ProProv

ProProv

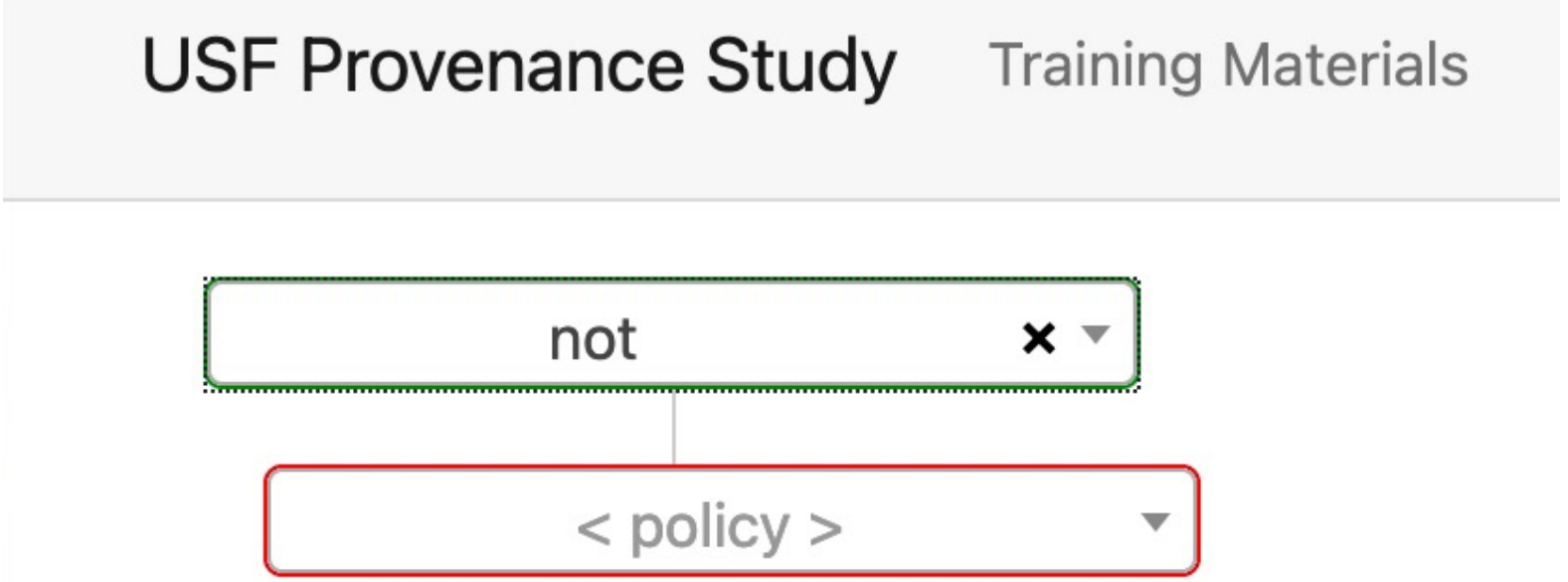
- Language and system for specifying provenance policies over graphs representing the provenance of secure computations
- Policies are expressed as logical statements, or predicates, over provenance graphs
- Contains a graphical user interface to specify and test policies

ProProv Syntax

- Provenance node types t: accountAgent, nodeAgent, agent, dataEntity, contractEntity, keyEntity, entity, or activity
- Edges e: wasDerivedFrom, wasAttributedTo, wasGeneratedBy, used, actedOnBehalfOf, or wasAssociatedWith
- Variables $x \in Strings$
- Nodes $n \in Strings$
- Policies p: $!\,p$, $p_1 \wedge p_2$, $p_1 \vee p_2$, $forall\ x{:}\,t.\,p$, $exists\ x{:}\,t.\,p$, $p_1 \Rightarrow p_2$, $e(n_1, n_2)$

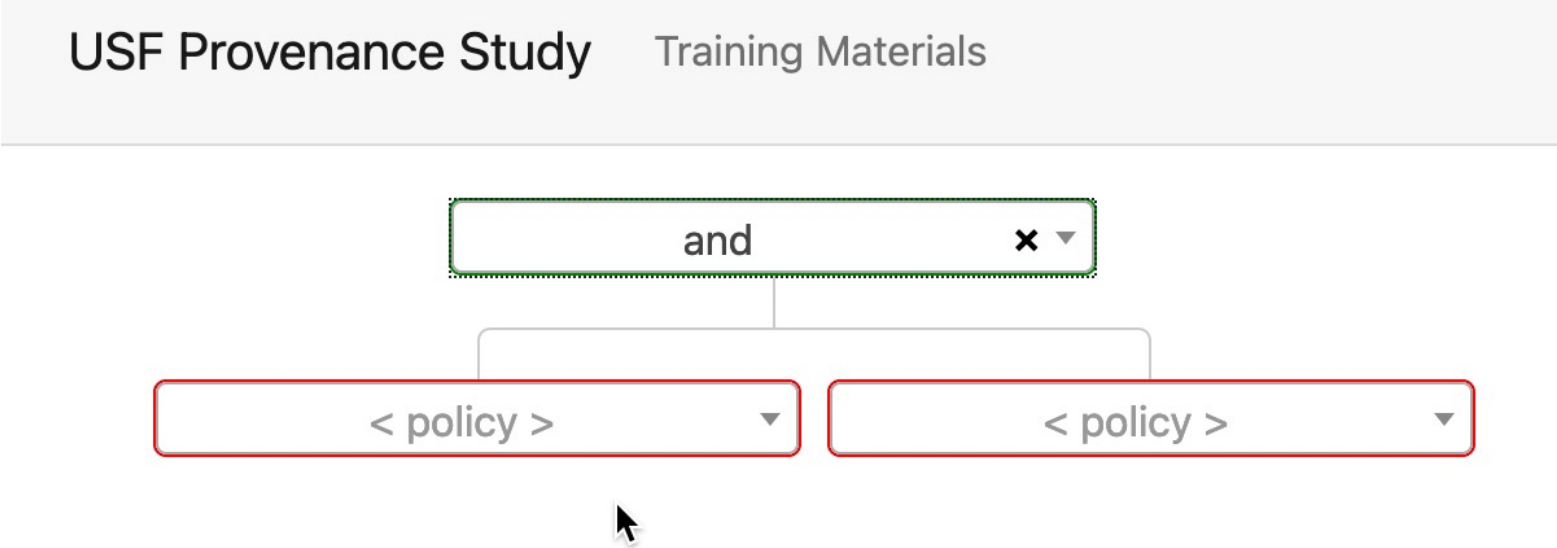
Negation policy ($!\,p$)

