Resource Allocation and Binding: Hierarchical Sequence Graph

Dr. Chandan Karfa

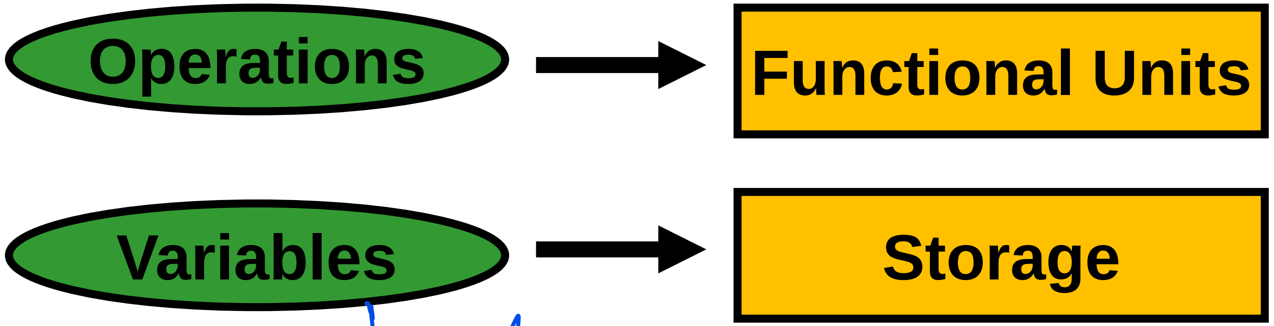
Department of Computer Science and Engineering

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

Recap: Allocation and Binding

- **Objectives: Maximize Resource sharing; hence, minimize resource usage**



Registers

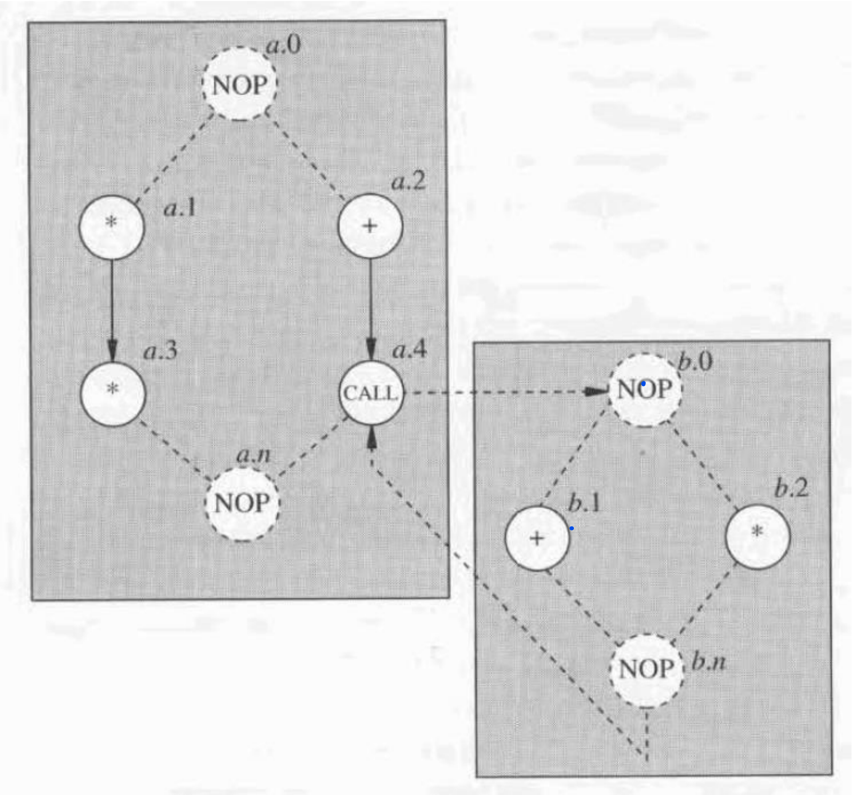
Subtasks:

