Algorithm 1: Related state collection

Input: Migrated test cases $TestCases$, target app app_t, time limit max_time
Output: Related states S_{final}

$T_{long} \leftarrow []$, $E_{long} \leftarrow []$, $S_{long} \leftarrow []$, $W_{long} \leftarrow \{\}$;
for ***each*** T_{tmp} *in* $TestCases$ **do**
  $driver \leftarrow Launch(app_t)$;
  $E_{tmp} \leftarrow []$, $S_{tmp} \leftarrow []$, $W_{tmp} \leftarrow \{\}$;
  for ***each*** *event in* T_{tmp} **do**
    if $driver.Execute(event) = True$ **then**
      $E_{tmp}.Append(event)$;
      $S_{tmp}.Append(driver.GetState())$;
      $W_{tmp} \leftarrow driver.GetWidgets()$;
    end
  end
  if $Len(T_{tmp}) > Len(T_{long})$ **then**
    $T_{long} \leftarrow T_{tmp}$;
    $E_{long} \leftarrow E_{tmp}$;
    $S_{long} \leftarrow S_{tmp}$;
    $W_{long} \leftarrow W_{tmp}$;
  end
end
$W_{all} \leftarrow TestCases.GetAllWidgets()$;
$W_{remain} \leftarrow \{w$ for w in W_{all} and not in $W_{long}\}$;
$S_{final} \leftarrow S_{long}$;
$driver \leftarrow Launch(app_t)$;
$driver.ExecuteTest(T_{long})$;
while $W_{remain} \neq \emptyset$ *and the time duration does not exceed* max_time **do**
  $widgets \leftarrow driver.GetWidgets()$;
  $event \leftarrow GenEventForRandomWidget(widgets)$;
  $driver.Execute(event)$;
  if $W_{remain} \cap driver.GetWidgets() \neq \emptyset$ **then**
    $W_{remain}.RemoveAll(driver.GetWidgets())$;
    $S_{final}.Append(driver.GetState())$;
    continue;
  end
  $driver.BacktrackToLastState()$;
end
return S_{final};