- Asserts that policy p is not true



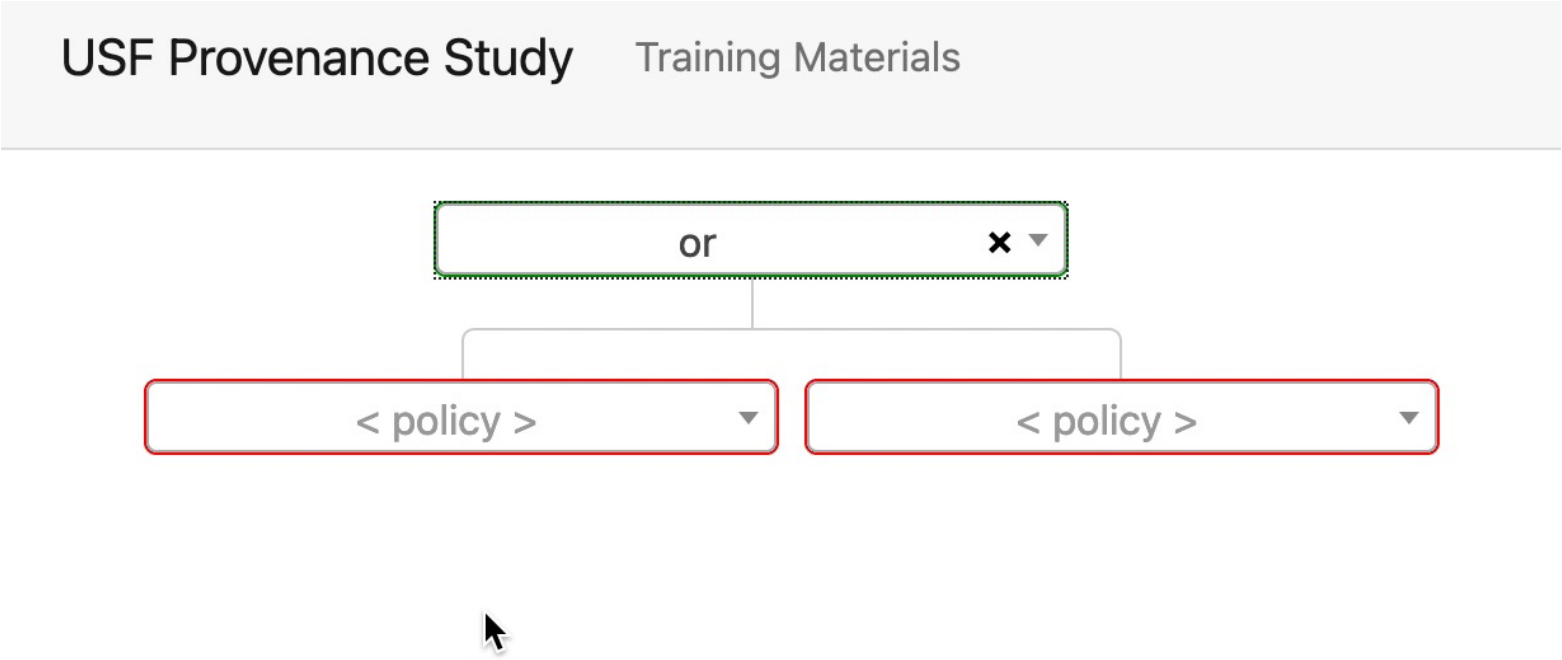
Conjunction Policy ($p_1 \wedge p_2$)

