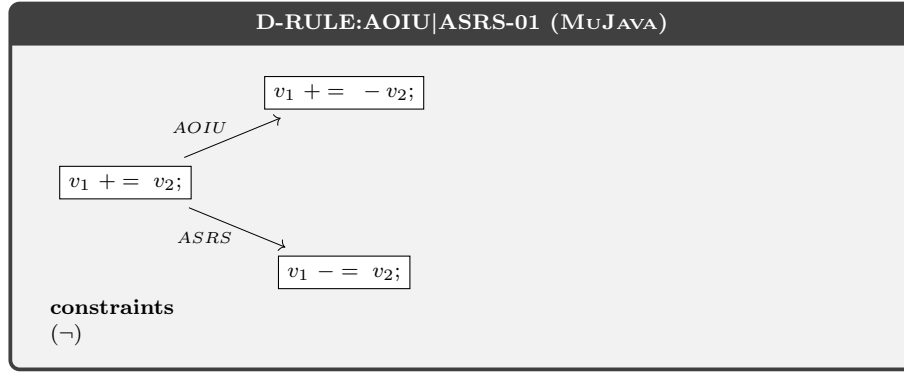
1.3.45 EVR|EVR-01 (MAJOR)



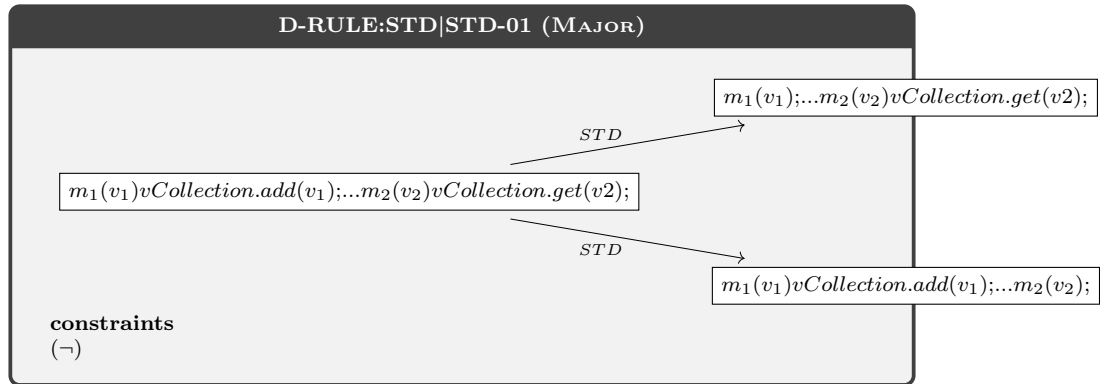
1.3.46 ConstructorCall|MemberVariable-01 (PIT)



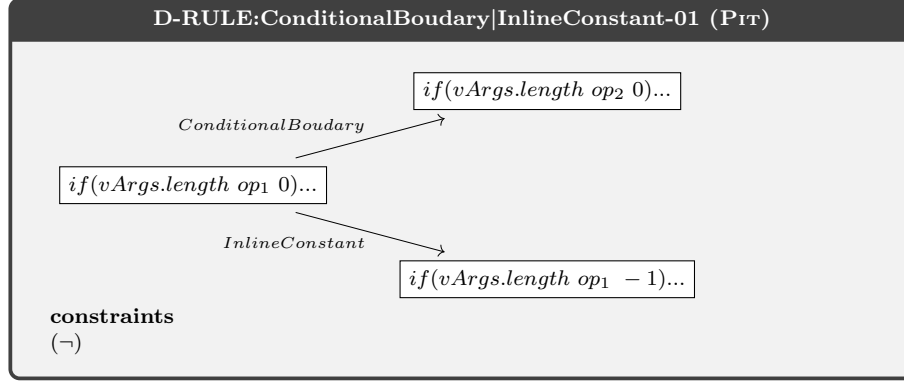
1.3.47 AOIU|ASRS-01 (MuJAva)



