**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.

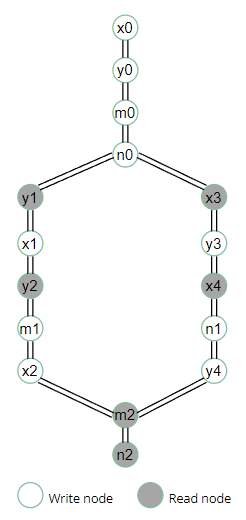
βrw.add(atLeastOne).

βrw.add(atMostOne).

**end.**

**return** βrw.

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



|  |  |
| --- | --- |
| int x = 1, y = 1, m = 0, n = 0;  void\* thr1(void \* arg) {  x = y + 1;  m = y;  x = 0;  }  void\* thr2(void \* arg) {  y = x + 1;  n = x;  y = 0;  }  void main() {  pthread\_t t1, t2;  pthread\_create(&t1, 0, thr1, 0);  pthread\_create(&t2, 0, thr2, 0);  pthread\_join(t1, 0);  pthread\_join(t2, 0);  assert (!(m == 2 && n == 3));  } | int x0 = 1, y0 = 1, m0 = 0, n0 = 0;  void\* thr1(void \* arg) {  x1 = y1 + 1;  m1 = y2 ;  x2 = 0;  }  void\* thr2(void \* arg) {  y3 = x3 + 1;  n1 = x4 ;  y4 = 0;  }  void main() {  pthread\_t t1, t2;  pthread\_create(&t1, 0, thr1, 0);  pthread\_create(&t2, 0, thr2, 0);  pthread\_join(t1, 0);  pthread\_join(t2, 0);  assert (!(m2 == 2 && n2 == 3));  } |
| (1) Chương trình gốc | (2) Mã SSA của chương trình |