- Also called AND
- Asserts that both policy p_1 and policy p_2 are true



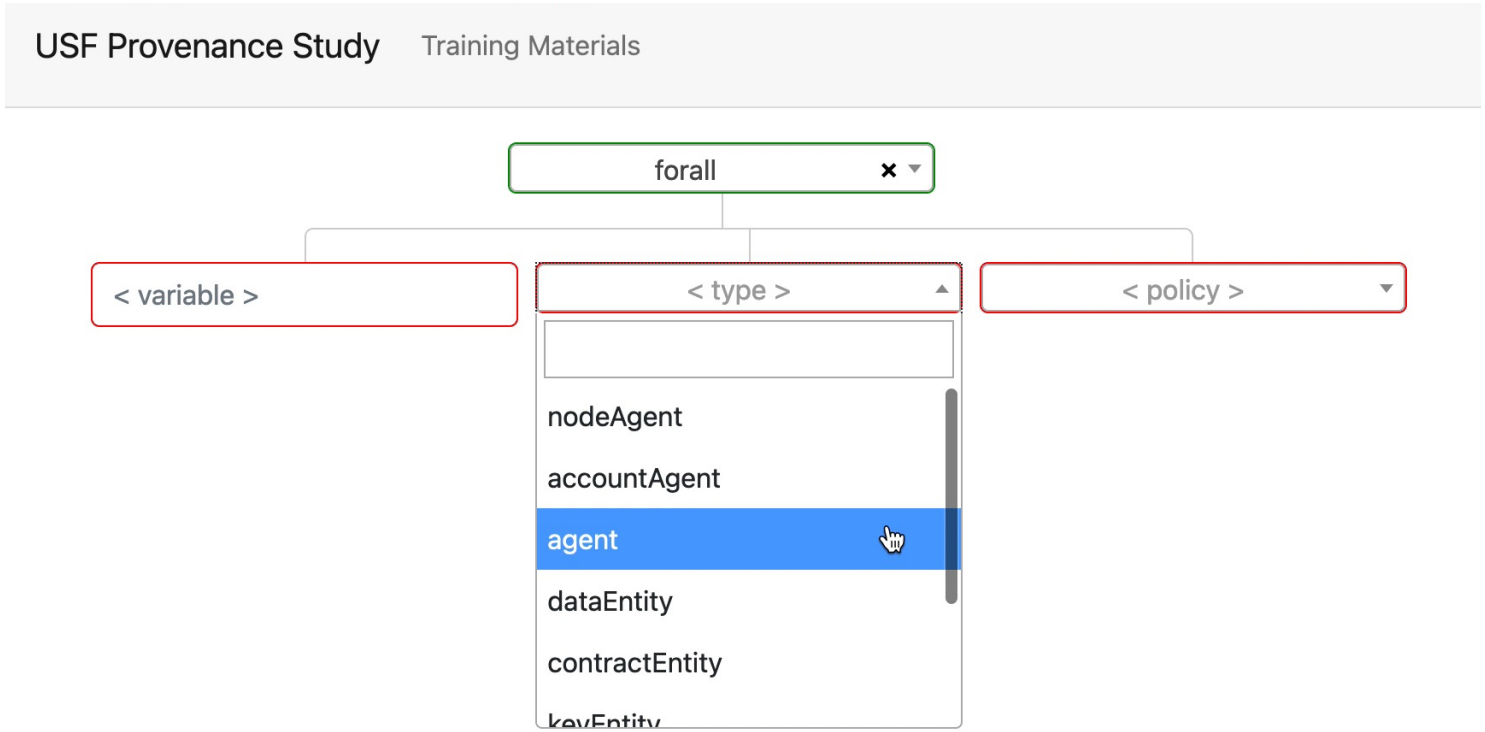
Disjunction Policy (p_1 V p_2)

- Also called OR
- Asserts that either policy p_1 is true, policy p_2 is true , or both p_1 and p_2 are true



