

0.1 Type Ambiguity Error Handler

0.1.1 Handler Algorithm

Algorithm 1: Handle	
Input:	L, B, R, M
1	$ID_L \leftarrow \{n \in A_L \mid n.type = IMPORTDECL\};$
2	$ID_R \leftarrow \{n \in A_R \mid n.type = IMPORTDECL\};$
3	if $ID_L = \emptyset \vee ID_R = \emptyset$ then return;
4	$T_L \leftarrow \text{treeToText}(L);$
5	$T_B \leftarrow \text{treeToText}(B);$
6	$T_R \leftarrow \text{treeToText}(R);$
7	$M_U \leftarrow \text{textualMerge}(T_L, T_B, T_R);$
8	$I_L, I_R \leftarrow \text{extractInsertions}(M_U);$
9	$cs \leftarrow \text{extractConflicts}(M_U);$
10	$c \leftarrow \text{compile}(M_U);$
11	$ps \leftarrow \text{problems}(c);$
12	foreach $l \in ID_L$ do
13	$m_l \leftarrow \text{extractPackageMember}(l.body);$
14	foreach $r \in ID_R$ do
15	$m_r \leftarrow \text{extractPackageMember}(r.body);$
16	if $m_l = m_r$ then
17	$p \leftarrow \text{importDeclarationsProblem}(l, r, ps);$
18	if $p \neq \text{null}$ then
19	$m \leftarrow \text{find}(m \in M \rightarrow m.body = l.body);$
20	$m.body \leftarrow \text{conflict}(l.body, \varepsilon, r.body);$
21	$m \leftarrow \text{find}(m \in M \rightarrow m.body = r.body);$
22	$\text{removeNode}(m, M);$
23	$ps \leftarrow ps - p;$
24	break;
25	end
26	else if $(m_l = * \vee m_r = *) \wedge \text{importDeclarationsConflict}(l, r, cs)$ then
27	if $m_l \neq *$ then
28	$I \leftarrow I_R;$
29	$m \leftarrow m_l;$
30	else
31	$I \leftarrow I_L;$
32	$m \leftarrow m_r;$
33	end
34	$i \leftarrow \text{find}(i \in I \rightarrow \text{IMPORT} \notin i \wedge m \in i);$
35	if $i \neq \text{null}$ then
36	$m \leftarrow \text{find}(m \in M \rightarrow m.body = l.body);$
37	$m.body \leftarrow \text{conflict}(l.body, \varepsilon, r.body);$
38	$m \leftarrow \text{find}(m \in M \rightarrow m.body = r.body);$
39	$\text{removeNode}(m, M);$
40	break;
41	end
42	end
43	end
44	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  $p \in ps$  do
2   if  $p.type = COLLISION$  then
3     foreach  $a \in p.arguments$  do
4       if  $a \in l.body \vee a \in r.body$  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:** whether there is an unstructured conflict in cs concerning l and r import declarations

```

1 foreach  $c \in cs$  do
2   if  $l.body \in c.left \wedge r.body \in c.right$  then return true;
3 end
4 return false;

```