1 Multiple Initialization Blocks Handler

1.1 Handler Algorithm

Algorithm 1: Handle

Input: L, B, R, M

$A_L \leftarrow \{l \in L \mid (\neg\exists b \in B)(l.id = b.id)\}$;
$A_R \leftarrow \{r \in R \mid (\neg\exists b \in B)(r.id = b.id)\}$;
$D_B \leftarrow \{b \in B \mid (\neg\exists l \in L)(b.id = l.id) \wedge (\neg\exists r \in R)(b.id = r.id)\}$;

$IB_L \leftarrow \{n \in A_L \mid n.type = INITBLOCK\}$;
$IB_R \leftarrow \{n \in A_R \mid n.type = INITBLOCK\}$;
$IB_B \leftarrow \{n \in D_B \mid n.type = INITBLOCK\}$;

$E_L \leftarrow$ `editedNodes(`IB_L, IB_B`)`;
$E_R \leftarrow$ `editedNodes(`IB_R, IB_B`)`;

Algorithm 2: Edited Nodes

Input: IB, IB_B
Output: set of pairs (b, a) consisting of a deleted base node b in IB_B and its correspondent branch added node a in IB

$D \leftarrow \{d \in IB_B \mid (\neg\exists a \in IB)(d.body = a.body)\}$;
$A \leftarrow \{a \in IB \mid (\neg\exists d \in IB_B)(a.body = d.body)\}$;

$matches \leftarrow \emptyset$;
foreach $a \in A$ **do**
    $S \leftarrow \{d \in D \mid a.body \approx d.body\}$;
    $b \leftarrow \underset{s \in S}{\text{argmax}}$ (`similarity(`$s.body$, $a.body$`)`);
    if $b \neq null$ **then** $matches \leftarrow matches \cup (b, a)$;
end
return $matches$