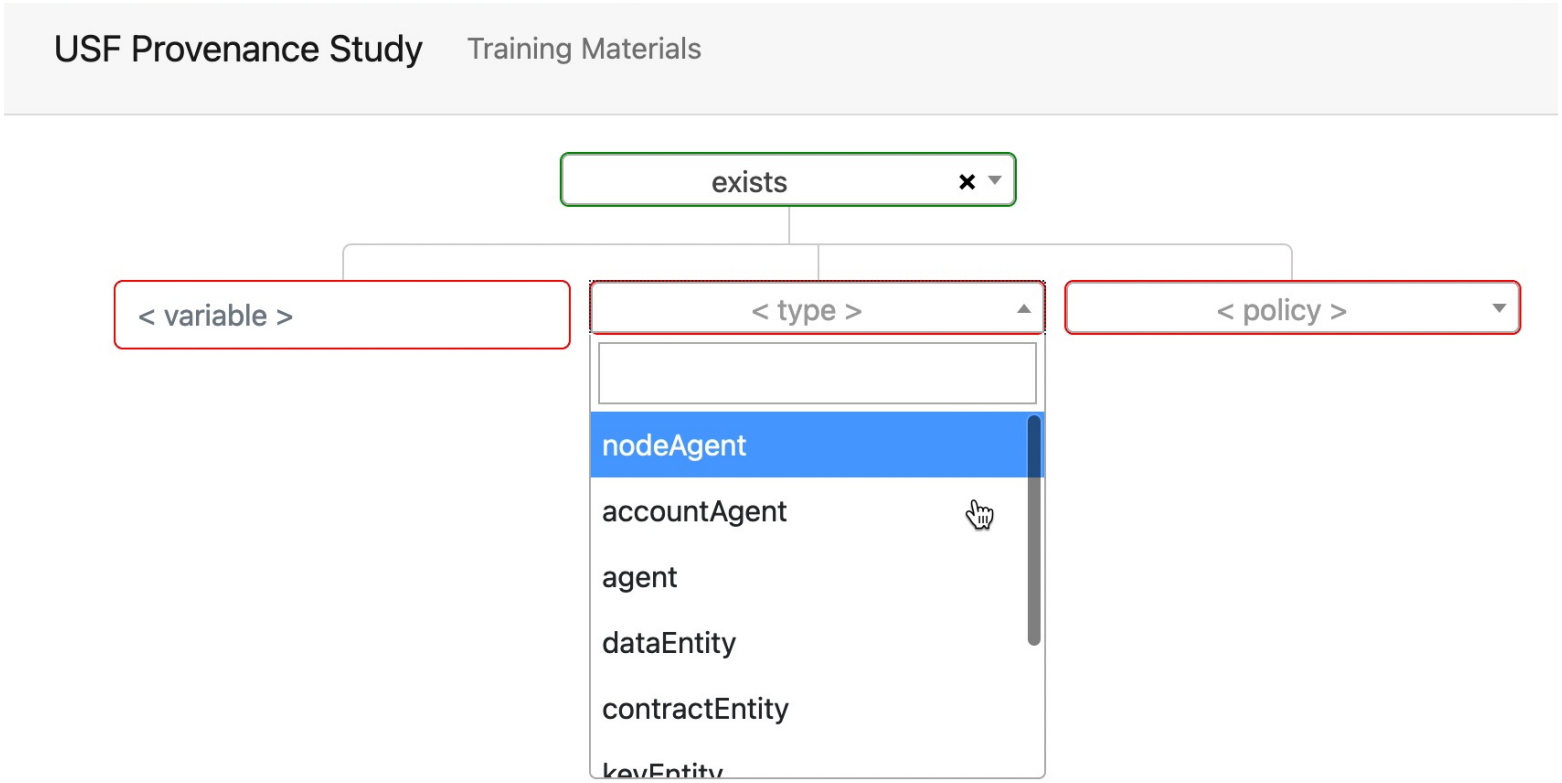
Universal Policy ($forall\ x{:}\ t.\ p$)

- Asserts that policy p is true for every provenance node of type t. Variable x can be used in p



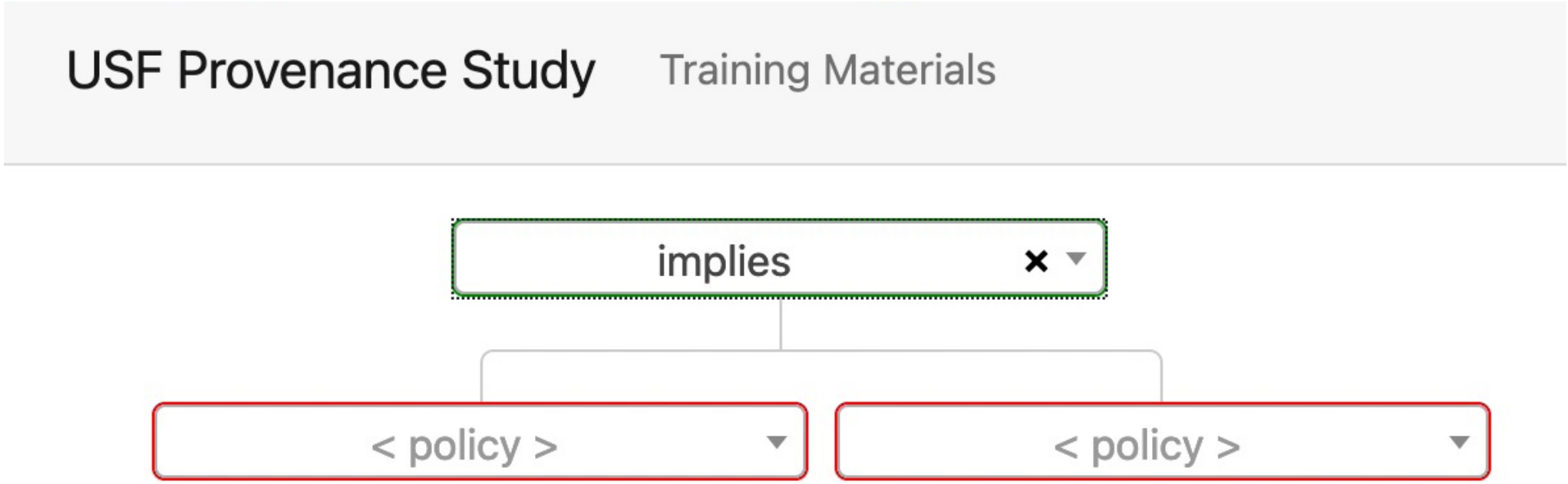
Existential Policy ($exist\ x\colon t.\ p$)

- Asserts that policy p is true for some provenance node of type t. Variable x can be used in p



