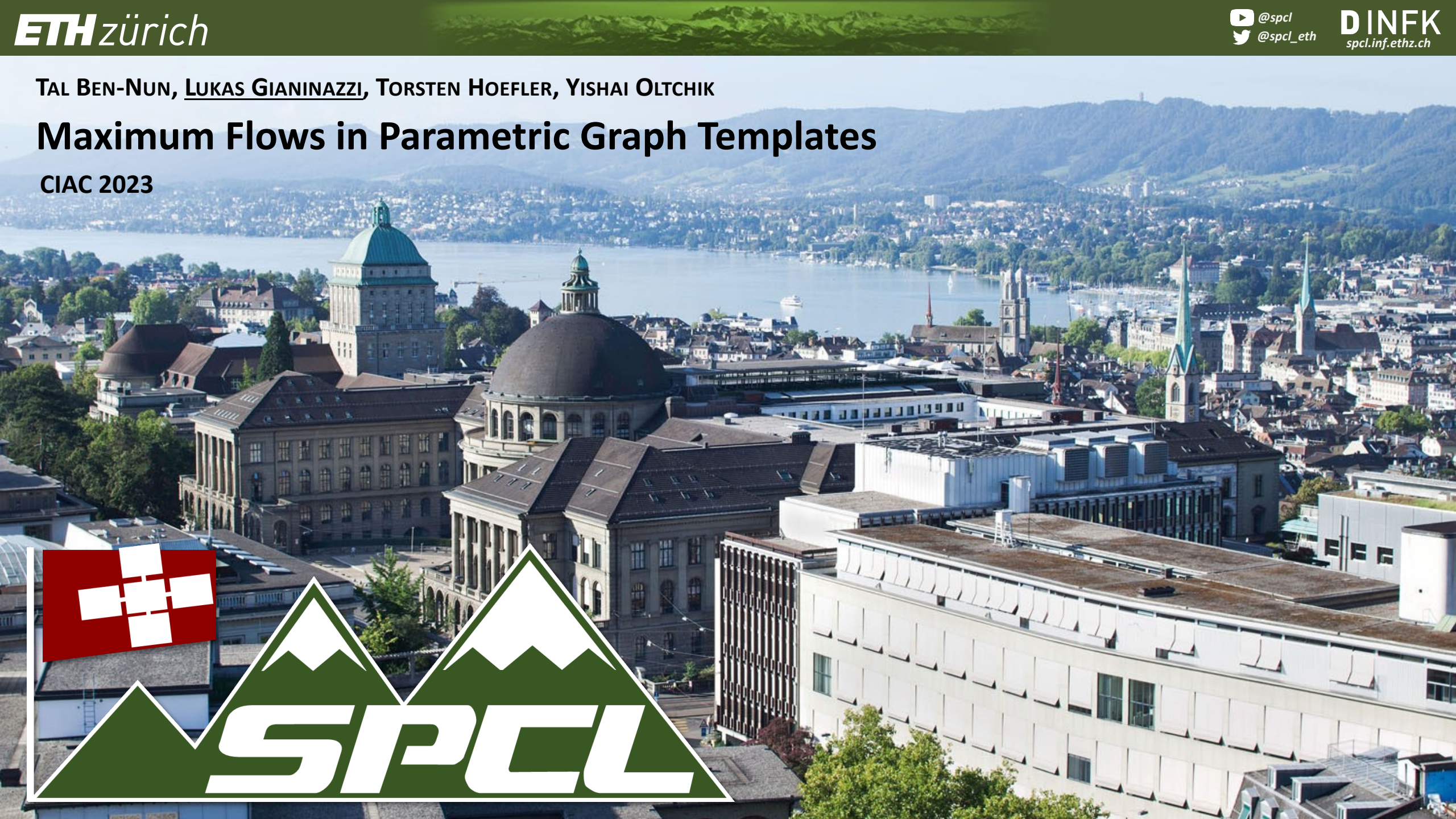
Tal Ben-Nun, Lukas Gianinazzi, Torsten Hoefler, Yishai Oltchik