1. **FU allocation & Binding**
2. **Register Allocation & Binding**

Binding for Hierarchical sequencing graphs

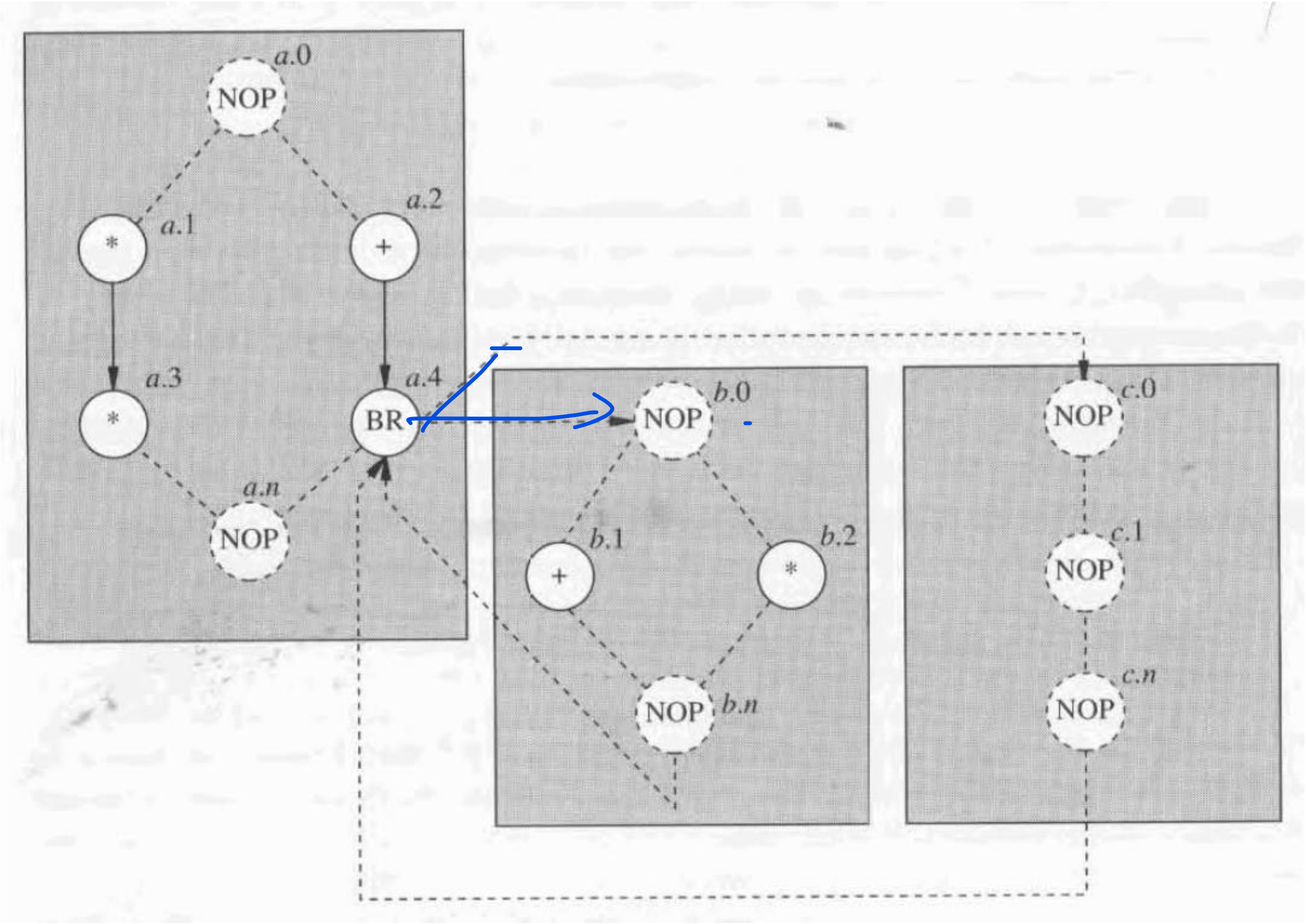
- Sequence graphs with
  - Function calls ✓
  - If-else ✓
  - Loops ✓
- Hierarchical conflict/compatibility graphs:
  - Easy to compute
  - Prevent sharing across hierarchy
- Flatten hierarchy:
  - Bigger graphs

