**Input:** EOG Gπ := ⟨Eπ, , £π⟩.

**Output:** Set of independent constraints of Gπ.

let Ewrite := set of write nodes of Gπ.

let βindependent beinitial set of independent constraints.

**foreach** (write node W in Ewrite) **do**

let Rs := set of read nodes corresponding with W.

let CRs := calculation of var(Rs), corresponding with W.

βindependent.add(var(W) == CRs).

**end.**

**return** βindependent.

**Algorithm 1. Independent constrains generation.**

**Input:** EOG Gπ := ⟨Eπ, , £π⟩.

**Output:** Set of read/write variables constraints of Gπ.

let Eread := set of read nodes of Gπ.

let βrw be initial set of read/write variables constraints.

**foreach** (read node R in Eread) **do**

let Ws := set of write nodes that can be read by R.

let RWs be initial set of read-write links created by R.

**foreach** (write node W in Ws) **do**

let signature := unique literal defined by R and W.

let constraint := (signature => (indexedVar(R) == indexedVar(W)).

RWs.add(constraint).

βrw.add(constraint).

**end.**

let atLeastOne := at least one constraint in RWs is true.

let atMostOne := at most one constraint in RWs is true.

βrw.add(atLeastOne).

βrw.add(atMostOne).

**end.**

**return** βrw.

**Algorithm 2. Read/write variables constraints generation.**