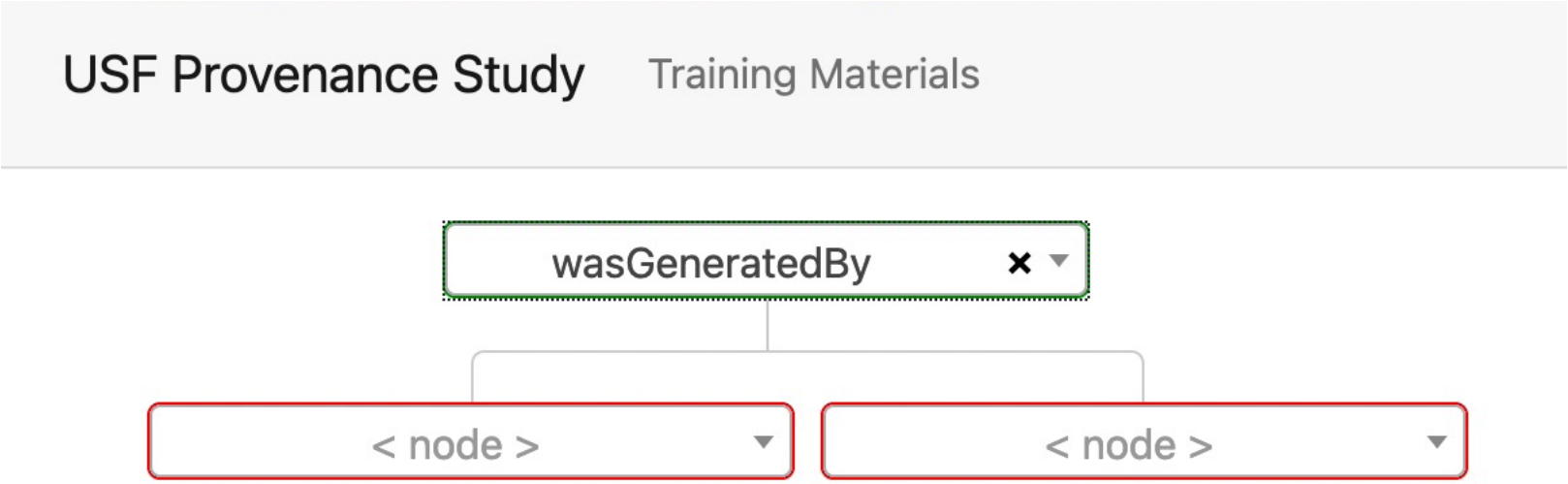
Implication policy ($p_1 \Rightarrow p_2$)

- Asserts that if policy p_1 is true then policy p_2 is true



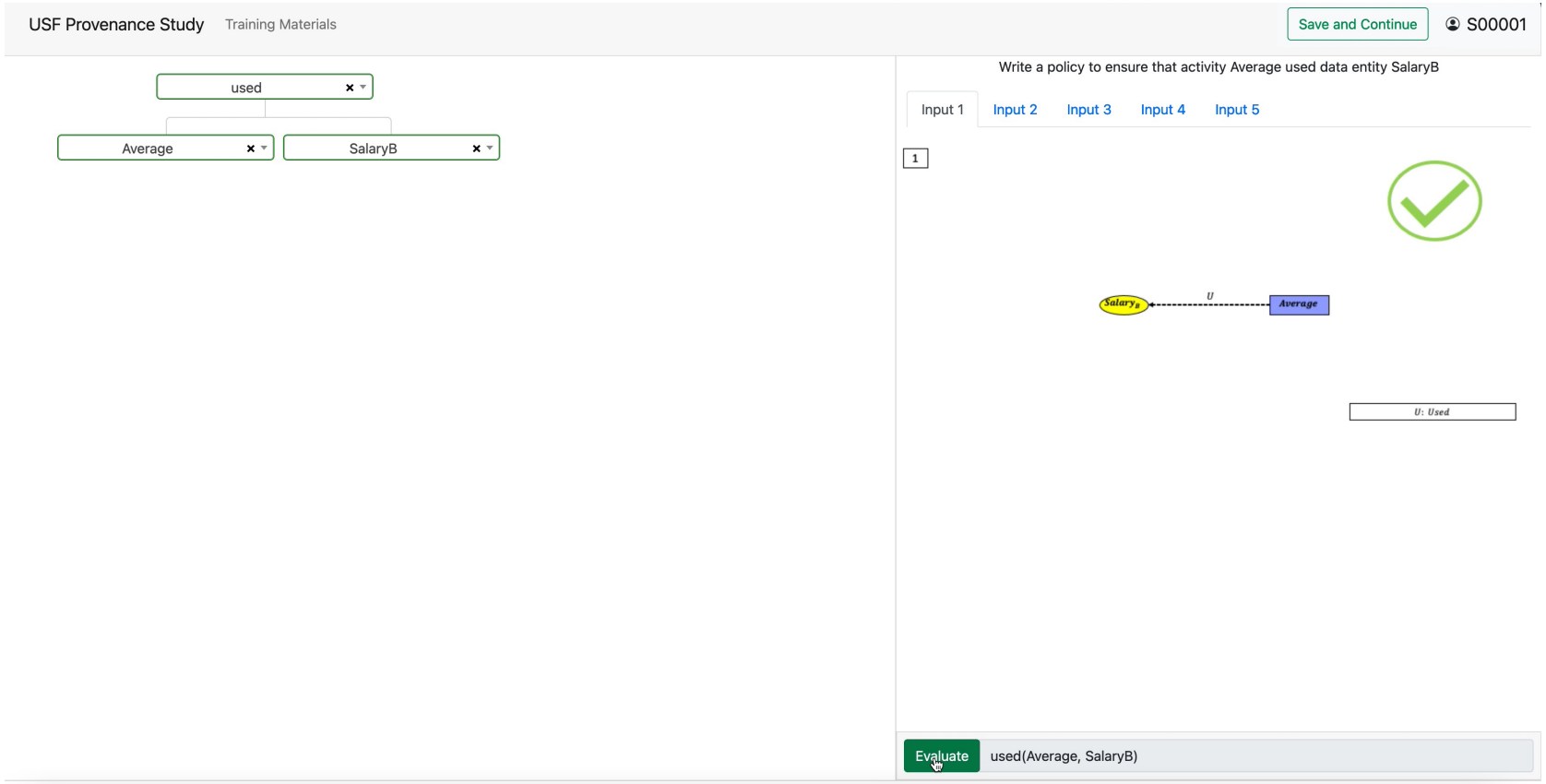
Edge policy ($e(n_1, n_2)$)