Hierarchical sequencing graphs – function call

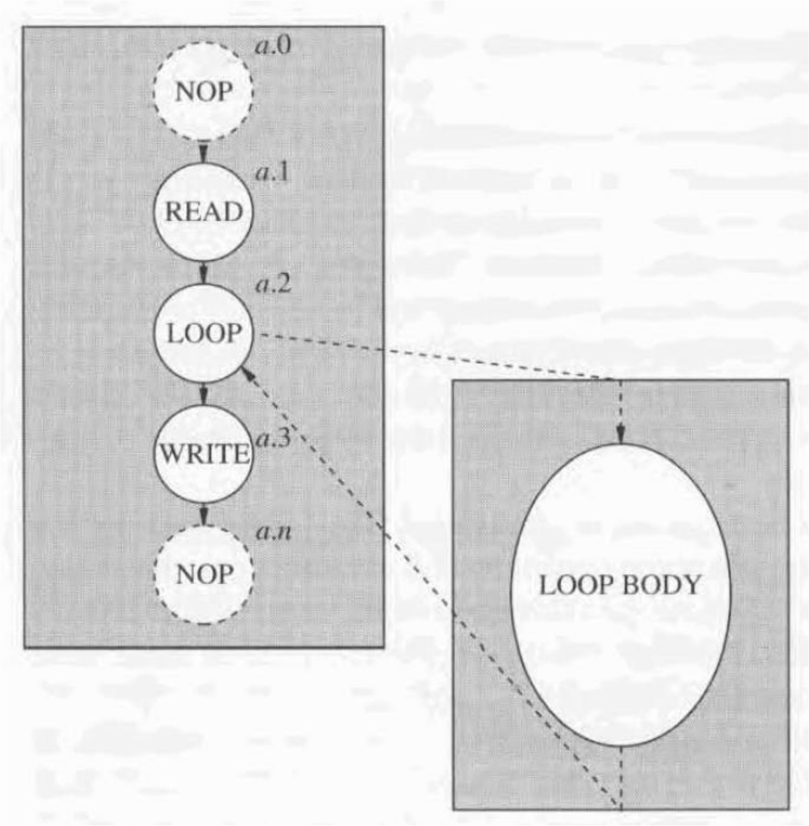


Function call

Hierarchical sequencing graphs – If-else

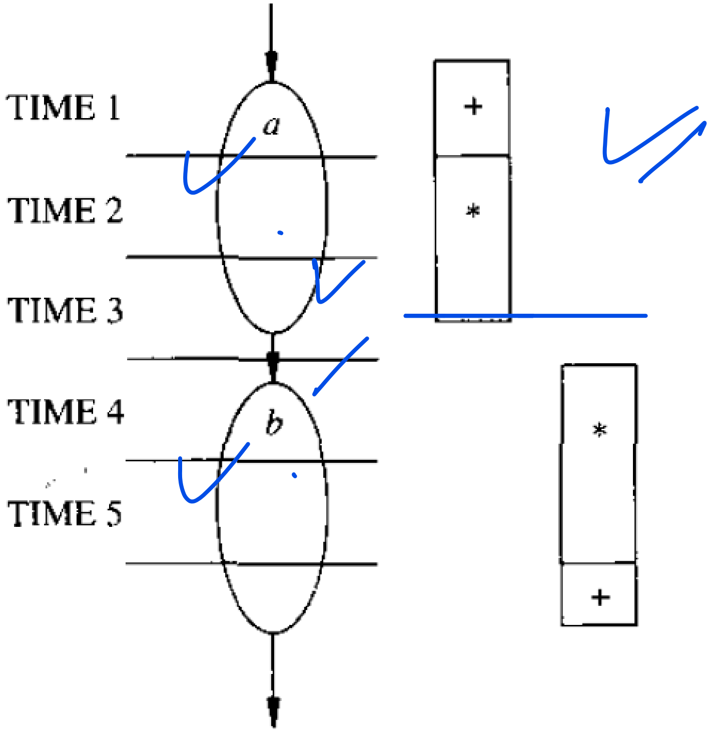


Hierarchical sequencing graphs – Loop



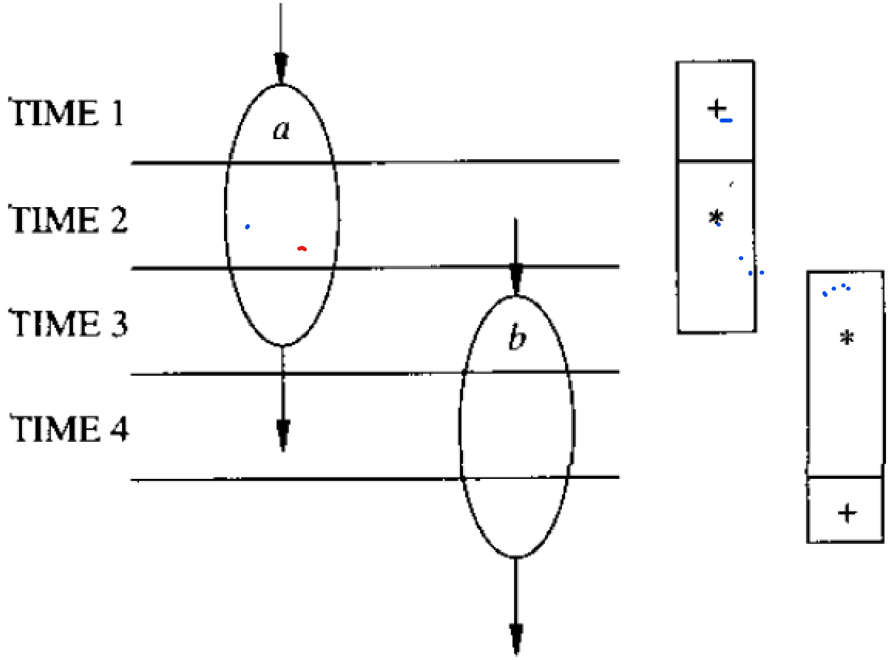
Function Calls

□ When two link vertices corresponding to different called models are not concurrent, any operation pair implementable by resources with the same type and in the different called models is **compatible.**



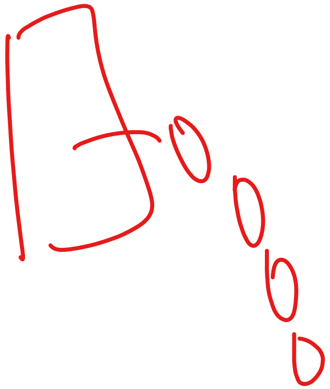
Function Calls

