
define: HandleInstForCollection(*inst, alias_graph, lock_set, key_fields*)

```
1:  alias_graph' := UpdateAliasGraph(alias_graph, inst);
2:  lock_set' := UpdateLockSet(inst, lock_set, alias_graph');
3:  field_access->var_field := NULL;
4:  field_access->lock_field := NULL;
5:  if inst is a write then
6:    field_access->access_type = write;
7:  else
8:    field_access->access_type = read;
9:  else
10:    return <alias_graph', lock_set', NULL>;
11:  end if
12:  var := GetOperand(inst);
13:  var_node := GetAliasNode(var, alias_graph');
14:  // Finding whether the accessed variable is protected by a lock.
15:  foreach lock_node in lock_set' do
16:    <var_field, lock_field> := GetProtectedFieldAccess(
17:      var_node, lock_node, alias_graph');
18:    if var_field is not NULL then
19:      field_access->var_field := var_field;
20:      field_access->lock_field := lock_field;
21:      return <alias_graph', lock_set', field_access>;
22:    end if
23:  end foreach
24:  // Finding whether the accessed var exist in a key field.
25:  foreach var_field in key_fields do
26:    if var exists in the data structure field var_field then
27:      field_access->var_field := var_field;
28:      field_access->lock_field := NULL;
29:      return <alias_graph', lock_set', field_access>;
30:    end if
31:  end foreach
32:  return <alias_graph', lock_set', NULL>;
```

define: CollectFieldAccess ()

```
33:  key_fields := ExtractKeyField();
34:  field_access_rec :=  $\emptyset$ ;
35:  foreach func in OS code without a caller function do
36:    foreach code_path in GetCodePath(func) do
37:      alias_graph :=  $\emptyset$ ;
38:      lock_set :=  $\emptyset$ ;
39:      foreach inst in GetInstructions(code_path) do
40:        <alias_graph, lock_set, field_access> :=
41:          HandleInstForCollection(
42:            inst, alias_graph, lock_set, key_fields);
43:        if field_access is not NULL then
44:          insert <code_path, field_access> into field_access_rec
45:        end foreach
46:      end foreach
47:    end foreach
48:  return field_access_rec;
```
