0.1 Type Ambiguity Error Handler

0.1.1 Handler Algorithm

Algorithm 1: Handle

Input: L, B, R, M

```
1  ID_L ← {n ∈ A_L | n.type = IMPORTDECL};
2  ID_R ← {n ∈ A_R | n.type = IMPORTDECL};
3  if ID_L = ∅ ∨ ID_R = ∅ then return;
4  M_U ← textualMerge(treeToText(L), treeToText(B), treeToText(R));
5  I_L, I_R ← extractInsertions(M_U);
6  cs ← extractConflicts(M_U);
7  c ← compile(M_U);
8  ps ← problems(c);
9  foreach l ∈ ID_L do
10     m_l ← extractPackageMember(l.body);
11     foreach r ∈ ID_R do
12         m_r ← extractPackageMember(r.body);
13         if m_l = m_r then
14             p ← importDeclarationsProblem(l, r, ps);
15             if p ≠ null then
16                 m ← find(m ∈ M → m.body = l.body);
17                 m.body ← conflict(l.body, ε, r.body);
18                 m ← find(m ∈ M → m.body = r.body);
19                 removeNode(m, M);
20                 ps ← ps − p;
21                 break;
22             end
23         else if (m_l = ∗ ∨ m_r = ∗) ∧ importDeclarationsConflict(l, r, cs) then
24             I ← I_L;
25             m ← m_r;
26             if m_l ≠ ∗ then
27                 I ← I_R;
28                 m ← m_l;
29             end
30             i ← find(i ∈ I → import ∉ i ∧ m ∈ i);
31             if i ≠ null then
32                 m ← find(m ∈ M → m.body = l.body);
33                 m.body ← conflict(l.body, ε, r.body);
34                 m ← find(m ∈ M → m.body = r.body);
35                 removeNode(m, M);
36                 break;
37             end
38         end
39     end
40 end
```

Algorithm 2: Import Declarations Problem

Input: l, r, ps
Output: compilation problem in ps concerning l and r import declarations, if there is one

```
1 foreach p ∈ ps do
2 |   if p.type = COLLISION then
3 |   |   foreach a ∈ p.arguments do
4 |   |   |   if a ∈ l.body ∨ a ∈ r.body then return p;
5 |   |   end
6 |   else if p.type = AMBIGUITY then return p;
7 end
8 return null;
```

Algorithm 3: Import Declarations Conflict

Input: l, r, cs
Output: wether there is a unstructured conflict in cs concerning l and r import declarations

```
1 foreach c ∈ cs do
2 |   if l.body ∈ c.left ∧ r.body ∈ c.right then return TRUE;
3 end
4 return FALSE;
```