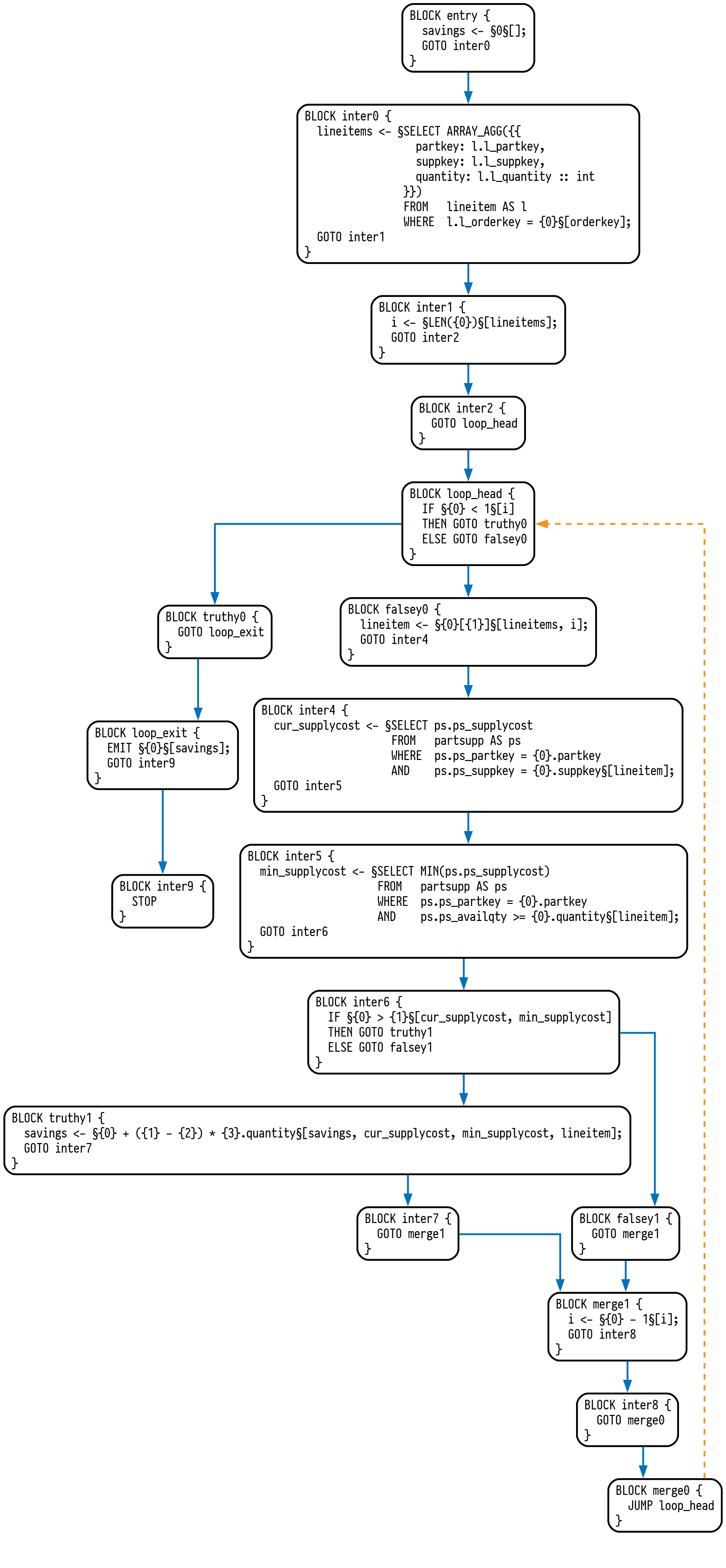
BLOCK entry {
savings <- §0§[];
GOTO inter0
}
BLOCK inter0 {
lineitems <- §SELECT ARRAY_AGG({{
partkey: l.l_partkey,
suppkey: l.l_suppkey,
quantity: l.l_quantity :: int
}})
FROM lineitem AS l
WHERE l.l_orderkey = {0}§[orderkey];
GOTO inter1
}
BLOCK inter1 {
i <- §LEN({0})§[lineitems];
GOTO inter2
}
BLOCK inter2 {
GOTO loop_head
}
BLOCK loop_head {
IF §{0} < 1§[i]
THEN GOTO truthy0
ELSE GOTO falsey0
}
BLOCK truthy0 {
GOTO loop_exit
}
BLOCK falsey0 {
lineitem <- §{0}[{1}]§[lineitems, i];
GOTO inter4
}
BLOCK loop_exit {
EMIT §{0}§[savings];
GOTO inter9
}
BLOCK inter4 {
cur_supplycost <- §SELECT ps.ps_supplycost
FROM partsupp AS ps
WHERE ps.ps_partkey = {0}.partkey
AND ps.ps_suppkey = {0}.suppkey§[lineitem];
GOTO inter5
}
BLOCK inter9 {
STOP
}
BLOCK inter5 {
min_supplycost <- §SELECT MIN(ps.ps_supplycost)
FROM partsupp AS ps
WHERE ps.ps_partkey = {0}.partkey
AND ps.ps_availqty >= {0}.quantity§[lineitem];
GOTO inter6
}
BLOCK inter6 {
IF §{0} > {1}§[cur_supplycost, min_supplycost]
THEN GOTO truthy1
ELSE GOTO falsey1
}
BLOCK truthy1 {
savings <- §{0} + ({1} - {2}) * {3}.quantity§[savings, cur_supplycost, min_supplycost, lineitem];
GOTO inter7
}
BLOCK inter7 {
GOTO merge1
}
BLOCK falsey1 {
GOTO merge1
}
BLOCK merge1 {
i <- §{0} - 1§[i];
GOTO inter8
}
BLOCK inter8 {
GOTO merge0
}
BLOCK merge0 {
JUMP loop_head
}