Maximum Flows in Parametric Graph Templates

CIAC 2023



Example: Matrix Multiplication

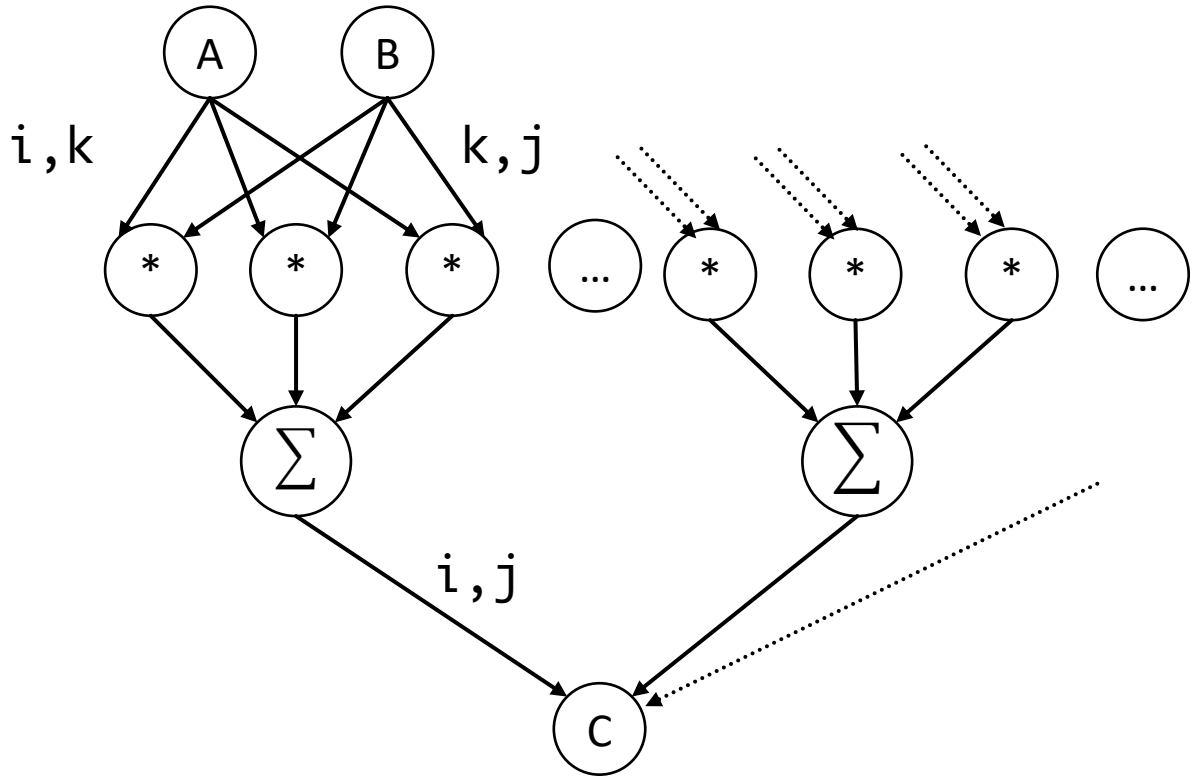
```
for (i=0; i<n; i++) {
   for (j=0; j<m; j++) {
      for (k=0; k<l; k++) {
         C[i, j] += A[i, k] * B[k, j]
      }
   }
}
```

Nested Loop Code

Example: Matrix Multiplication

```
for (i=0; i<n; i++) {
   for (j=0; j<m; j++) {
      for (k=0; k<l; k++) {
         C[i, j] += A[i, k] * B[k, j]
      }
   }
}
```

