Algorithm 1 CalculateTLs(components, interfaces, packages, methods, unitsDatatypes, pathsOfCompUnit)

unitMappings ← standaloneHeuristic$_{name}$(components, unitsDatatypes)
unitMappings ← selectBest({(archItem,codeItem,conf)} unitMappings)
inheritedMappings ← dependentHeuristic$_{inheritance}$(unitMappings)
unitMappings ← max(unitMappings ∪ inheritedMappings)

packageMappings ← map(package, components, packages)
ambigPackages ← applyH$_{ambigPkg}$(packageMappings)
packageMappings.remove(ambigPackages)

interfMethodMappings ← map(name, interfaces ∪ methods)

pathMappings ← map(path, components, pathsOfCompUnit)

unitPackageMappings ← matchFirst(unitMappings, packageMappings)
similarMappings ← dependentHeuristic$_{commonWords}$(unitPackageMappings)
unitPackageMappings ← max(similarMappings, unitPackageMappings)

ambigCompRelations ← dependentHeuristic$_{compRelations}$(unitPackageMappings)
unitPackageMappings.remove(ambigCompRelations)

possibleTLs ← interfMethodMappings ∪ pathMappings
possibleTLs ← possibleTLs ∪ unitPackageMappings
possibleTLs ← max(possibleTLs)
provideRelationMappings ← dependentHeuristic$_{interfProv}$(possibleTLs)
return possibleTLs.remove(provideRelationMappings)

Algorithm 2 StandaloneHeuristic$_H$(archItems, codeItems)

```
mappings ← ∅
for all archItem ← archItems, codeItem ← codeItems do
    confidence ← similarity_H(archItem,codeItem)
    mappings ← mappings ∪ (archItem, codeItem, confidence)
return mappings
```

Algorithm 3 SelectBest(mappings : List of (e1,e2,conf))

```
bestMappings ← ∅
for all e1 ← mappings do
    for all (e2, conf) ← mappings[e1] do
        if conf is maxConf(mappings[e1]) then
            bestMappings.add((e1, e2, conf))
return bestMappings
```

Algorithm 4 DependentHeuristic$_H$(mappings)

```
for all m ← mappings do
    for all m_a ← affectedMappings_H(m) do
        m_a.confidence ← getUpdatedConfidence_H(m_a, m)
return mappings
```

Algorithm 5 max(mappings : List of (e1,e2,conf))

```
bestMappings ← ∅
for all (e1, e2, conf) in mappings[(e1, e2)] do
    if conf is maxConf(mappings[(e1, e2)]) then
        bestMappings.add((e1, e2, conf))
return bestMappings
```

Algorithm 6 map(StandaloneHeuristic, archItems : List of archItem, codeItems : List of codeItem)

```
mappings ← standaloneHeuristic_StandaloneHeuristic(codeItems, archItems)
bestArchMappings ← selectBest(mappings)
bestMappings ← selectBest(mappings)
return bestMappings
```

Algorithm 7 matchFirst(set1 : List of (e1,e2,conf), set2 : List of (e1,e2,conf))

```
finalSet ← set1
for all s ← set2 do
    if (e1, e2, _) not in s then
        finalSet.add((e1, e2, conf))
return finalSet
```