□ Concurrency of the called models does not necessarily imply conflicts of operation pairs in the models themselves

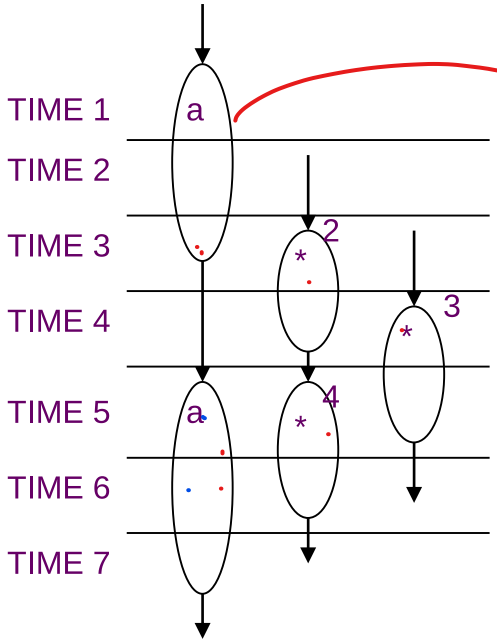


Calling a Function more than once

- When all models are called only once, the hierarchy is only a structured representation of the data-flow information. Thus conflict graphs is the interval graph.
- In case of multiple calls to a single function, the resultant **conflict graph may not be an interval graph.**



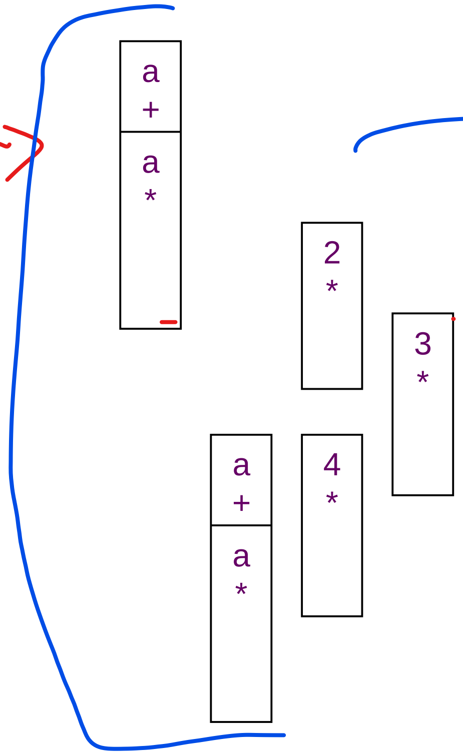
Example



(a)

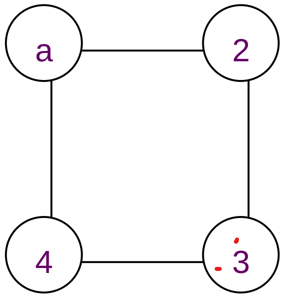
IG

Ch →



(b)

