The tool invFinder works in a semi-proving and semi-searching workflow. In an iterating step, it tries to prove some consistent relation exists between an invariant and a rule, and automatically generates a new auxiliary invariant if there is no such an invariant in the current invariant set, and records the corresponding causal relation information between the current rule and invariant. This workflow is not finished until no new invariants is created. The core part of the searching algorithm of the invFinder is given roughly as follows:

```
1  let findInvsFromRule chk choose tautChk isNew pRule paras inv newInvs invs casRel=
2  let rule=ruleApp pRule paras in
3  val (g ▷ S)=rule in
5  let inv'=preCond S inv in
7  inv=inv'=>
8  let relItem=(pRule, paras, inv,invHoldForRule2 inv r) in (newInvs, relItem:casRel)
9   | tautChk (g ⟶inv')
10   =>let relItem=(pRule, paras, inv, invHoldForRule1 inv r) in (newInvs, relItem:casRel)
11   |let newInv= choose chk inv' g then
12   let relItem=(pRule, paras, inv, invHoldForRule3 inv newInv ) in
13     ((isNew newInv (newInvs@invs))=> (newInvs@[normalize newInv], relItem :  casRel)
14     |(newInvs, relItem:casRel)
15   | error no new invariant;
```

The above function findInvsFromRule tries to find new invariants and construct the causal relation between the rule instance $r = pRule\ para$ and the invariant inv. The statement $cond => te|fe$ is an abbreviation of the if-then-else expression that if $cond$ is true then te else fe. Parameters $newInvs$, $invs$, and $casRel$ are new invariants, invariants, and all the causal relations constructed up to now, the above oracle functions are also passed as parameters. Causal relations are still not checked between the ones in newInvs and rules. The returned result is updated new invariants, and causal relations.

1. After computing the pre-condition $inv' = preCond\ S\ inv$ at line 5, invFinder performs case analysis on inv0: if $inv = inv'$, then no change made to inv by statement S, the new causal relation item marked with tag $invHoldForRule2$ is recorded between r and inv, but at this moment there are no new invariants to be added; for instance, let $pRule = crit$, $paras = (3), inv = mutualInv\ 1\ 2, inv' = preCond\ S\ inv = inv)$, thus only a new relation item $(crit, (3), inv, invHoldForRule2)$ will be added.

2. Secondly, if $tautChk\ g \rightarrow inv'$is true, then the new causal relation item marked with tag $invHoldForRule1$ is recorded between r and inv, but at this moment there are no new invariants to be added either; for instance, let$pRule = crit, paras = (3), inv = aux\ 1, inv' = preCond\ S\ inv = \neg(false = true \wedge n[1] = C)$, obviously, $g \rightarrow inv'$ holds forever, thus a new relation item $(crit, (3), inv, invHoldForRule2)$ will be added.

3. Thirdly, the function choose at line 12 chooses the weakest precondition $newInv$ from the conjuncts of $g \wedge \neg inv0$. choose need call chk to guarantee that the constructed formula is an invariant in the given reference model; and function $isNew$ is used to check whether the invariant is new. If yes, the invariant will be normalized, then be added into newInvs, and the new

causal relation item marked with tag invHoldForRule3 inv' will be added into the causal relations. Here, the meaning of the word "new" is modulo to the symmetry relation. For instance, mutualInv 1 2 is equivalent to mutualInv 2 1 in symmetry view. The normalize function normalize the numbering order of the use of parameters in the invariant inv. The result formula should be a normal form, whose parameters always start from 1, and increase one by one if there are more parameters. Namely, $\neg(x = true \wedge n[1] = C)$ is normalized, but $\neg(x = true \wedge n[2] = C$ not. Let $invs = newInvs = \emptyset, pRule = crit, paras = (1), inv = mutualInv\ 1\ 2, inv' = preCond\ S\ inv = \neg(true = true \wedge n[2] = C)$, from two conjunction items n[1]=T and x=true in g, the choose oracle selects the second, and combines it with inv0, then constructs a new invariant $\neg(x = true \wedge n[2] = C)$ after necessary simplication by removing C=C and normalization, then the new invariant $\neg(x = true \wedge n[1] = C)$ will be added, and a new relation item $(crit, (1), inv, invHoldForRule3\ newInv)$ will be added.