1.3.48 STD|STD-01 (MAJOR)

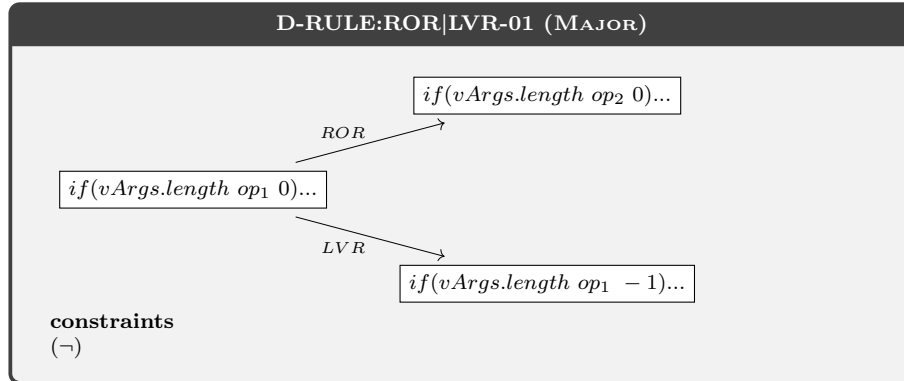


1.3.49 ConditionalBoudary|InlineConstant-01 (PIT)



Obs 1. This D-RULE also applies to MAJOR (see 1.3.50).

1.3.50 ROR|LVR-01 (MAJOR)



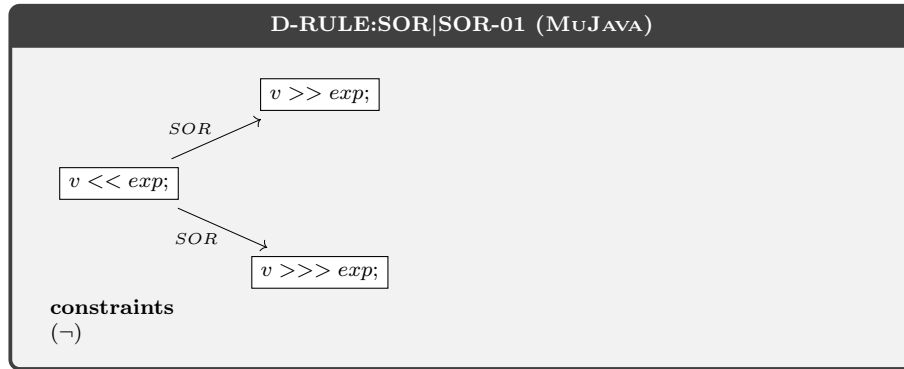
Obs 1. This D-RULE also applies to PIT (see 1.3.49).

1.3.51 SOR|SOR-01 (MAJOR)



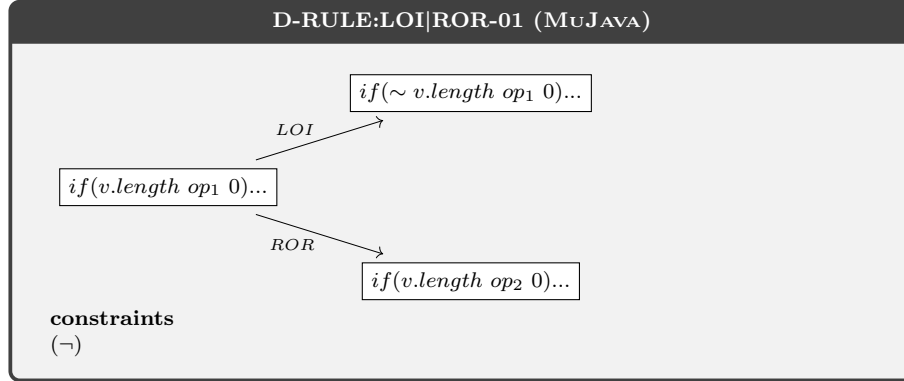
Obs 1. This D-RULE also applies to MUJAVA (see [1.3.52](#)).

