Table 1: Conditions in DSL. This table shows the meaning of each kind of Condition.

Condition type	Meaning	Example in natural language
$StackCond(N, T)$	There are N operands with type T on the stack top	Assert: due to validation, **a value of value type i32 is on the top of the stack**
$SameTypeCond(N)$	N operands on the stack top are of the same type	Assert: due to validation, **two more values (of the same value type) are on the top of the stack**
$EqualCond(V_1, V_2)$	$V1$ is equal to $V2$	If $\boldsymbol{c}$ **is 0**, then: ...
$OpDefinedCond(Op)$	Op is a defined opcode	If $\boldsymbol{unop_t(c_1)}$ **is defined**, then: ...
$ExprCond(Expr)$	An equation or inequality represented as $Expr$ holds	Assert: due to validation, $\boldsymbol{x < dim(shape)}$
$ExistCond(Elem(Instance[idx]))$	The $Instance[idx]$ exists	Assert: due to validation, $\boldsymbol{S.mem[a]}$ **exists**
$InstPartCond(V)$	V is part of the instruction	If $\boldsymbol{N}$ **is part of the instruction**, then: ...
$CompareCond(V1, V2, R)$	$V1$ and $V2$ hold the comparison relation R	If $\boldsymbol{ea + N/8}$ **is larger than the length of** $\boldsymbol{mem.data}$, then: ...