```
BLOCK entry {
             schedule_start <- §SELECT o.o_orderdate</pre>
                                 FROM orders AS o
                                 WHERE o.o_orderkey = {0}\s[orderkey];
             GOTO inter0
                       BLOCK inter0 {
                         IF §{0} IS NULL§[schedule_start]
                          THEN STOP
                          ELSE GOTO falsey0
   BLOCK falsey0 {
     details <- §SELECT {{items: COUNT(*), last_shipdate: MAX(l.l_shipdate)}}</pre>
                 FROM lineitem AS l
                 WHERE 1.1_orderkey = {0}\s[orderkey];
     busy <- §ARRAY[] :: struct(start date, finish date)[]§[];</pre>
     priority <- §1§[];</pre>
     GOTO inter1
                 BLOCK inter1 {
                   schedule_end <- §{0}.last_shipdate§[details];</pre>
                   IF §{0} > {1}.items§[priority, details]
                   THEN STOP
                   ELSE GOTO falsey1
BLOCK falsey1 {
 lineitem <- §SELECT l.lineitem</pre>
                      (SELECT ROW_NUMBER() OVER (ORDER BY p.p_retailprice),
                              {{l_linenumber: l.l_linenumber,
                               l_shipdate: l.l_shipdate,
                               l_quantity: l.l_quantity}}
                       FROM lineitem AS l, part AS p
                       WHERE l.l_orderkey = {0}
                       AND l.l_partkey = p.p_partkey) AS l(priority, lineitem)
               WHERE l.priority = {1}\s[orderkey, priority];
  GOTO inter5
                          BLOCK inter5 {
                            item_end <- §LEAST({0}.l_shipdate, {1})§[lineitem, schedule_end];</pre>
                            GOTO inter6
                           BLOCK inter6 {
                             item_start <- §{0} - {1}.l_quantity :: int§[item_end, lineitem];</pre>
                             GOTO inner_loop_head
                                  BLOCK inner_loop_head {
                                    new_end <- §SELECT slot.start</pre>
                                                 FROM unnest({0}) AS _(slot)
                                                 WHERE slot.start < {1}</pre>
                                                 AND {2} < slot.finish
                                                 ORDER BY (slot.start, slot.finish)
                                                 LIMIT 1§[busy, item_end, item_start];
                                    GOTO inter8
                                                       BLOCK inter8 {
                                                         IF §{0} >= {1} AND {2} IS NOT NULL§[item_start, schedule_start, new_end]
                                                         THEN GOTO truthy2
                                                         ELSE IF §{0} >= {1}§[item_start, schedule_start]
                                                              THEN GOTO truthy3
                                                              ELSE GOTO merge3
                                                                                                 BLOCK truthy3 {
                                                        BLOCK truthy2 {
                                                                                                   busy <- §list_append({0}, ({1}, {2}))§[busy, item_start, item_end];</pre>
                                                          item_end <- §{0}§[new_end];
                                                                                                   EMIT §({0}.l_linenumber, {1})§[lineitem, item_start];
                                                          GOTO inter9
                                                                                                   GOTO merge3
                           BLOCK inter9 {
                                                                                                    BLOCK merge3 {
                                                                                                      priority <- §{0} + 1§[priority];</pre>
                            item_start <- §{0} - {1}.l_quantity :: int§[item_end, lineitem];</pre>
                                                                                                      JUMP outer_loop_head
                            JUMP inner_loop_head
                                                BLOCK outer_loop_head {
                                                  IF §{0} > {1}.items§[priority, details]
                                                  THEN STOP
                                                  ELSE GOTO falsey1
```