

# Definition of Safety

Günther Wullaert

May 2022

## 1 Language

### 1.1 Term and Pools

We inductively define terms, tuples of terms, and pools as

- all numerals, symbolic constants, and variables are terms,
- $f(\mathbf{t})$  is a term, if  $f$  is a symbolic constant and  $\mathbf{t}$  is a pool,
- $t_1 \star t_2$  is a term, if  $\star$  is among the symbols  $+$ ,  $-$ ,  $\times$ ,  $/$  or  $..$  and  $t_1, t_2$  are terms,
- $\langle \mathbf{t} \rangle$  is a term, if  $\mathbf{t}$  is a pool, which can have a possibly empty set of terms,
- $t_1, \dots, t_n$  is a tuple of terms, if  $n \geq 0$  and  $t_i$  is a term,
- $\dot{t}_1; \dots; \dot{t}_n$  is a pool, if  $n \geq 1$  and each  $\dot{t}_i$  is a tuple of terms.

### 1.2 Constants

We inductively define a term to be *constant* if

- it is a numeral,
- it has form  $t \star u$  where  $t$  and  $u$  are *constant* and  $\star$  is among the symbols  $+$ ,  $-$ ,  $\times$  or  $/$ .

### 1.3 Atoms and Literals

An atom has form  $p(\mathbf{t})$  where  $p$  is a predicate symbol and  $\mathbf{t}$  is a pool.

A literal is either an atom, atom preceded by not, comparison literal or comparison literal preceded by not.

A comparison literal has form  $t_1 \prec t_2$ , where  $t_1, t_2$  are terms and  $\prec$  is among the symbols  $\leq, \geq, <, >$  or  $\neq$ .

A conditional literal has form  $\dot{t} : \dot{c}$ , where  $t$  and  $c$  are tuples.

## 1.4 Aggregates

An aggregate has one of the forms

$$\alpha\{l_1; \dots; l_n\} \prec_1 s_1 \quad (1)$$

$$s_1 \prec_1 \alpha\{l_1; \dots; l_n\} \quad (2)$$

$$s_1 \prec_1 \alpha\{l_1; \dots; l_n\} \prec_2 s_2 \quad (3)$$

( $n \geq 0$ ), where

- $\alpha$  is an aggregate name,
- each  $l_i$  is a conditional literal,
- each  $\prec_1, \prec_2$  is among the symbols  $\leq, \geq, <, >, =$  or  $\neq$ ,
- each  $s_1, s_2$  is a term.

## 1.5 Rules

A rule  $r$  has the form

$$H_1 \vee \dots \vee H_m \leftarrow B_1 \wedge \dots \wedge B_n \quad (4)$$

( $m, n \geq 0$ ), where each  $H_i$  and  $B_j$  are a literal.  $H_1 \vee \dots \vee H_m$  is called the head.  $B_1 \wedge \dots \wedge B_n$  is called the body.

## 2 Safety

We define operator  $C_r$  for a rule  $r$  applied to a set of variables  $V$  as

$$C_r(V) = \bigcup_{(P,D) \in \text{dep}(r), D \subseteq V} P.$$

A rule  $r$  is safe if  $\text{vars}(r)$  is the least fixed point of  $C_r$ .

The  $\text{vars}(e)$  function returns all variables for an expression  $e$ .  
For Example:

$$\text{vars}(a(X) = b(Y)) = \{X, Y\}$$

The  $\text{eval}(c)$  function takes a constant term and returns the arithmetical evaluation of that term. If this function for a *constant* returns a value that is not zero a *constant* is called *nonzero*.

$$\begin{aligned} \text{eval}(c) &= c \\ \text{eval}(a + b) &= (\text{eval}(a) + \text{eval}(b)) \\ \text{eval}(a - b) &= (\text{eval}(a) - \text{eval}(b)) \\ \text{eval}(a \times b) &= (\text{eval}(a) \times \text{eval}(b)) \\ \text{eval}(a/b) &= (\text{eval}(a) \div \text{eval}(b)) \end{aligned}$$

For Example:

$$\begin{aligned}
eval(5 + 3 \times 2) &= (eval(5 + 3) \times eval(2)) \\
&= ((eval(5) + eval(3)) \times 2) \\
&= ((5 + 3) \times 2) = (8 \times 2) = 16
\end{aligned}$$

## 2.1 Terms

### 2.1.1 Constants

For any numeral  $n$  and symbolic constant  $f$

$$pt(n) = pt(f) = dt(n) = dt(f) = \emptyset$$

### 2.1.2 Variables

For any variable  $X$

$$\begin{aligned}
pt(X) &= \{X\} \\
dt(X) &= \emptyset
\end{aligned}$$

### 2.1.3 Tuples

For any tuple of terms  $t_1, \dots, t_n$

$$\begin{aligned}
pt(t_1, \dots, t_n) &= pt(t_1) \cup \dots \cup pt(t_n) \\
dt(t_1, \dots, t_n) &= dt(t_1) \cup \dots \cup dt(t_n)
\end{aligned}$$