1.3.52 SOR|SOR-01 (MUJAVA)

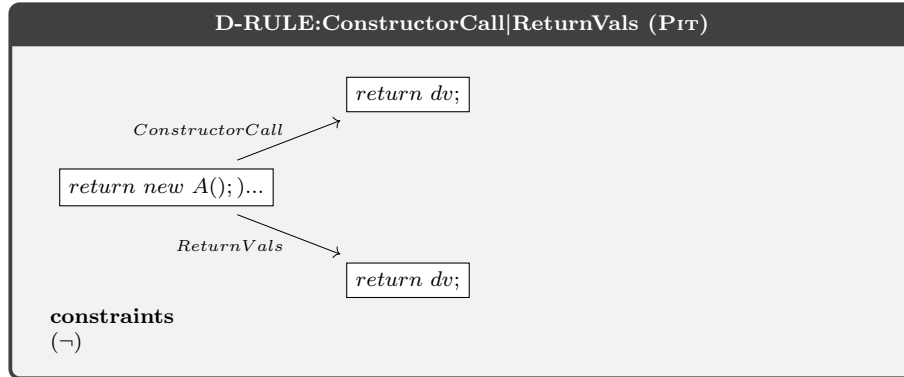


Obs 1. This D-RULE also applies to MAJOR (see [1.3.51](#)).

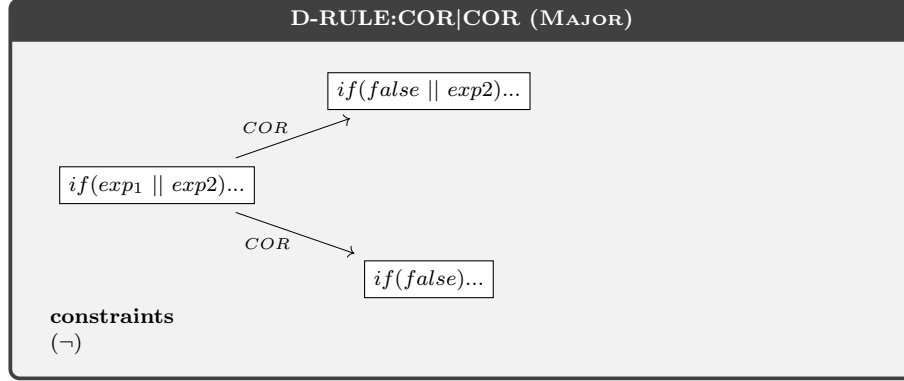
1.3.53 LOI|ROR-01 (MuJAvA)



