
Algorithm Fully symbolic memory: naive implementation

Immutable objects:

M_c	$:= \{(e_c, v)\}$
e_c	$:=$ an expression over concrete values
M_s	$:= \{(e, v)\}$
e	$:=$ an expression over symbols and concrete values
v	$:=$ a 1-byte expression over symbols and concrete values
V	$:=$ ordered set of v
π	$:=$ set of assumptions
$equiv(e, \tilde{e}, \pi)$	$:= (e \neq \tilde{e} \wedge \pi) == UNSAT$
$disjoint(e, \tilde{e}, \pi)$	$:= (e = \tilde{e} \wedge \pi) == UNSAT$
$intersect(e, \tilde{e}, \pi)$	$:= (e = \tilde{e} \wedge \pi) == SAT$

```
1: function STORE( $e, v, size$ ):
2:   for  $k = 0$  to  $size - 1$  do
3:     _STORE( $e + k, v_k$ )
4:   end for
5: end function
```

```
1: function _STORE( $e, V$ ):
2:    $min_e = \min(e)$ 
3:    $max_e = \max(e)$ 
4:    $flag = false$ 
5:    $constant = false$ 
6:    $M_c' \leftarrow M_c$ 
7:    $M_s' \leftarrow M_s$ 
8:   if  $min_e == max_e$  then
9:      $constant = true$ 
10:     $M_c' \leftarrow M_c'|_{e \mapsto v}$ 
11:  end if
12:  for  $(\tilde{e}, \tilde{v}) \in M_s$  do
13:    if  $disjoint(\tilde{e}, e, \pi)$  then
14:      continue
15:    else if  $equiv(\tilde{e}, e, \pi)$  then
16:      if  $constant$  then
17:         $M_s' \leftarrow M_s' - (\tilde{e}, v)$ 
18:      else
19:         $M_s' \leftarrow M_s'|_{\tilde{e} \mapsto v}$ 
20:      end if
21:       $flag = true$ 
22:    else
23:       $M_s' \leftarrow M_s'|_{\tilde{e} \mapsto ite(\tilde{e}=e, v, \tilde{v})}$ 
24:    end if
25:  end for
26:  if  $\neg flag \wedge \neg constant$  then
27:     $M_s' \leftarrow M_s'|_{e \mapsto v}$ 
28:  end if
29:   $M_c \leftarrow M_c'$ 
30:   $M_s \leftarrow M_s'$ 
31: end function
```

```

1: function LOAD( $e$ ,  $size$ ):
2:    $V = \langle \rangle$ 
3:   for  $k = 0$  to  $size - 1$  do
4:      $v_k = \perp$ LOAD( $e + k$ )
5:      $V = V \cdot v_k$ 
6:   end for
7:   return  $V$ 
8: end function

```

```

1: function  $\perp$ LOAD( $e$ ):
2:    $v = \perp$ 
3:    $min_e = min(e)$ 
4:    $max_e = max(e)$ 
5:   if  $min_e == max_e$  then
6:     if  $(min_e, \tilde{v}) \in M_c$  then
7:        $v = \tilde{v}$ 
8:     end if
9:   else
10:    for  $e_c = min_e$  to  $max_e$  do
11:      if  $(e_c, \tilde{v}) \in M_c$  then
12:         $v = ite(e_c = e, \tilde{v}, v)$ 
13:      end if
14:    end for
15:   end if
16:   for  $(\tilde{e}, \tilde{v}) \in M_s$  do
17:     if  $intersect(\tilde{e}, e, \pi)$  then
18:        $v = ite(\tilde{e} = e, \tilde{v}, v)$ 
19:     end if
20:   end for
21:   return  $v$ 
22: end function

```
