# 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
- $f(\boldsymbol{t})$ is a term, if $f$ is a symbolic constant and $\boldsymbol{t}$ is a pool
- $(t_1 \star t_2)$ is a term, if $t_1$ and $t_2$ are terms and $\star$ is one of the symbols (+ - $\times$ / ..)
- $\langle \boldsymbol{t} \rangle$ is a term, if $\boldsymbol{t}$ is a pool, which can have a possibly empty set of terms.
- $t_1, ..., t_n$ is a tuple of terms, if $n \geq 0$ and $t_i$ is a term.
- $\dot{t_1}; ...; \dot{t_n}$ is a pool, if $n \geq 1$ and each $\dot{t_i}$ is a tuple of terms. In particular, every tuple of terms is a pool.

## 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 one of the symbols (+ - $\times$ /)

## 1.3 Atoms and Literals

An atom has form $p(\boldsymbol{t})$ where $p$ is a predicate symbol and $\boldsymbol{t}$ is a pool.
A literal is either an atom or atom preceded by not.

## 1.4 Comparisons

A comparison has form $t_1 \prec t_2$, where $t_1, t_2$ are terms and $\prec$ is on of the symbols $(\leq, \geq, <, >, \neq)$

### 1.5 Rules

A rule $r$ has the form

$$H_1 \vee ... \vee H_k \leftarrow B_1 \wedge ... \wedge B_m \tag{1}$$

$(k, m \geq 0)$, where each $H_i$ is either a symbolic literal, literal or an comparison and each $B_j$ is either a symbolic symbolic literal, literal or an comparison. $H_1 \vee ... \vee H_k$ is called the head. $B_1 \wedge ... \wedge B_m$ is called the body.

# 2 Safety

We define the function *provide*(). The provide function returns a set of pairs. Each pair has the form $(p, d)$, where $p$ is a set of variables the statement provides if $d$ variables are provided.

## 2.1 Helper Functions

The $vars(e)$ function returns all variables in an expression.
For Example:

$$vars(a(X) = b(Y)) = \{X, Y\} \tag{2}$$

## 2.2 Terms

### 2.2.1 Constants

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

$$pt(n) = pt(f) = dt(n) = dt(f) = \emptyset \tag{3}$$

### 2.2.2 Variables

For any variable $X$:

$$pt(X) = \{X\} \tag{4}$$
$$dt(X) = \emptyset \tag{5}$$

### 2.2.3 Tuples

For any tuple of terms $t_1, ..., t_n$:

$$pt(t_1, ..., t_n) = pt(t_1) \cup \ldots \cup pt(t_n) \tag{6}$$
$$dt(t_1, ..., t_n) = dt(t_1) \cup \ldots \cup dt(t_n) \tag{7}$$

### 2.2.4 Pools

For any pool of terms $\dot{t_1}; ...; \dot{t_n}$:

$$pt(\dot{t_1}; ...; \dot{t_n}) = pt(\dot{t_1}) \cap \ldots \cap pt(\dot{t_n}) \tag{8}$$
$$dt(\dot{t_1}; ...; \dot{t_n}) = dt(\dot{t_1}) \cup \ldots \cup dt(\dot{t_n}) \tag{9}$$

### 2.2.5 Terms

For a term of form $f(\boldsymbol{t})$, where $f$ a symbolic constant and $\boldsymbol{t}$ a pool:

$$pt(f(\boldsymbol{t})) = pt(\boldsymbol{t}) \tag{10}$$
$$dt(f(\boldsymbol{t})) = dt(\boldsymbol{t}) \tag{11}$$

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

$$pt(a \star b) = pt(b \star a) = pt(a) \cup pt(b) \tag{12}$$
$$dt(a \star b) = dt(b \star a) = dt(a) \cup dt(b) \tag{13}$$

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

$$pt(a \star b) = \emptyset \tag{14}$$
$$pt(a \star b) = vars(a \star b) \tag{15}$$

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

$$pt(-t) = pt(t) \tag{16}$$
$$dt(-t) = dt(t) \tag{17}$$

For a term of form $t * 0$, where $t$ is a term:

$$pt(t * 0) = pt(0 * t) = \emptyset \tag{18}$$
$$dt(t * 0) = dt(0 * t) = \emptyset \tag{19}$$