1.3.54 ConstructorCall|ReturnVals-01 (PIT)

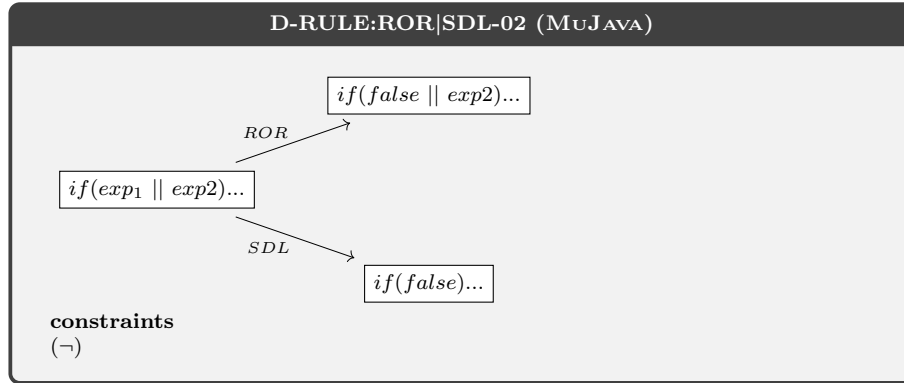


1.3.55 COR|COR-01 (MAJOR)



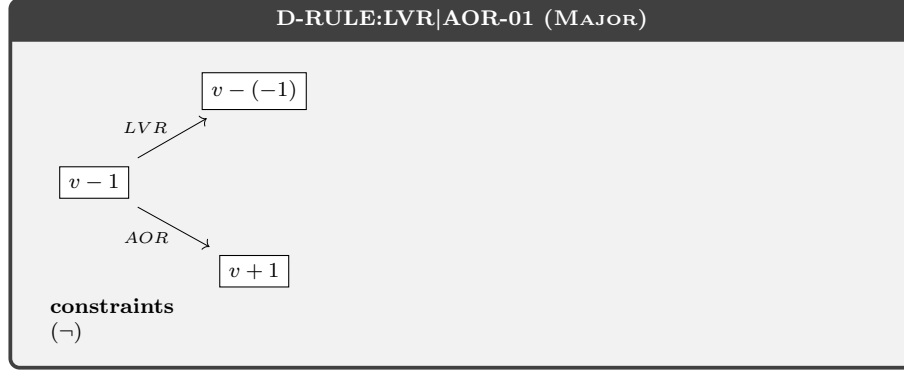
Obs 1. This D-RULE also applies to MUJAV (see 1.3.56).

1.3.56 ROR|SDL-02 (MUJAV)

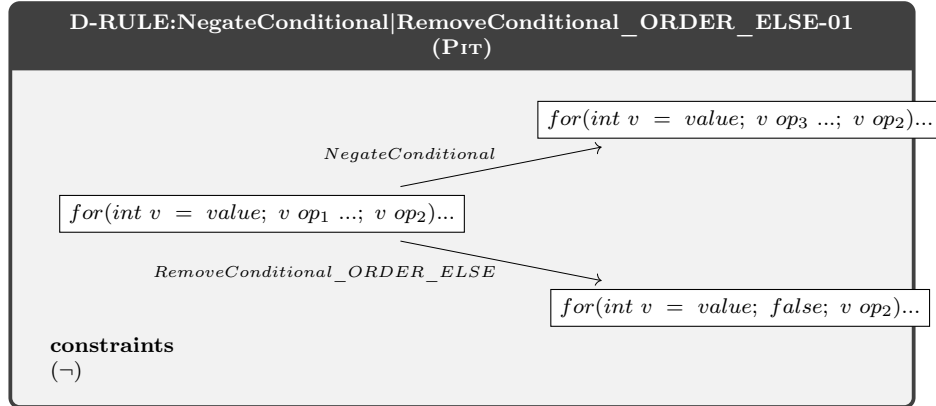


Obs 1. This D-RULE also applies to MAJOR (see 1.3.55).

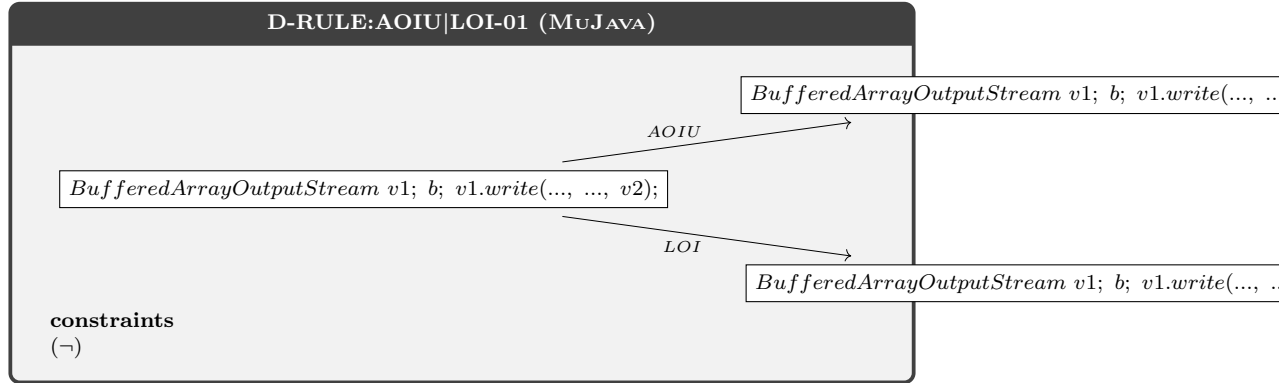
1.3.57 LVR|AOR-01 (MAJOR)