- e: wasAttributedTo, wasDerivedFrom, wasGeneratedBy, used, actedOnBehalfOf, or wasAssociatedWith
- Asserts that there is an edge with label e between node n_1 and node n_2



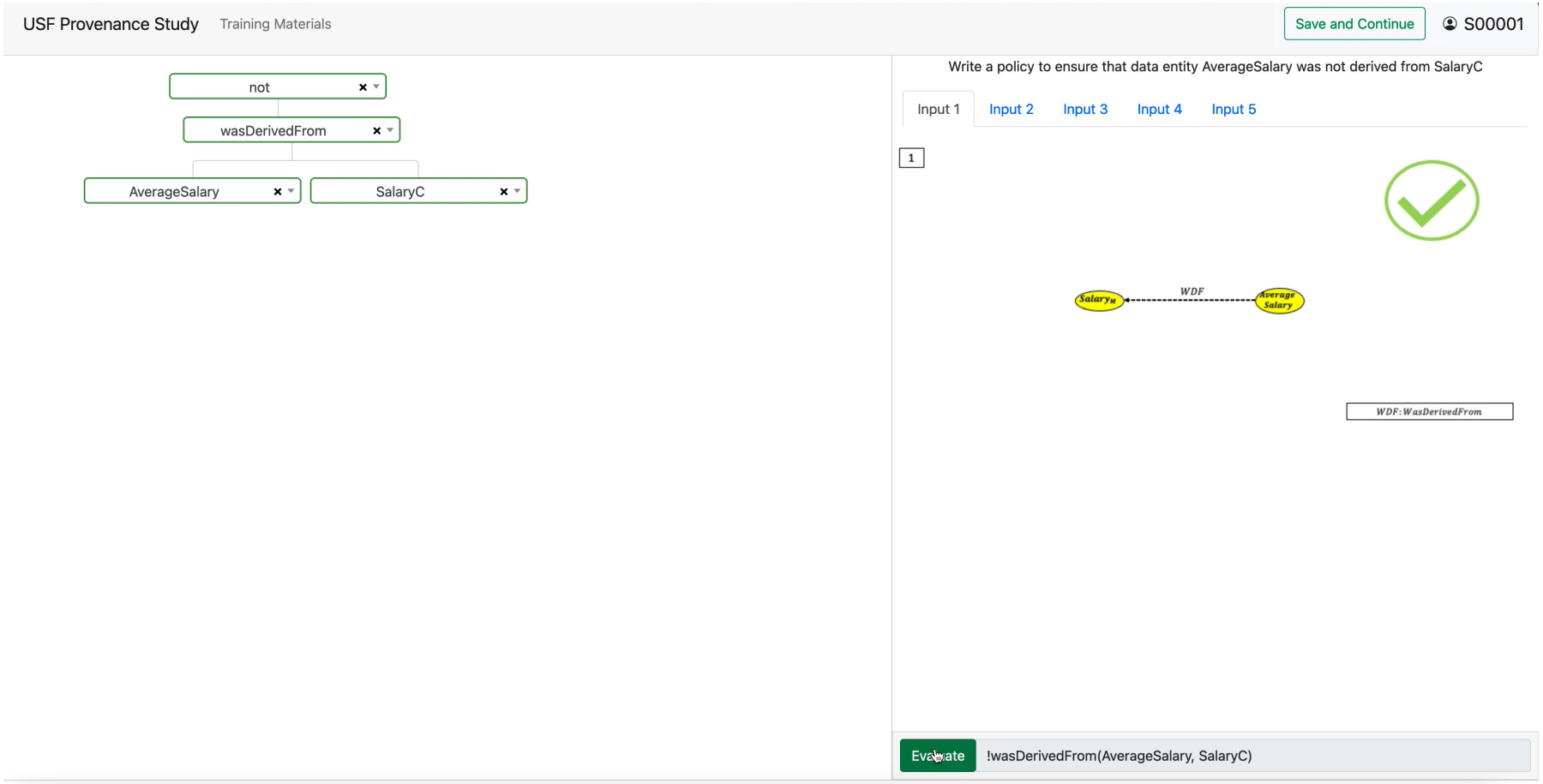
Provenance policy examples

- ProProv is case insensitive, *Average* ≡ *average*
- Write a policy to ensure that activity *Average* used data entity *SalaryB*