## 2.3 Atoms and Literals

### 2.3.1 Atoms

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

$$dep(p(\boldsymbol{t})) = \{(pt(\boldsymbol{t}), \emptyset), (\emptyset, dt(\boldsymbol{t}))\} \tag{20}$$

### 2.3.2 Literals

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

$$dep(not\ a) = \{(\emptyset, vars(a))\} \tag{21}$$

For an literal of form $a$, where $a$ is an atom:

$$dep(a) = dep(a) \tag{22}$$

For an literal of form *not* $l$, where $l$ is another literal:

$$dep(not\ l) = \{(\emptyset, vars(a))\} \tag{23}$$

## 2.4 Comparisons

### 2.4.1 Comparisons

For an comparison of form $a \prec b$, where $a$ and $b$ are terms and $\prec$ is one of the symbols $(\leq, \geq, <, >, \neq)$:

$$dep(a \prec b) = \{(\emptyset, vars(a \prec b))\} \tag{24}$$

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

$$dep(a = b) = \{(pt(a), vars(b)), (pt(b), vars(a)), (\emptyset, dt(a) \cup dt(b))\} \tag{25}$$

## 2.5 Rule

For a rule $r$ in the form of

$$H_1 \vee ... \vee H_k \leftarrow B_1 \wedge ... \wedge B_m \tag{26}$$

The following holds:

$$dep(r) = \{(\emptyset, vars(H_1 \vee ... \vee H_k))\} \cup dep(B_1) \cup ... \cup dep(B_m) \tag{27}$$

# 3 Extras

If 2 pairs share the same $p$, the $d$ can be merged. If 2 pairs share the same $d$, the $p$ can be merged.
For example:

$$\{(\{X\}, \{\}), (\{Y\}, \{\})\} = \{(\{X, Y\}, \{\})\} \tag{28}$$

$$\{(\{\}, \{X\}), (\{\}, \{Y\})\} = \{(\{\}, \{X, Y\})\} \tag{29}$$

# 4 Other Examples

$$\begin{aligned} dep(p(X, Y+Y)) &= \{(pt(X, Y+Y), \emptyset), (\emptyset, dt(X, Y+Y))\} \\ &= \{(pt(X) \cup pt(Y+Y), \emptyset), (\emptyset, dt(X) \cup dt(Y+Y))\} \\ &= \{(\{X\} \cup \emptyset, \emptyset), (\emptyset, \emptyset \cup vars(Y+Y))\} \\ &= \{(\{X\}, \emptyset), (\emptyset, \{Y\})\} \end{aligned} \tag{30}$$

$$\begin{aligned} &dep(a(Y) \leftarrow a(X), X = Y) \\ &\quad = \{(\emptyset, vars(a(Y)))\} \cup dep(a(X)) \cup dep(X{=}Y) \\ &\quad = \{(\emptyset, \{Y\})\} \cup \{(pt(a(X)), \emptyset), (\emptyset, dt(a(X)))\} \\ &\qquad \cup \{(pt(X), vars(Y)), (pt(Y), vars(X)), (\emptyset, dt(X) \cup dt(Y))\} \\ &\quad = \{(\emptyset, \{Y\}), (pt(X), \emptyset), (\emptyset, dt(X)), (\{X\}, \{Y\}), (\{Y\}, \{X\}), (\emptyset, \emptyset \cup \emptyset)\} \\ &\quad = \{(\emptyset, \{Y\}), (\{X\}, \emptyset), (\emptyset, \emptyset), (\{X\}, \{Y\}), (\{Y\}, \{X\}), (\emptyset, \emptyset)\} \\ &\quad = \{(\emptyset, \{Y\}), (\{X\}, \emptyset), (\emptyset, \emptyset), (\{X\}, \{Y\}), (\{Y\}, \{X\})\} \end{aligned} \tag{31}$$

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

$$C_r V = \bigcup_{(P,D) \in dep(r), D \subseteq V} P. \tag{32}$$

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