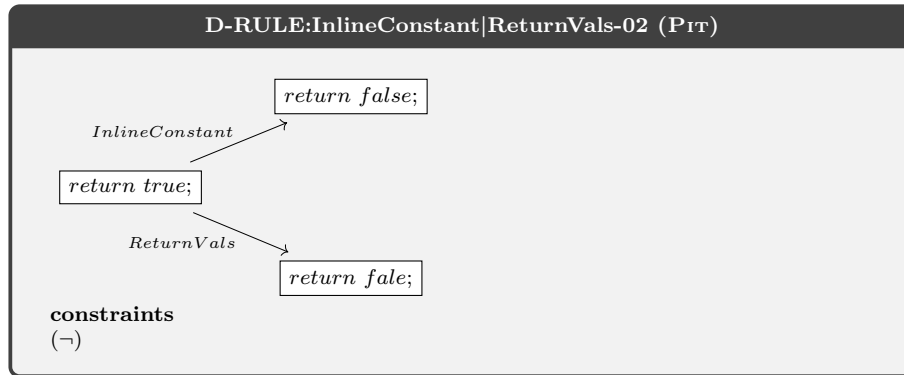
1.3.58 NegateConditional|RemoveConditional_ORDER_ELSE-01 (PIT)



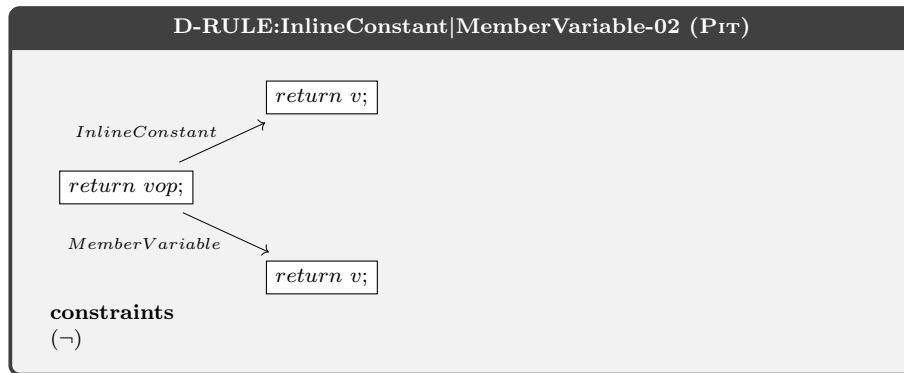
1.3.59 AOIU|LOI-01 (MuJAVa)



1.3.60 InlineConstant|ReturnVals-02 (PIT)



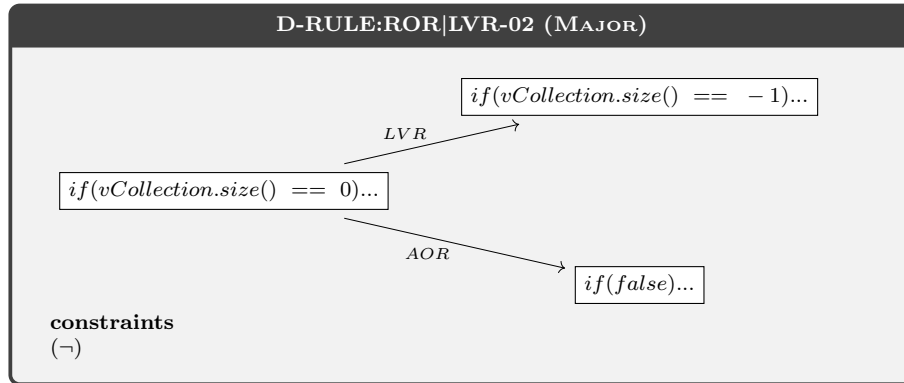
1.3.61 InlineConstant|MemberVariable-02 (PIT)