Conflict graph



(c)

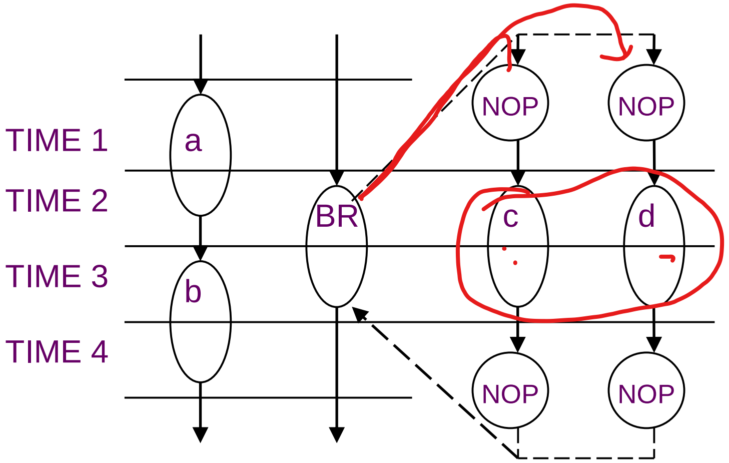
↓
Not chordal

So, not interval

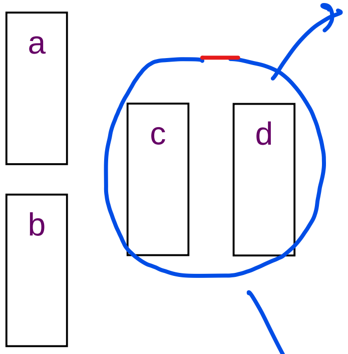
So, No left edge algo.

If-else

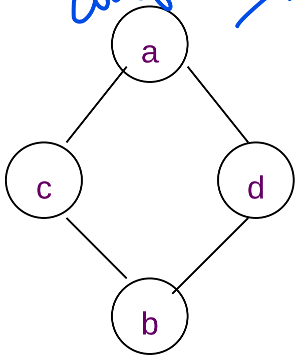
The resulting compatibility and conflict graphs may not have any special property



(a)



(b)



(c)

If else is mutually exclusive

conflict graph → not chordal

So not an intr al g.