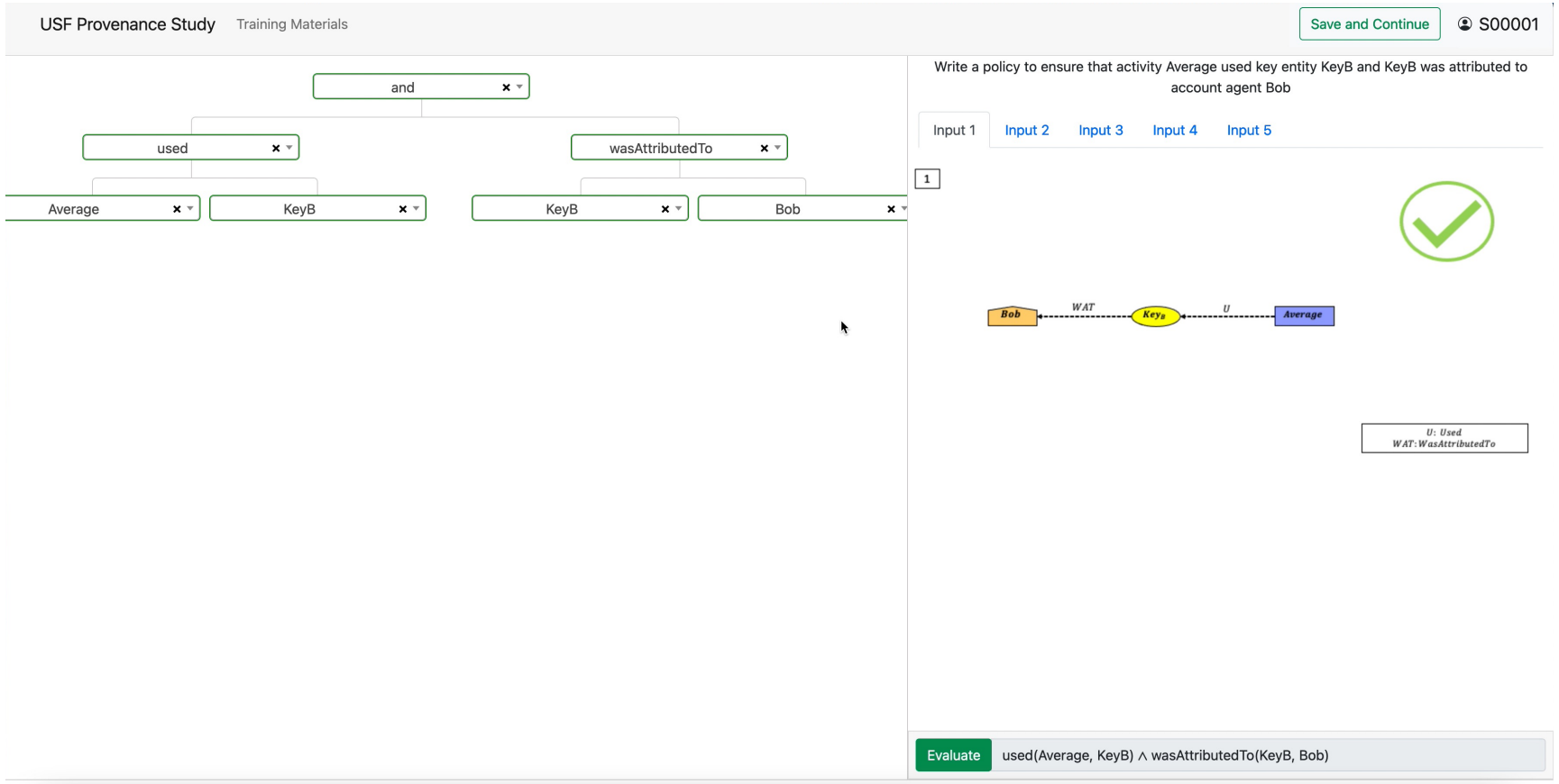
Provenance policy examples

- Write a policy to ensure that data entity *AverageSalary* was not derived from *SalaryC*