Nested Loop Code

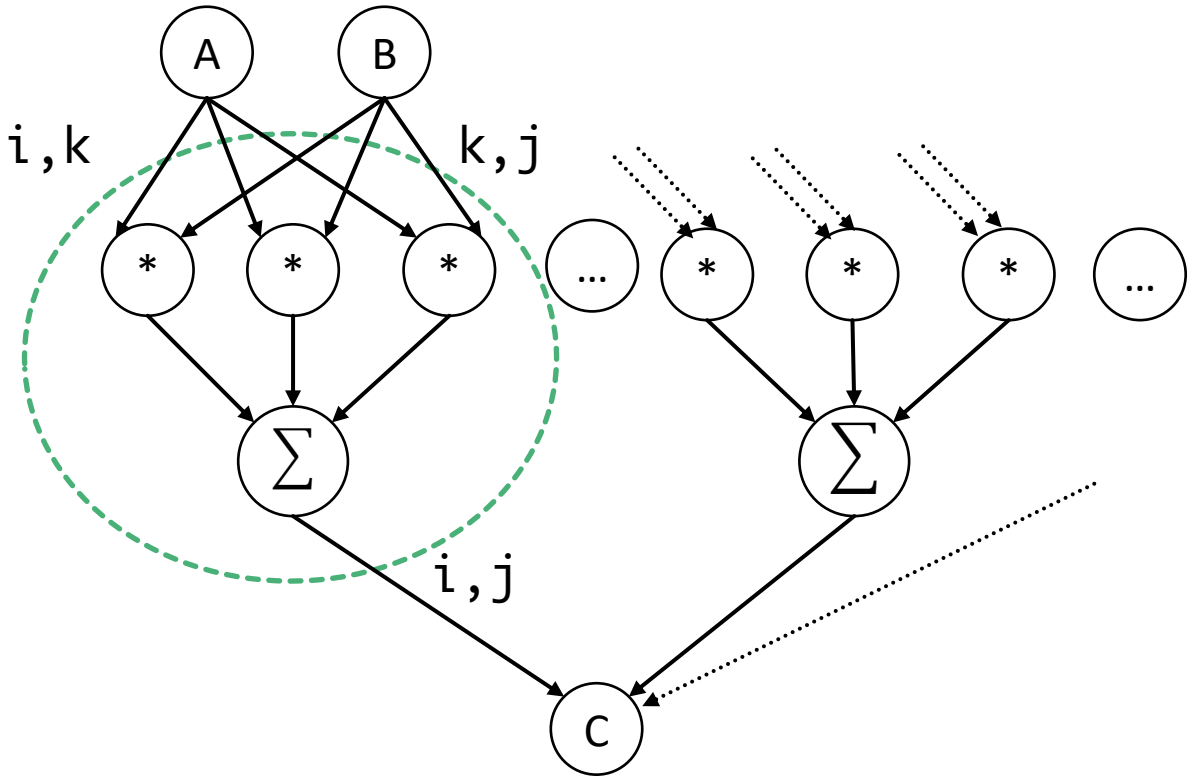


Dataflow Graph

Example: Matrix Multiplication

Cuts & flows
correspond to data movement

```
for (i=0; i<n; i++) {
   for (j=0; j<m; j++) {
      for (k=0; k<l; k++) {
         C[i, j] += A[i, k] * B[k, j]
      }
   }
}
```