if it was not if else then

(c → d) would have been an edge

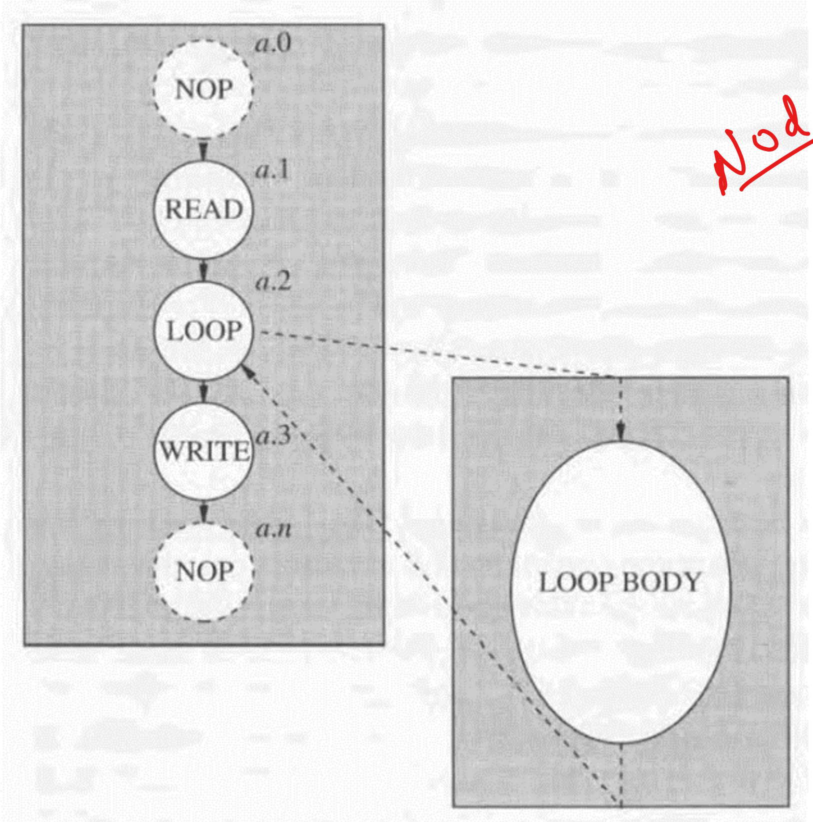
Loop

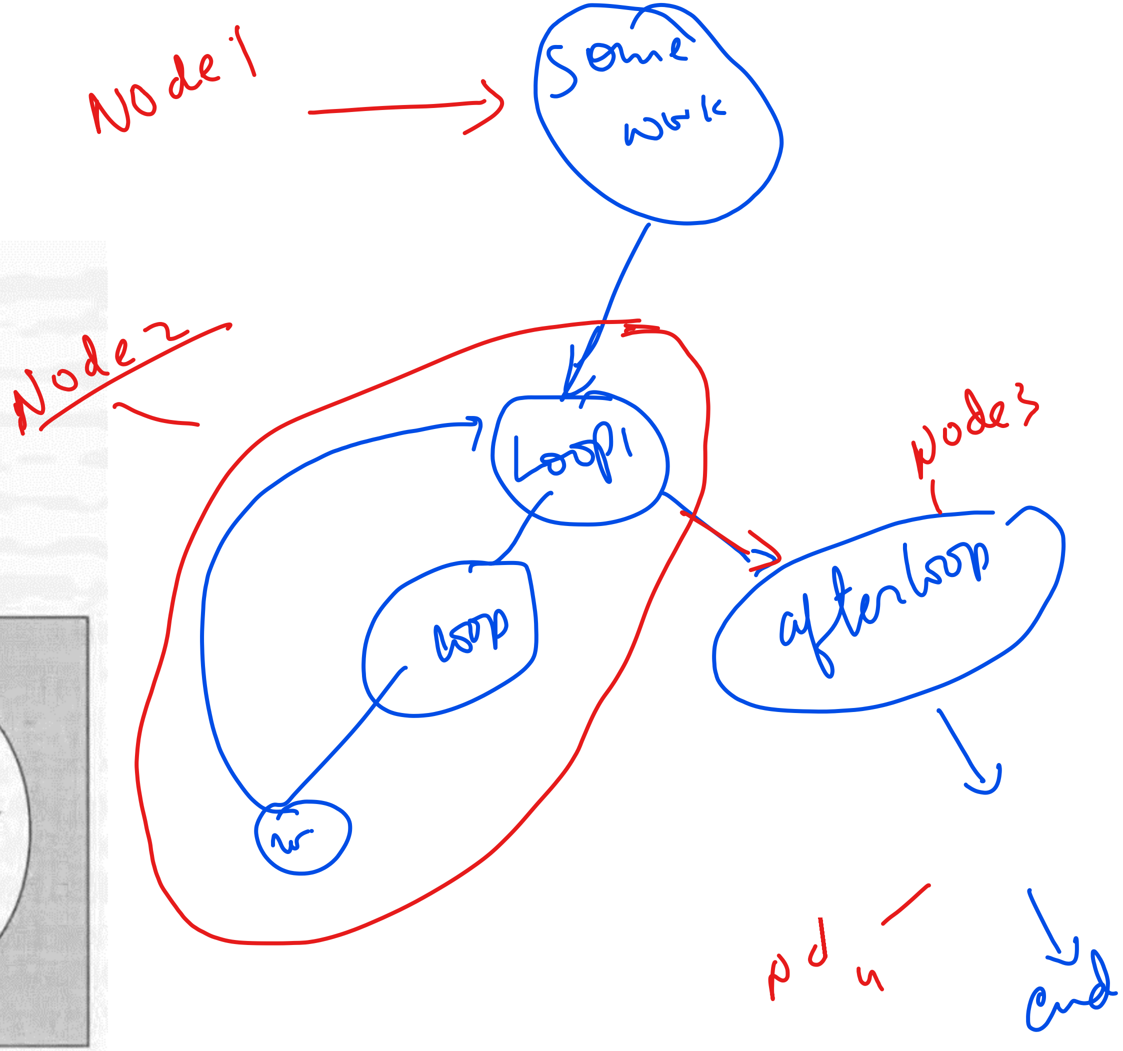
- □ Either Unroll or Iterative or pipeline
- □ Conflict graph is an interval graph