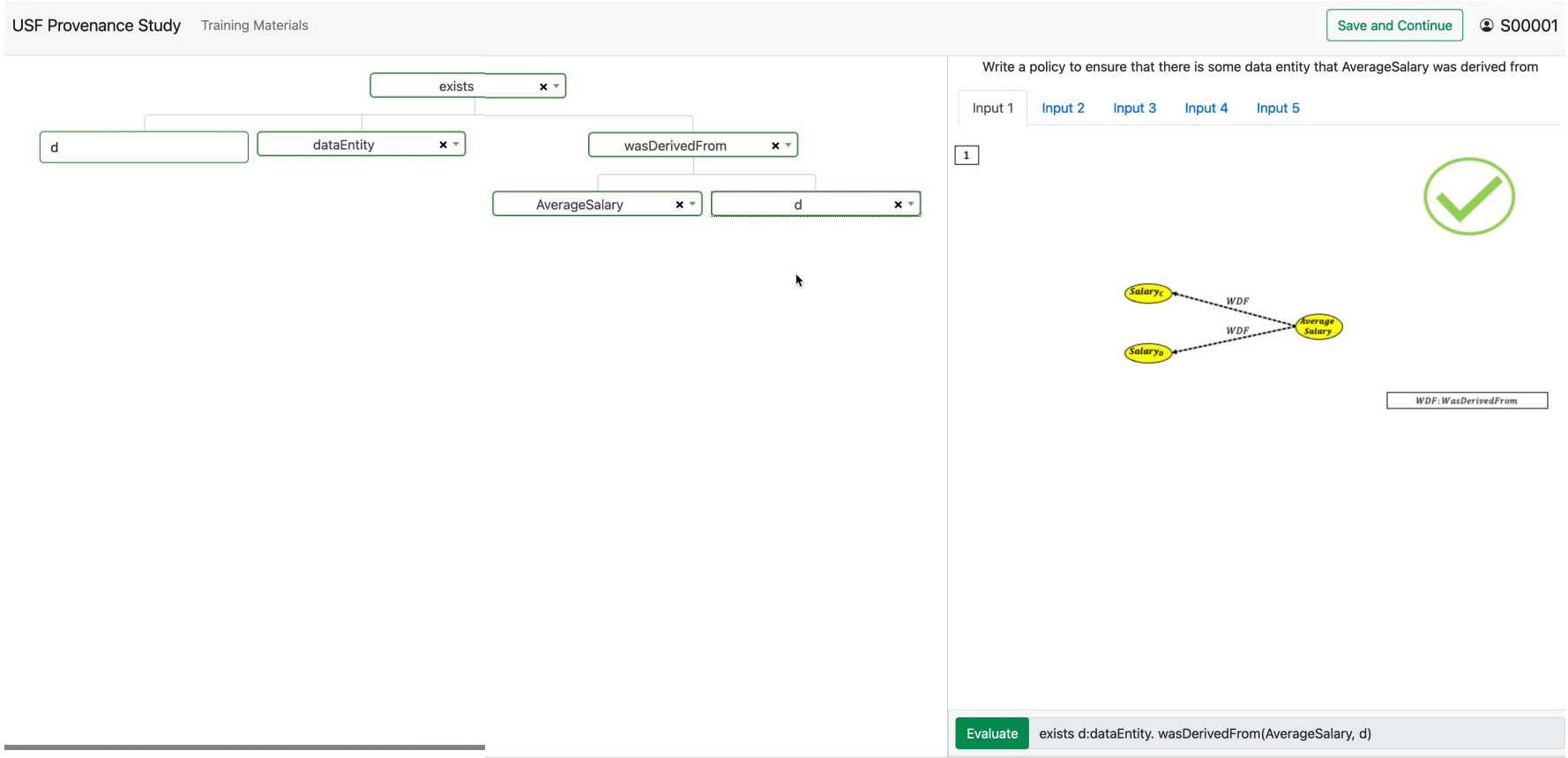
Provenance policy examples

- Write a policy to ensure that activity *Average* used key entity *KeyB* and *KeyB* was attributed to account agent *Bob*



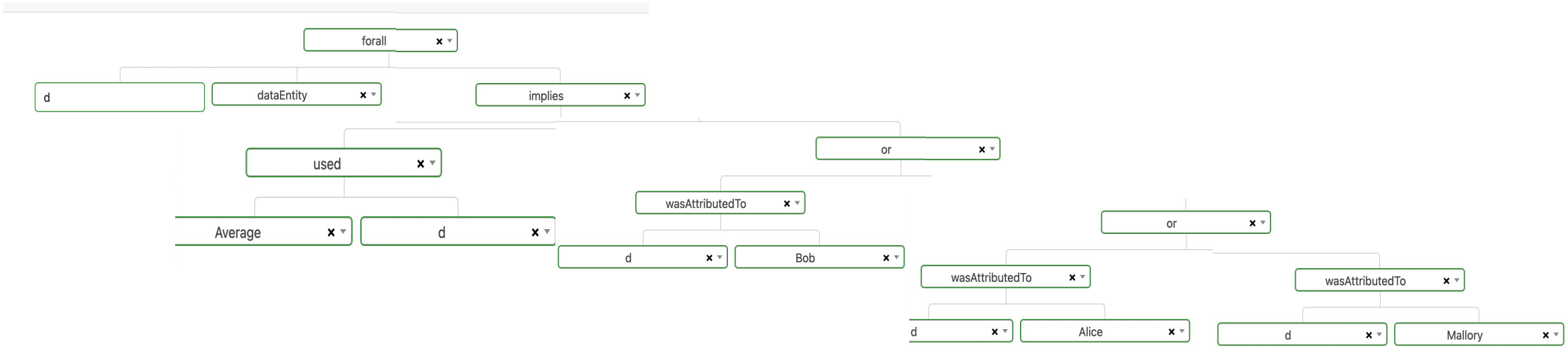
Provenance policy examples

- Write a policy to ensure that there is some data entity that *AverageSalary* was derived from



Provenance policy examples

- Write a policy to ensure for every data entity, if activity *Average* used the data entity, then the data entity was attributed to either Bob, Alice, or Mallory



ProProv Incorrect Answers