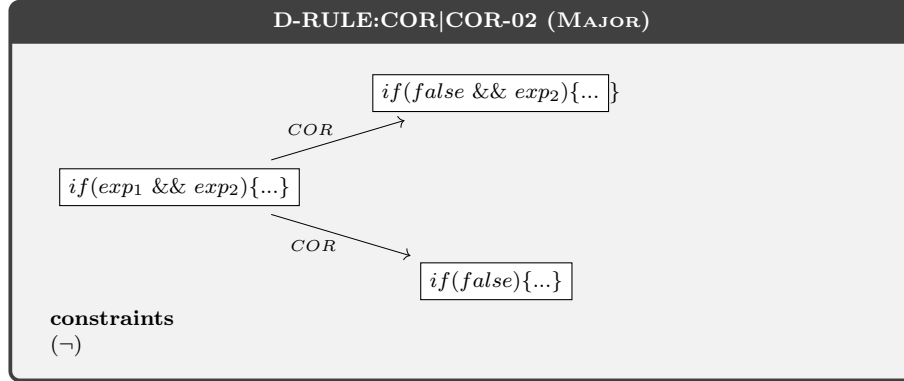
1.3.62 LVR|AOR-02 (MAJOR)



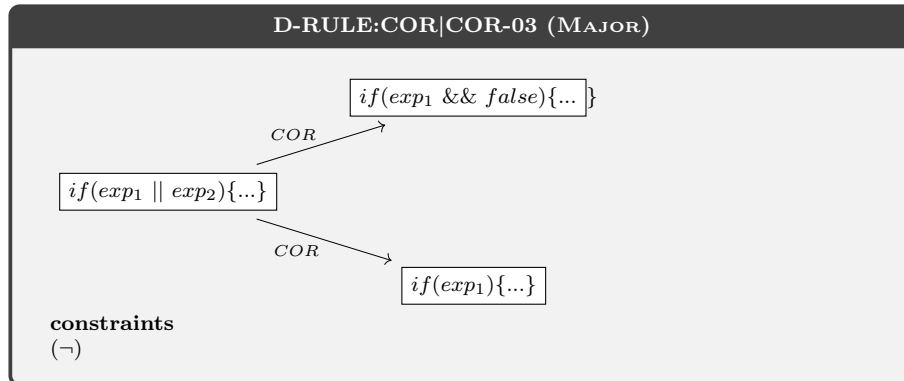
1.3.63 ROR|LVR-02 (MAJOR)



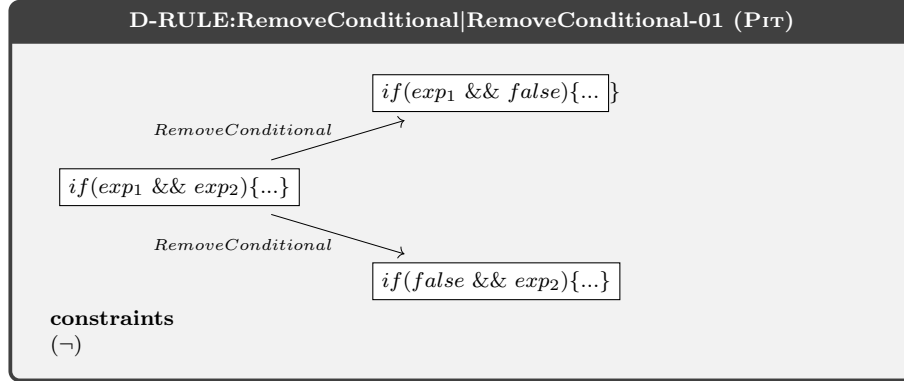
1.3.64 COR|COR-02 (MAJOR)