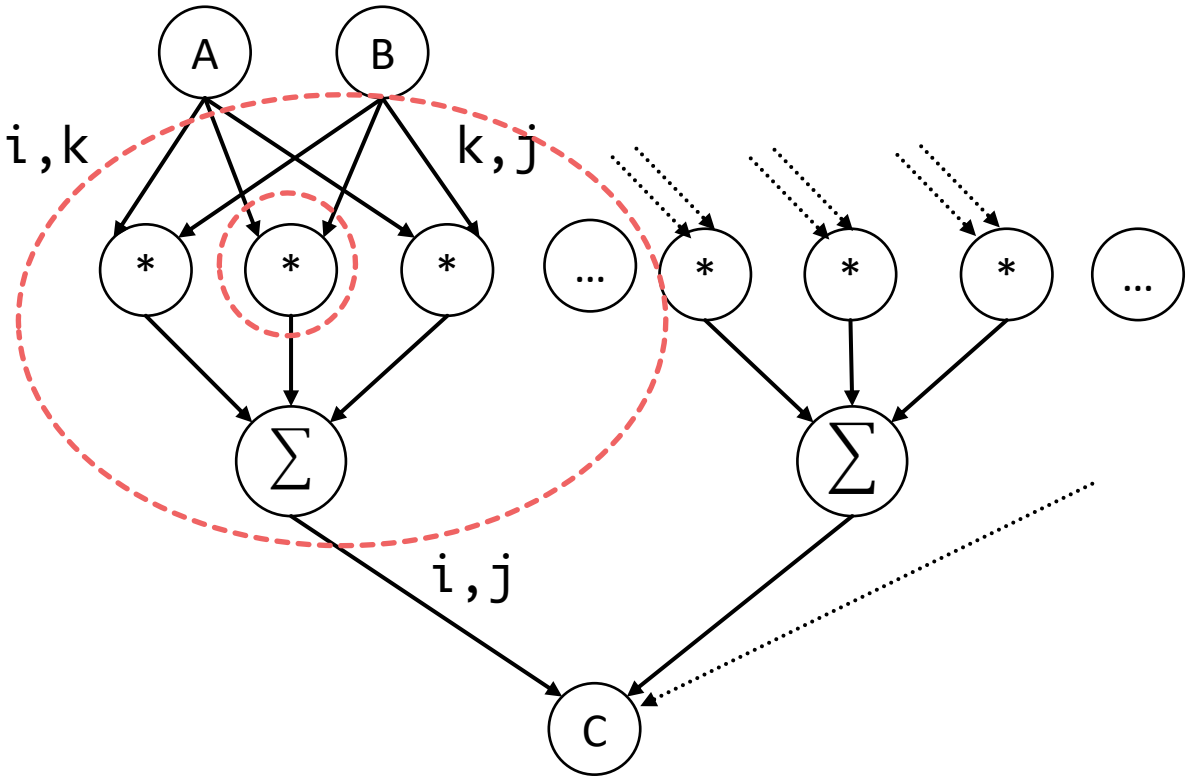
Nested Loop Code

Dataflow Graph

Example: Matrix Multiplication

Nested repeating subgraphs

```
for (i=0; i<n; i++) {
   for (j=0; j<m; j++) {
      for (k=0; k<l; k++) {
         C[i, j] += A[i, k] * B[k, j]
      }
   }
}
```

