Evolutionary Inference of Attribute-based Access Control Policies

Eric Medvet[1], Alberto Bartoli[1], Barbara Carminati[2], and Elena Ferrari[2]

[1] Dip. di Ingegneria e Architettura, Università degli Studi di Trieste, Italy
[2] Dip. di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Italy

A Appendix: Genetic operators

The mutation operators are the following, given a parent rule $\rho = \langle e_U, e_R, O, c \rangle$—for the operators described using the placeholder $*$, the operator is actually applied with $* = U$ or $* = R$ with equal probability.

Attribute expression addition An $a_* \in A_*$ is randomly chosen such that $e_*(a_*) = \top$, then $e_*(a_*) := \{v\}$ with v randomly chosen in $V_*(a_*)$, if $a_* \in A_{*,\mathbb{1}}$, or $e_*(a_*) := \{s\}$ with s randomly chosen in $\text{Set}(V_*(a_*))$, if $a_* \in A_{*,\infty}$.
Attribute expression removal An $a_* \in A_*$ is randomly chosen such that $e_*(a_*) \neq \top$, then $e_*(a_*) := \top$.
Single-valued addition An $a_{*,\mathbb{1}} \in A_{*,\mathbb{1}}$ is randomly chosen such that $e_*(a_{*,\mathbb{1}}) \neq \top$, then $e_*(a_{*,\mathbb{1}}) := e_*(a_{*,\mathbb{1}}) \cup v$ with v randomly chosen in $V_*(a_{*,\mathbb{1}}) \setminus e_*(a_{*,\mathbb{1}})$.
Single-valued removal An $a_{*,\mathbb{1}} \in A_{*,\mathbb{1}}$ is randomly chosen such that $e_*(a_{*,\mathbb{1}}) \neq \top$, then $e_*(a_{*,\mathbb{1}}) := e_*(a_{*,\mathbb{1}}) \setminus v$ with v randomly chosen in $e_*(a_{*,\mathbb{1}})$; if $e_*(a_{*,\mathbb{1}})$ becomes empty, then $e_*(a_{*,\mathbb{1}}) := \top$.
Multi-valued addition An $a_{*,\infty} \in A_{*,\infty}$ is randomly chosen such that $e_*(a_{*,\infty}) \neq \top$ and a set $s \in e_*(a_{*,\infty})$ is randomly chosen, then $s := s \cup v$ with v randomly chosen in $V_*(a_{*,\infty}) \setminus s$.
Multi-valued removal An $a_{*,\infty} \in A_{*,\infty}$ is randomly chosen such that $e_*(a_{*,\infty}) \neq \top$ and a set $s \in e_*(a_{*,\infty})$ is randomly chosen, then $s := s \setminus v$ with v randomly chosen in s; if s becomes empty, then it is removed from $e_*(a_{*,\infty})$, if $e_*(a_{*,\infty})$ becomes empty, then $e_*(a_{*,\infty}) := \top$.
Constraint addition A pair $a_U, a_R \in A_U \times A_R$ is randomly chosen such that $c(a_U, a_R) = \top$ and $V_U(a_U) \cap V_R(a_R) \neq \emptyset$ (i.e., a_U and a_R have some values in common), then $c(a_U, a_R) := \neg\top$.
Constraint removal A pair $a_U, a_R \in A_U \times A_R$ is randomly chosen such that $c(a_U, a_R) = \neg\top$, then $c(a_U, a_R) := \top$.
Operation addition An operation $o \in \mathcal{O}$ is randomly chosen such that $o \notin O$, then $O := O \cup \{o\}$.
Operation removal An operation $o \in \mathcal{O}$ is randomly chosen such that $o \in O$, then $O := O \setminus \{o\}$.

The crossover operators are the following, given two parent rules $\rho_1 = \langle e_{U,1}, e_{R,1}, O_1, c_1 \rangle$ and $\rho_2 = \langle e_{U,2}, e_{R,2}, O_2, c_2 \rangle$. The child rule is ρ_1 after the actual application of the operator.

Attribute expression donation An $a_* \in A_*$ is randomly chosen such that $e_{*,1}(a_*) = \top \wedge e_{*,2}(a_*) \neq \top$, then $e_{*,1}(a_*) := e_{*,2}(a_*)$.

Single-valued donation An $a_{*,\mathbb{1}} \in A_{*,\mathbb{1}}$ is randomly chosen such that $e_{*,1}(a_{*,\mathbb{1}}) \neq \top \wedge e_{*,2}(a_{*,\mathbb{1}}) \neq \top$, then $e_{*,1}(a_{*,\mathbb{1}}) := e_{*,1}(a_{*,\mathbb{1}}) \cup v$ with v randomly chosen in $e_{*,2}(a_{*,\mathbb{1}})$.

Multi-valued donation An $a_{*,\infty} \in A_{*,\infty}$ is randomly chosen such that $e_{*,1}(a_{*,\infty}) \neq \top \wedge e_{*,2}(a_{*,\infty}) \neq \top$ and two sets $s_1 \in e_{*,1}(a_{*,\infty}), s_2 \in e_{*,2}(a_{*,\infty})$ are randomly chosen, then $s_1 \cup v$ with v randomly chosen in s_2.

Constraint donation A pair $a_U, a_R \in A_U \times A_R$ is randomly chosen such that $c_1(a_U, a_R) = \top \wedge c_2(a_U, a_R) = \neg\top$, then $c_1(a_U, a_R) := c_2(a_U, a_R)$.

Operation donation An operation $o \in O_2$ is randomly chosen such that $o \notin O_1$, then $O_1 := O_1 \cup \{o\}$.

When a genetic operator cannot be applied (e.g., when $O_1 = \mathcal{O}$ for the operation donation crossover operator), the generated rule is set equal to the (first) parent.