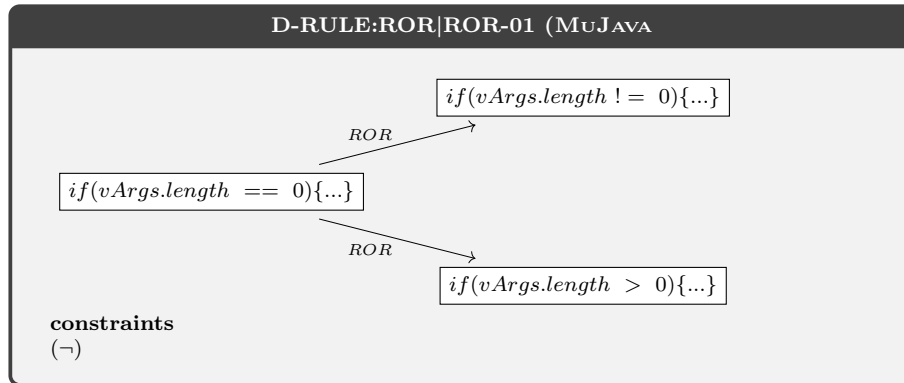
1.3.65 COR|COR-03 (MAJOR)



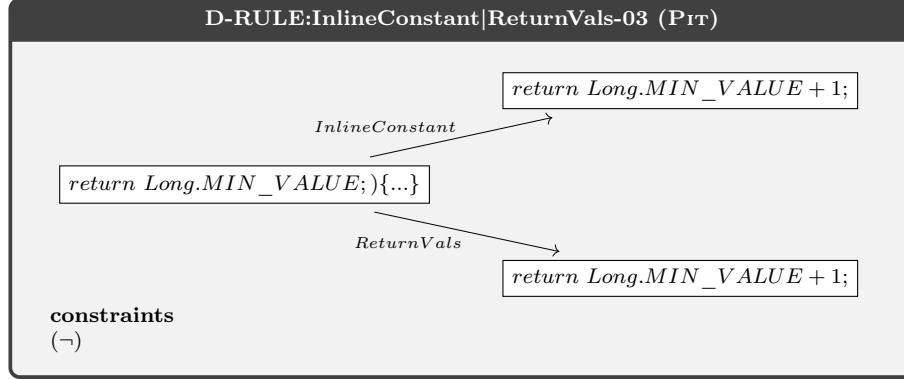
1.3.66 RemoveConditional|RemoveConditional-01 (PIT)



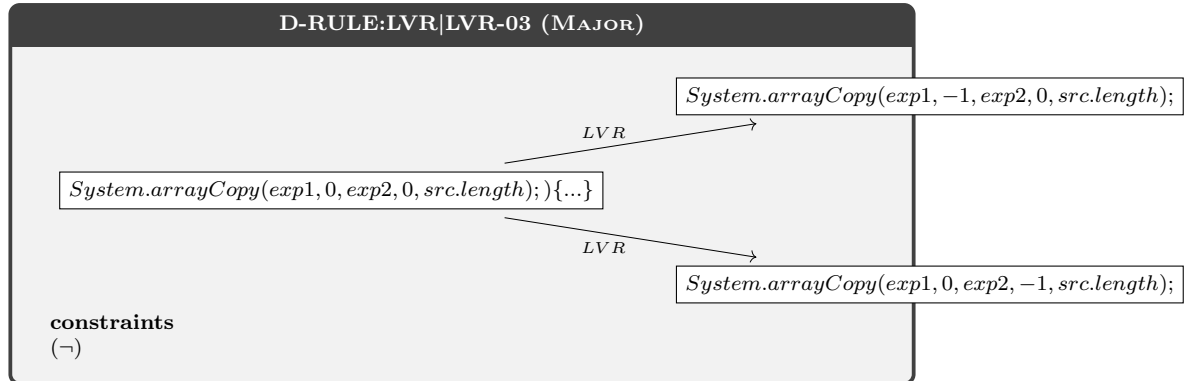
1.3.67 ROR|ROR-01 (MuJAVa)



1.3.68 InlineConstant|ReturnVals-03 (PIT)



1.3.69 LVR|LVR-03 (MAJOR)



Obs. In both cases they rise `IndexOutOfBoundsException`.

References

- [1] L. Fernandes, M. Ribeiro, P. Pinheiro, F. Ferrari, R. Gheyi, and A. Santos. Improving transformation rules to avoid useless mutants. 2020. To Appear.