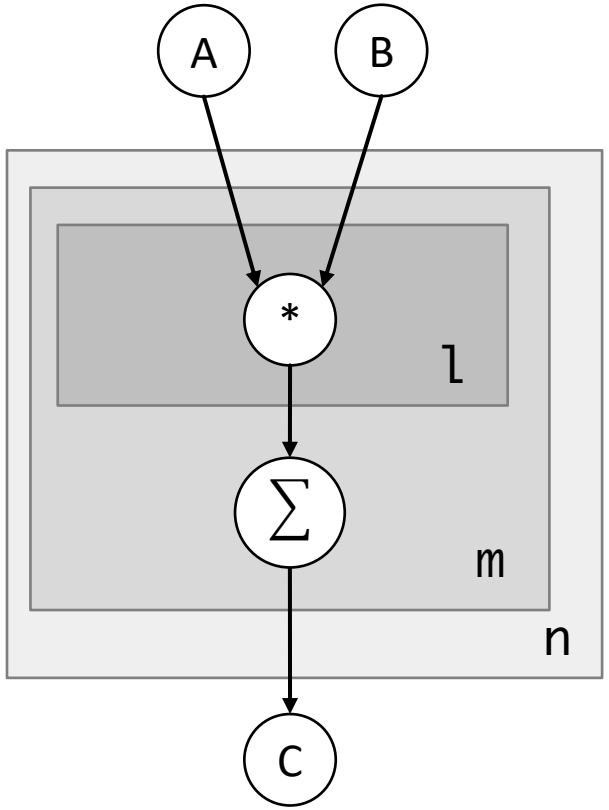


Nested Loop Code

Dataflow Graph

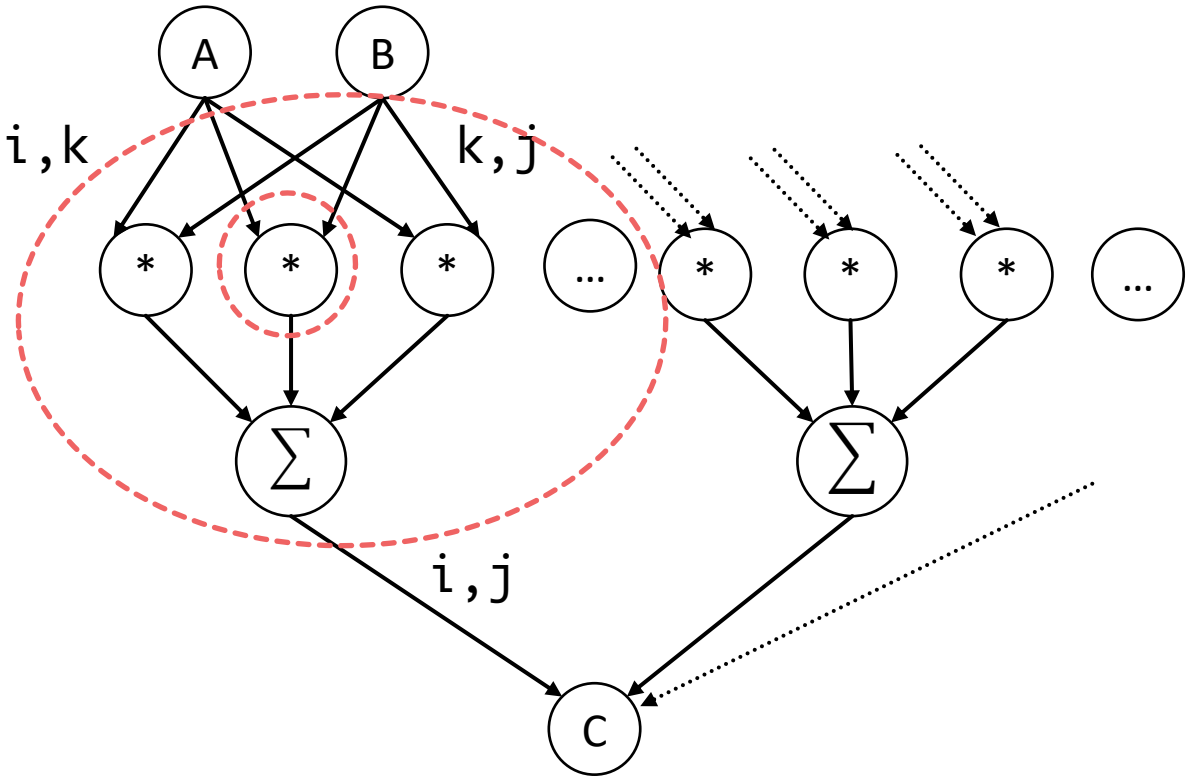
Example: Matrix Multiplication

Nested repeating subgraphs



Instantiation

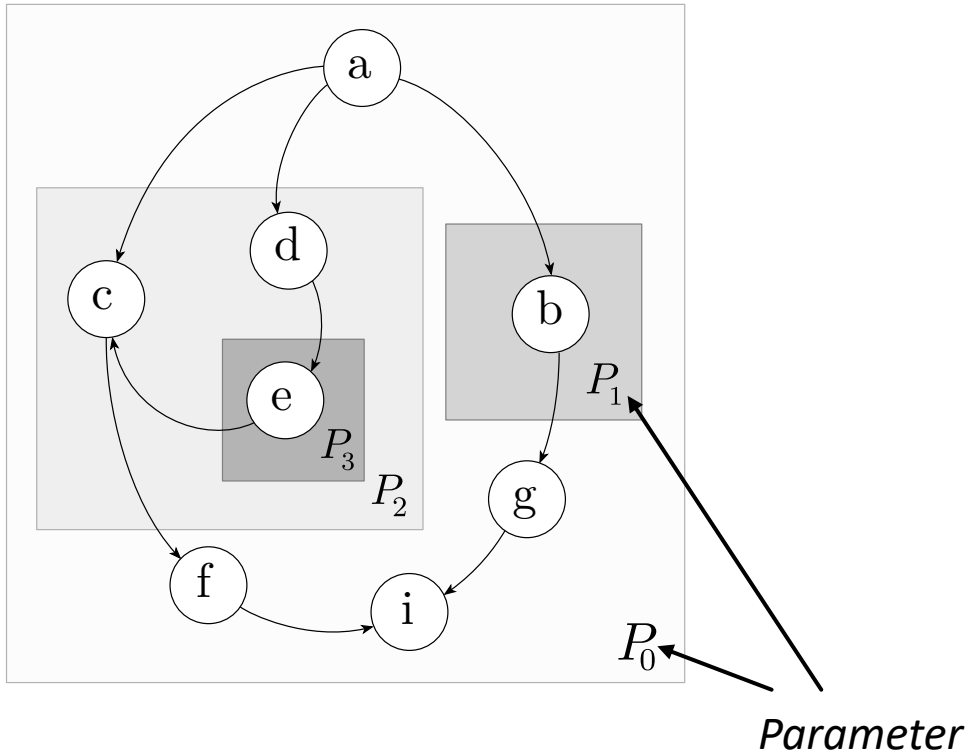