### 2.1.4 Pools

For any pool of terms  $\dot{t}_1; \dots; \dot{t}_n$

$$\begin{aligned}
pt(\dot{t}_1; \dots; \dot{t}_n) &= pt(\dot{t}_1) \cap \dots \cap pt(\dot{t}_n) \\
dt(\dot{t}_1; \dots; \dot{t}_n) &= dt(\dot{t}_1) \cup \dots \cup dt(\dot{t}_n)
\end{aligned}$$

### 2.1.5 Functions

For a term of form  $f(\mathbf{t})$ , where  $f$  is a function and  $\mathbf{t}$  a pool

$$\begin{aligned}
pt(f(\mathbf{t})) &= pt(\mathbf{t}) \\
dt(f(\mathbf{t})) &= dt(\mathbf{t})
\end{aligned}$$

### 2.1.6 Arithmetics

For a term of form  $t \times c$ , where  $t$  is a term and  $c$  a constant and  $eval(c) = 0$

$$\begin{aligned} pt(t \times c) &= \emptyset \\ dt(t \times c) &= \emptyset \end{aligned}$$

For a term of form  $a \star b$ , where  $a, b$  are terms and one of them is a *constant* and the *constant* is *nonzero* and  $\star$  is among the symbols  $+$ ,  $-$  or  $\times$  or  $a$  and  $b$  are both *constant* and  $\star$  is among the symbols  $+$ ,  $-$ ,  $\times$ ,  $/$  or  $..$

$$\begin{aligned} pt(a \star b) &= pt(a) \cup pt(b) \\ dt(a \star b) &= dt(a) \cup dt(b) \end{aligned}$$

Otherwise for a term of form  $a \star b$

$$\begin{aligned} pt(a \star b) &= \emptyset \\ pt(a \star b) &= vars(a \star b) \end{aligned}$$

For a term of form  $-t$ , where  $t$  is a term

$$\begin{aligned} pt(-t) &= pt(t) \\ dt(-t) &= dt(t) \end{aligned}$$

## 2.2 Atoms and Literals

### 2.2.1 Atoms and Literals

For an literal of form *not*  $a$ , where  $a$  is an atom

$$dep(not\ a) = \{(\emptyset, vars(a))\}$$

For an atom of form  $p(\mathbf{t})$ , where  $\mathbf{t}$  is a pool

$$dep(p(\mathbf{t})) = \{(pt(\mathbf{t}), \emptyset), (\emptyset, dt(\mathbf{t}))\}$$

### 2.2.2 Comparison Literals

For an comparison literal of form  $a \prec b$ , where  $a$  and  $b$  are terms and  $\prec$  is among the symbols  $\leq, \geq, <, >, \neq$

$$dep(a \prec b) = \{(\emptyset, vars(a \prec b))\}$$

For an comparison literal of form  $a = b$ , where  $a$  and  $b$  are terms

$$dep(a = b) = \{(pt(a), vars(b)), (pt(b), vars(a)), (\emptyset, dt(a) \cup dt(b))\}$$

For an literal of form  $\text{not } a \prec b$ , where  $a$  and  $b$  are terms

$$\begin{aligned} \text{dep}(\text{not } a = b) &= \text{dep}(a \neq b) \\ \text{dep}(\text{not } a \neq b) &= \text{dep}(a = b) \\ \text{dep}(\text{not } a \leq b) &= \text{dep}(a > b) \\ \text{dep}(\text{not } a > b) &= \text{dep}(a \leq b) \\ \text{dep}(\text{not } a \geq b) &= \text{dep}(a < b) \\ \text{dep}(\text{not } a < b) &= \text{dep}(a \geq b) \end{aligned}$$

### 2.2.3 Conditional Literals

For an literal of form  $t : \dot{c}$ , where  $t$  and  $c$  are tuples

$$\text{dep}(t : \dot{c}) = \emptyset$$

## 2.3 Aggregates

For an aggregate  $a$  of form (1) or (2), where  $\prec_1$  is =

$$\text{dep}(a) = \{(\text{vars}(s_1), \emptyset)\}$$

Otherwise and for an aggregate  $a$  of form (3)

$$\text{dep}(a) = \emptyset$$

## 2.4 Rule

For a rule  $r$  in the form of (4) the following holds:

$$\text{dep}(r) = \{(\emptyset, \text{vars}(H_1 \vee \dots \vee H_k))\} \cup \text{dep}(B_1) \cup \dots \cup \text{dep}(B_m)$$

## 3 Other Examples

$$\begin{aligned} \text{dep}(p(X, Y + Y)) &= \{(pt(X, Y + Y), \emptyset), (\emptyset, dt(X, Y + Y))\} \\ &= \{(pt(X) \cup pt(Y + Y), \emptyset), (\emptyset, dt(X) \cup dt(Y + Y))\} \\ &= \{(\{X\} \cup \emptyset, \emptyset), (\emptyset, \emptyset \cup \text{vars}(Y + Y))\} \\ &= \{(\{X\}, \emptyset), (\emptyset, \{Y\})\} \end{aligned}$$