Chordal

complement

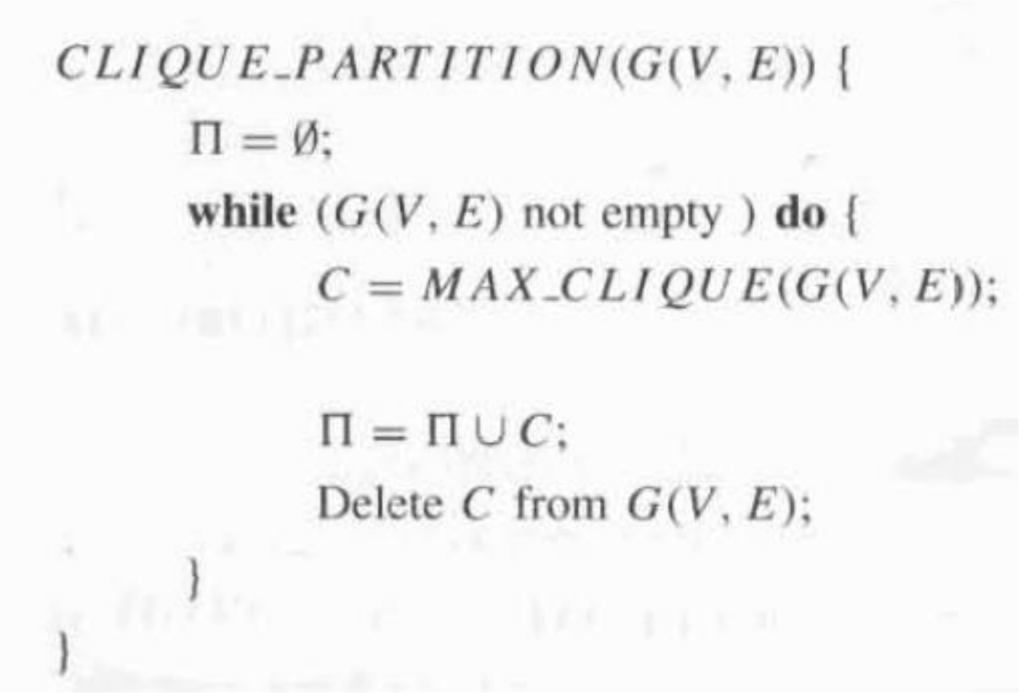
comparability





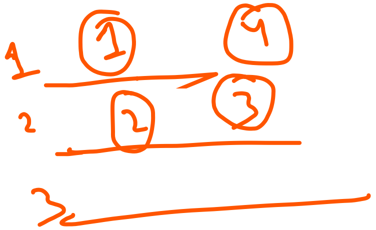
A Heuristic Algorithm for Clique Partition Problem

find NO. of cliques in compatibility graph

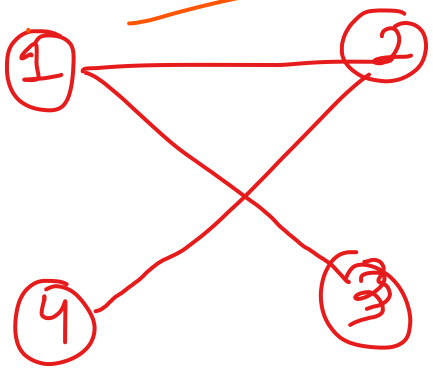


```
CLIQUE_PARTITION(G(V, E)) {
    $\Pi = \emptyset$;
    while (G(V, E) not empty ) do {
        $C = MAX\_CLIQUE(G(V, E))$;

        $\Pi = \Pi \cup C$;
        Delete C from G(V, E);
    }
}
```



compatible G.

