|  |  |
| --- | --- |
| **define:** HandleInstForCollection(*inst*, *alias\_graph*, *lock\_set*, *key\_fields*) | |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26:  27:  28:  29:  30:  31:  32: | *alias\_graph* := UpdateAliasGraph(*alias\_graph*, *inst*);  *lock\_set* := UpdateLockSet(*inst*, *lock\_set*, *alias\_graph*);  *field\_access*->*var\_field* := NULL;  *field\_access*->*lock\_field* := NULL;  **if** *inst* is a write **then**  *field\_access*->*access\_type* = write;  **else**  *field\_access*->*access\_type* = read;  **else**  **return** <*alias\_graph*, *lock\_set*, *NULL*>;  **end if**  *var* := GetOperand(*inst*);  *var\_node* := GetAliasNode(*var*, *alias\_graph*);  *// Finding whether the accessed variable is protected by a lock.*  **foreach** *lock\_node* in *lock\_set* **do**  <*var\_field*, *lock\_field*> := GetProtectedFieldAccess(  *var\_node*, *lock\_node*, *alias\_graph*);  **if** *var\_field* is not NULL **then**  *field\_access*->*var\_field* := *var\_field*;  *field\_access*->*lock\_field* := *lock\_field*;  **return** <*alias\_graph*, *lock\_set*, *field\_access*>;  **end if**  **end foreach**  *// Finding whether the accessed var exist in a key field.*  **foreach** *var\_field* in *key\_fields* **do**  **if** *var* exists in the data structure field *var\_field* **then**  *field\_access*->*var\_field* := *var\_field*;  *field\_access*->*lock\_field* := NULL;  **return** <*alias\_graph*, *lock\_set*, *field\_access*>;  **end if**  **end foreach**  **return** <*alias\_graph*, *lock\_set*, *NULL*>; |

|  |  |
| --- | --- |
| **define:** CollectFieldAccess () | |
| 33:  34:  35:  36:  37:  38:  39:  40:  41:  42:  43:  44:  45:  46:  47:  48: | *key\_fields* := ExtractKeyField();  *field\_access\_rec* := ;  **foreach** *func* in OS code without a caller function **do**  **foreach** *code\_path* in GetCodePath(*func*) **do**  *alias\_graph* := ;  *lock\_set* := ;  **foreach** *inst* in GetInstructions(*code\_path*) **do**  <*alias\_graph*, *lock\_set*, *field\_access*> :=  HandleInstForCollection(  *inst*, *alias\_graph*, *lock\_set*, *key\_fields*);  **if** *field\_access* is not NULL **then**  insert <*code\_path*, *field\_access*> into *field\_access\_rec*  **end foreach**  **end foreach**  **end foreach**  **return** *field\_access\_rec*; |