



# Context Sensitive(CoS) SSA for Recursive Programs



Arpana Prajapati<sup>1</sup>, Sathwika Reddy<sup>1</sup>, Supriya Bhide<sup>1</sup>  
Pritam Gharat<sup>2</sup>, Uday Khedker<sup>1</sup>, Alan Mycroft<sup>3</sup>

<sup>1</sup>IIT Bombay, <sup>2</sup>Microsoft Research India, <sup>3</sup>University of Cambridge

## 1 Challenges

- Recursive functions can have seemingly infinite invocations
- Leads to an unbounded number of data dependencies across different invocations

## 2 Key Ideas

- Despite the infinite invocations, the number of defs is finite
- After a certain number of iterations, dependency patterns start repeating
- Need to detect the **longest subsequence** where dependencies follow the same pattern

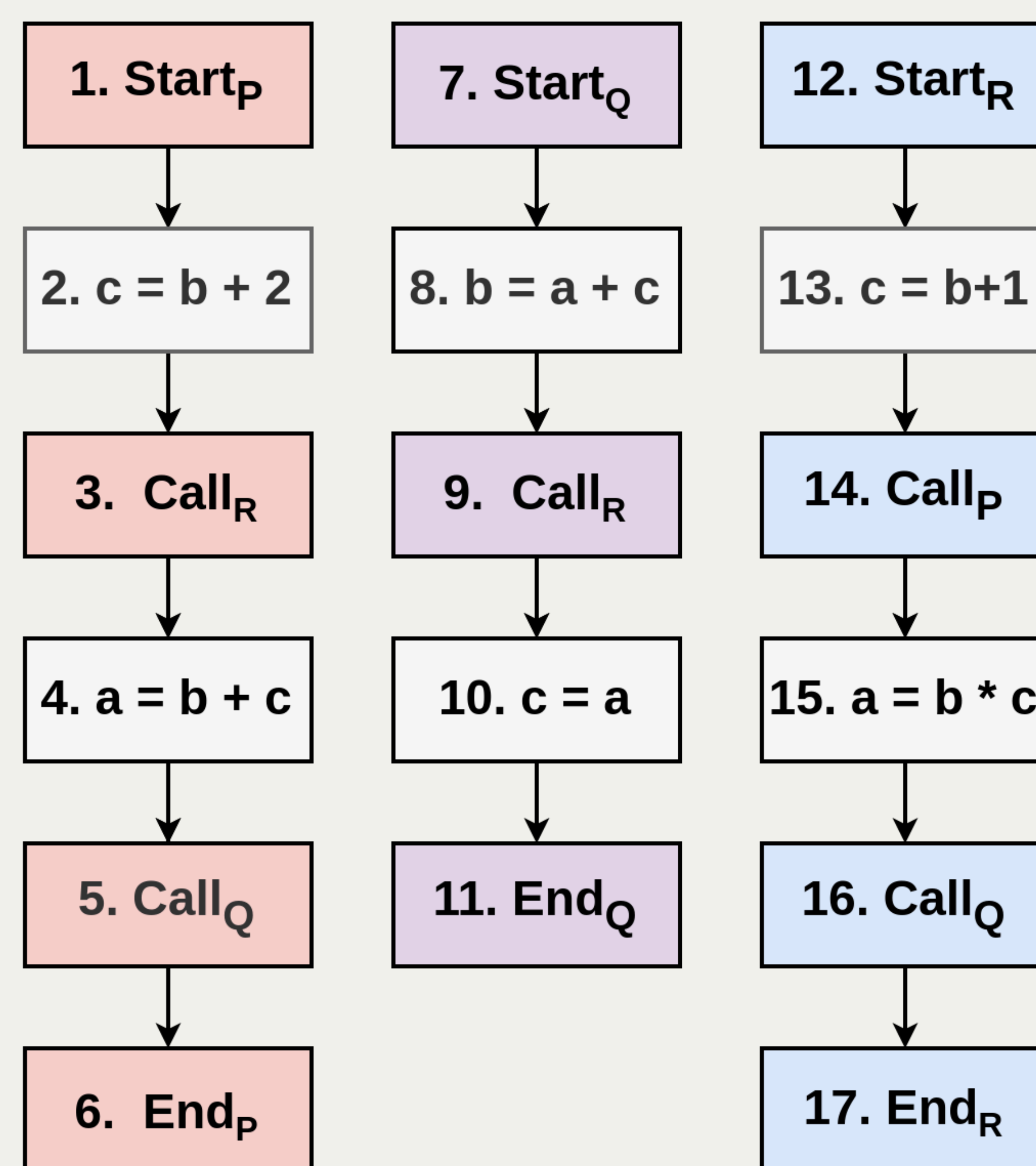
## 3 Current Implementation

- Calculates each DDG,  $\Delta_1, \Delta_2, \dots, \Delta_i$ , of a procedure from scratch using data flow equations
- $\Delta_i$  doesn't use previous  $\Delta$  information in its calculation
- Lot of redundant calculations