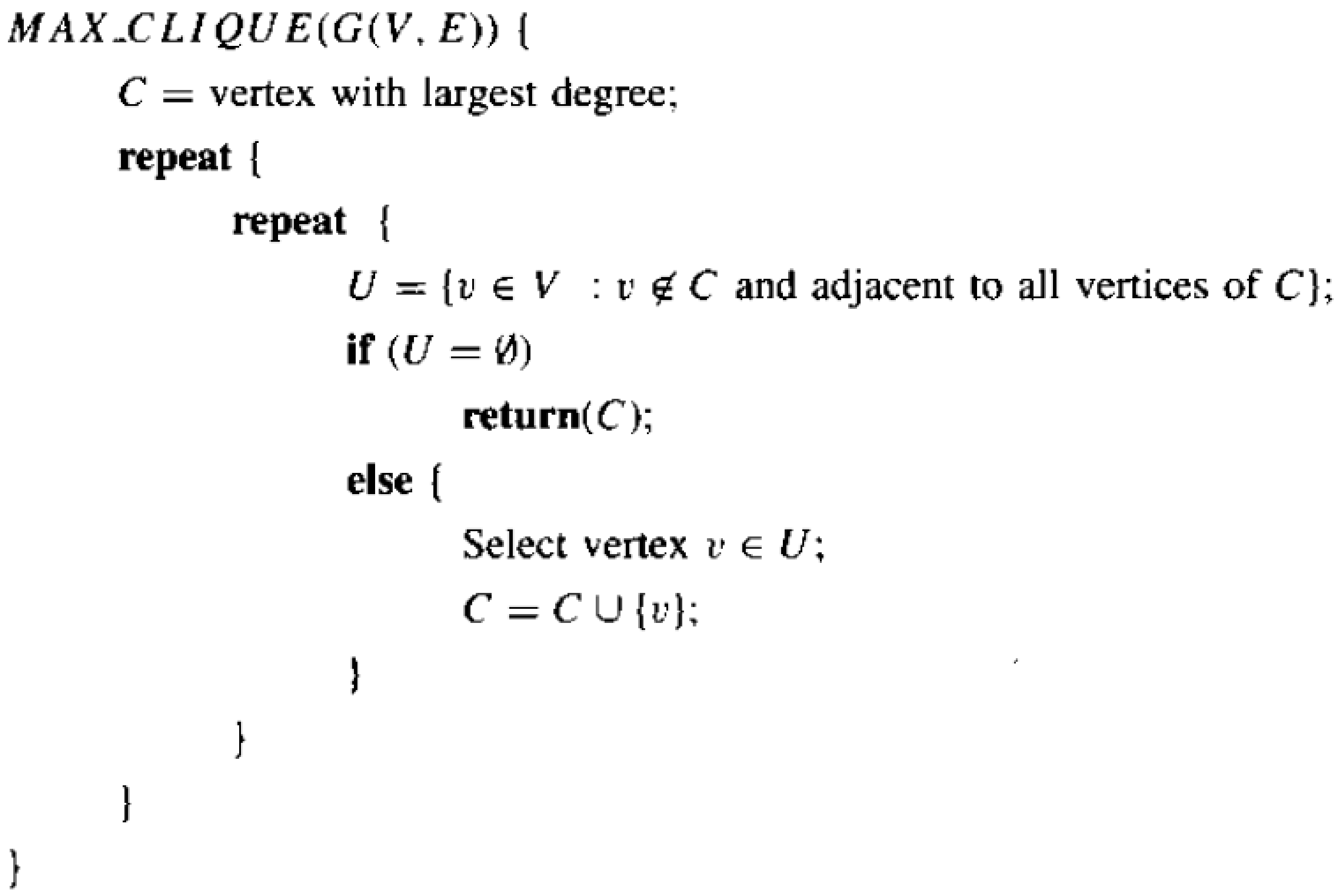


Now, we may get

3 cliques ((1,2)(4)(3))

or 2 cliques ((1,3) (2,4))

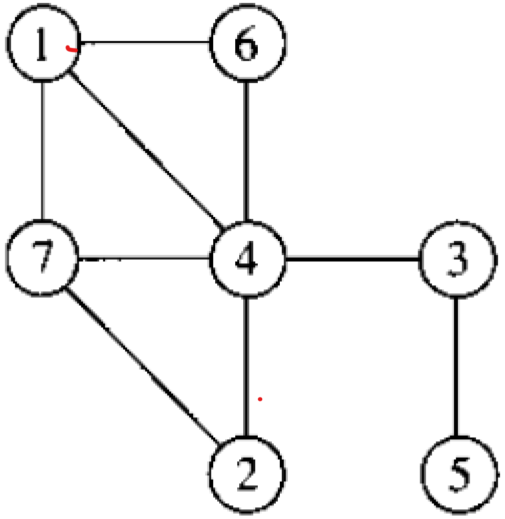
So, the heuristic p. may not give optimal.



```
$MAX\_CLIQUE(G(V, E))$ {
    $C$ = vertex with largest degree;
    **repeat** {
        **repeat** {
            $U = \{v \in V : v \notin C$ and adjacent to all vertices of $C\}$;
            **if** $(U = \emptyset)$
                **return**$(C)$;
            **else** {
                Select vertex $v \in U$;
                $C = C \cup \{v\}$;
            }
        }
    }
}
```

Example

conflict



Ch.NO ⇒ size of Lar Cliq.

Cl. cover ⇒ $\geq \alpha(G)$

↓

compatible

Thank You