

The Packing Algorithm in VeGen

1 Pseudo Code in The Paper

Algorithm 1: Find the set of (non-load) packs that produce a given vector operand x . Load packs are found separately by enumeration.

Input :

x : The vector operand that we need to produce
 M : The match table, which contains the mapping $\langle \text{live-out}(m), \text{operation}(m) \rangle \mapsto m$ for each match m .
 I : A list of instruction descriptions.

Output: A (potentially empty) set of producer packs of x .

```
1 if there are dependent values in x then
2   | return {}
3 end
4 producers  $\leftarrow$  {}
5 for  $vinst \in I$  do
6   | matches  $\leftarrow$  []
7   | for  $i \leftarrow 1$  to number of lanes of vinst do
8     |  $f \leftarrow$  the  $i$ 'th operation of  $vinst$ 
9     |  $m \leftarrow M[\langle x_i, f \rangle]$ 
10    | if  $x_i$  is don't-care or  $m$  is not null then
11      | append  $m$  to matches
12    | end
13  | end
14  | if  $|matches| =$  number of lanes of vinst then
15    |  $producers \leftarrow producers \cup pack(vinst, matches)$ 
16  | end
17 end
18 return producers
```

The beginning of the algorithm is in the MatchManager. This code does not construct a match table, which is a map from live-out values and operations to a matching expression, according to the pseudo code in the paper. The live-out values are bound in the construction of a vector pack, which happens elsewhere.

MatchManager::MatchManager(ArrayRef<const InstBinding*> Insts,

```

                                Function &F) {
for (auto &Inst : Insts)
    for (auto &LaneOp : Inst->getLaneOps())
        OpMatches.FindAndConstruct(LaneOp.getOperation());

for (auto &I : instructions(&F))
    match(&I);

for (auto &KV : OpMatches) {
    auto &Matches = KV.second;
    std::sort(Matches.begin(), Matches.end(), sortByOutput);
}
}

```

2 Packer

In the VeGen project, I have listed all the header files and marked their def-use relationships. To understand all the concepts defined by the author, it is necessary to read these header files in a top-down manner.

header	user
InstSema.h	['VectorPackContext.h', 'Packer.h', 'IRVec.h', 'MatchManager.h']
VectorPackContext.h	['Packer.h', 'VectorPack.h']
LoopUnrolling.h	['UnrollFactor.h']
VectorPack.h	['VectorPackSet.h']
VLoop.h	['Packer.h']
DependenceAnalysis.h	['Packer.h']
ControlDependence.h	['Packer.h']
MatchManager.h	['Packer.h']
Reduction.h	['VectorPack.h']
IntrinsicBuilder.h	['InstSema.h']