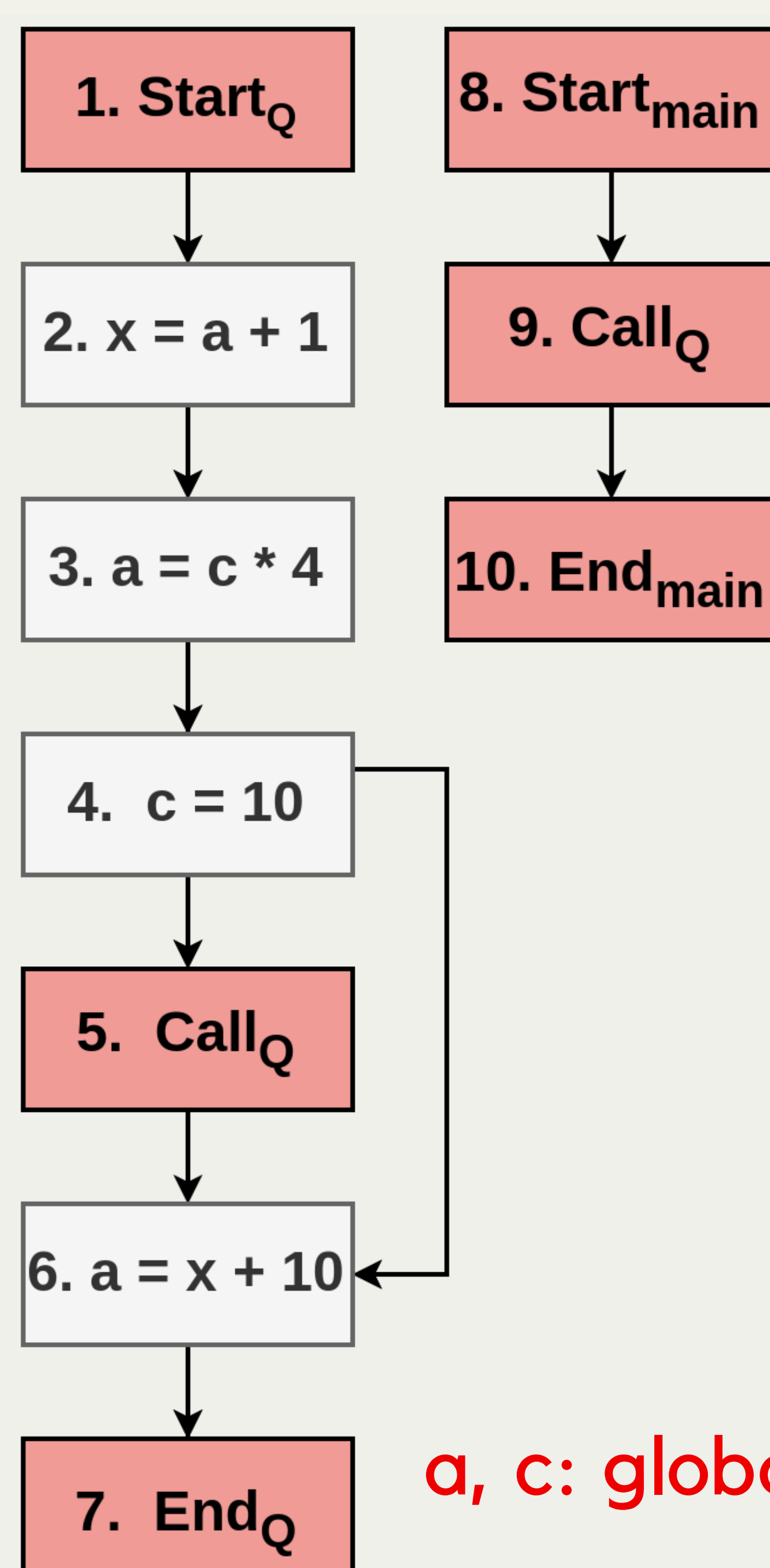
## 4 Incremental DDGs

- Construct  $\Delta_i$  using previous  $\Delta$  information
- Since the dependency pattern eventually repeats, a **summary** can be computed to capture the structure
- Summary can be computed iteratively until longest Repeating subsequence of dependencies is identified
- Used to construct subsequent DDGs without referring to the original program

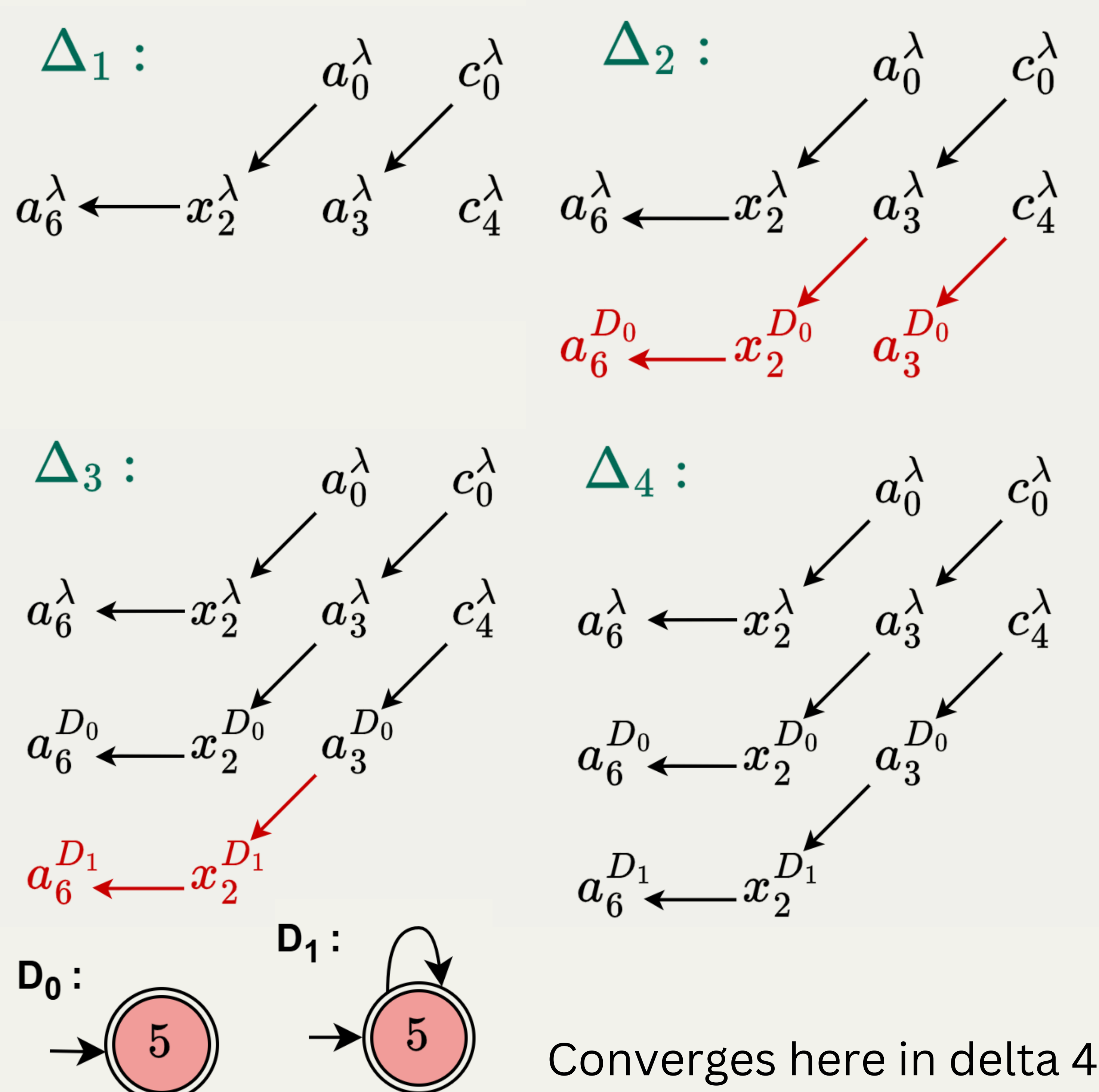
## 7 Indirect Recursive Example



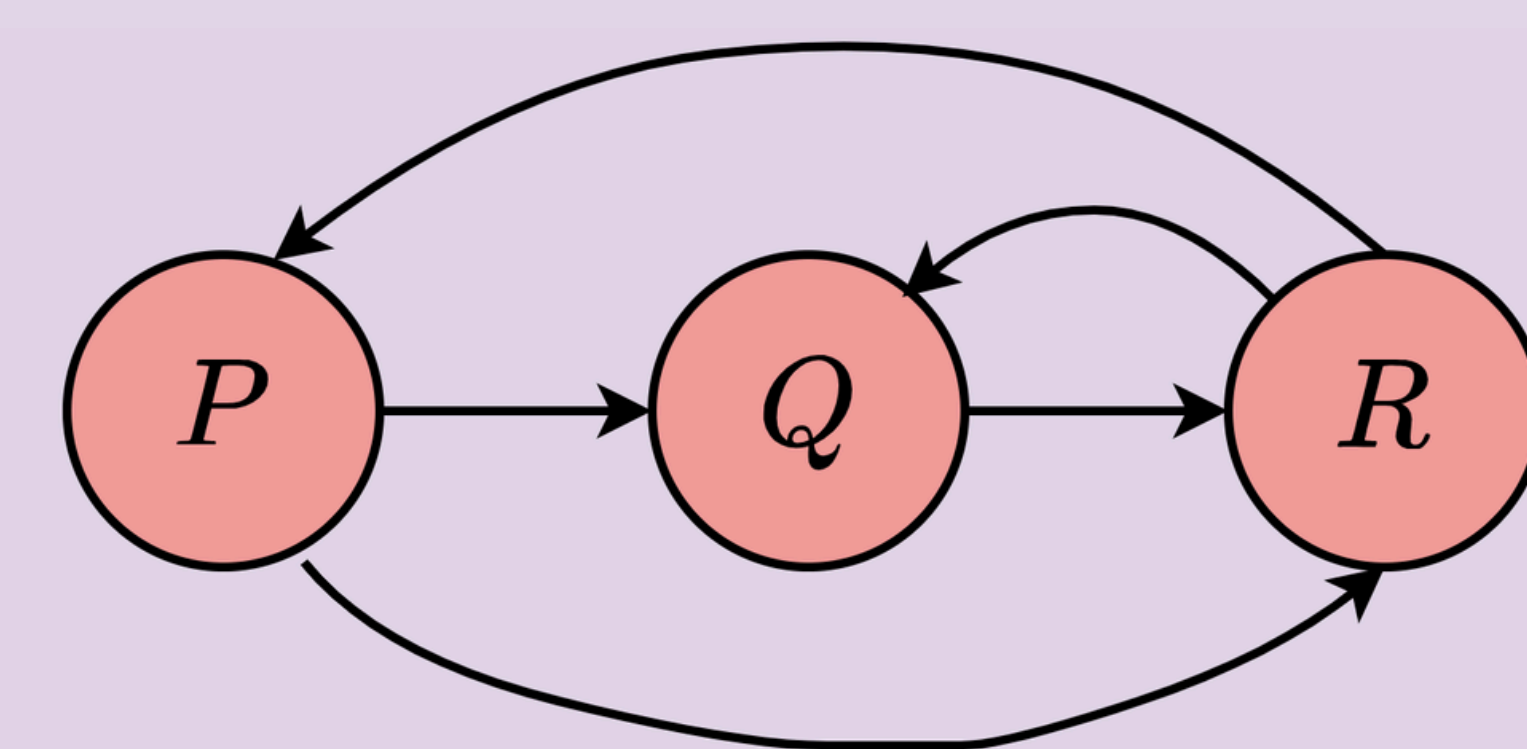
## 5 Example DDG of Self-Recursive Program



a, c: global



## 8 Observations (Indirect Recursion)



Direction of construction of DDGs

- One Possible dependency pattern is **P→Q→R** (var in R depends on var in Q which further depends on var in P)
- In  $\Delta_2$  of P, **P→Q** dependency is captured, in  $\Delta_2$  of Q, **Q→R** is captured but **P→Q→R** is not captured in  $\Delta_2$  of any procedure
- P→Q→R** is captured in  $\Delta_3$  of P
- No. of  $\Delta$ 's required for summary calculation may depend on the max no. of **incompatible edges** in any non-repeating path
- Incompatible edges are the ones whose direction aligns with the direction of construction of DDGs
- No. of incompatible edges in **P→Q→R** is 2

## 6 Observations (Self-Recursive)

- Two invocations** of the procedure are sufficient to capture all the dependencies
- Further invocations do not introduce new variable dependencies—they only replicate existing dependencies in new contexts
- All dependencies within same context are captured in the first invocation
- The second captures dependencies from the outer function to its recursive instance
- All the  $\Delta_i$ ,  $i \geq 2$ , can be constructed from the summary obtained from  $\Delta_1$  and  $\Delta_2$