Fixes k,m,n



Parametric Graph Template

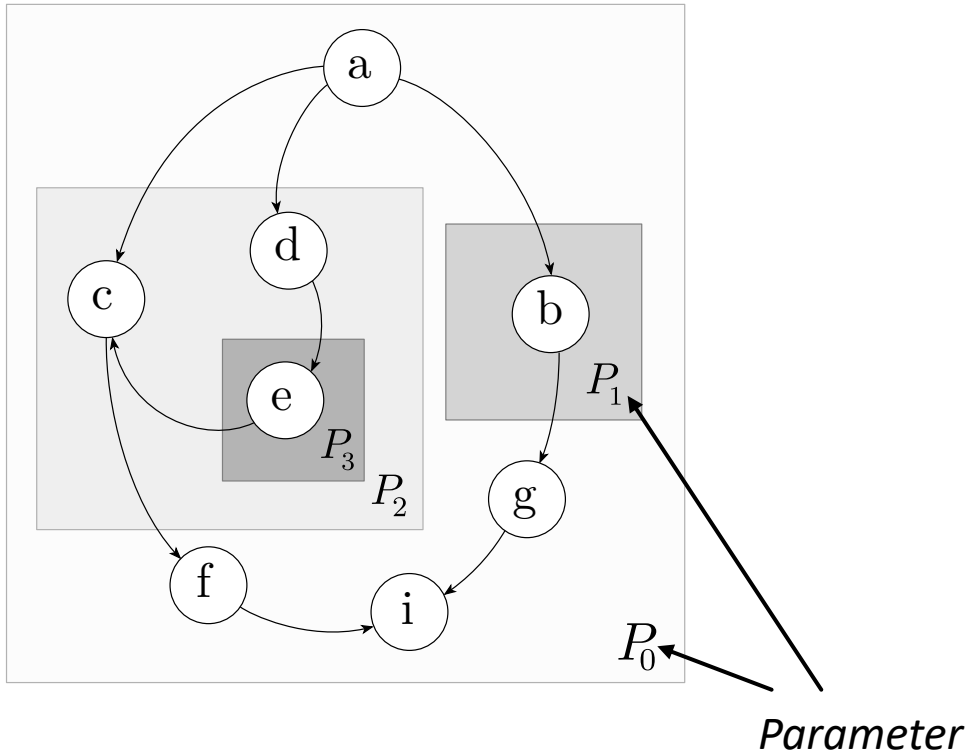
Dataflow Graph

Nested Templates

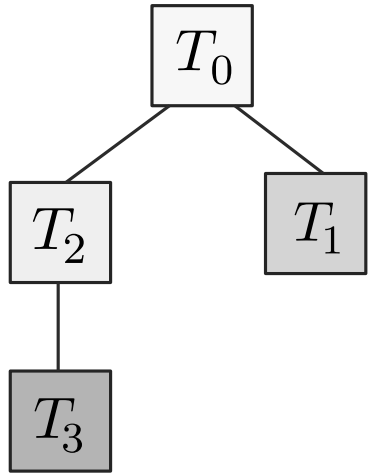


Parametric Graph Template

Nested Templates



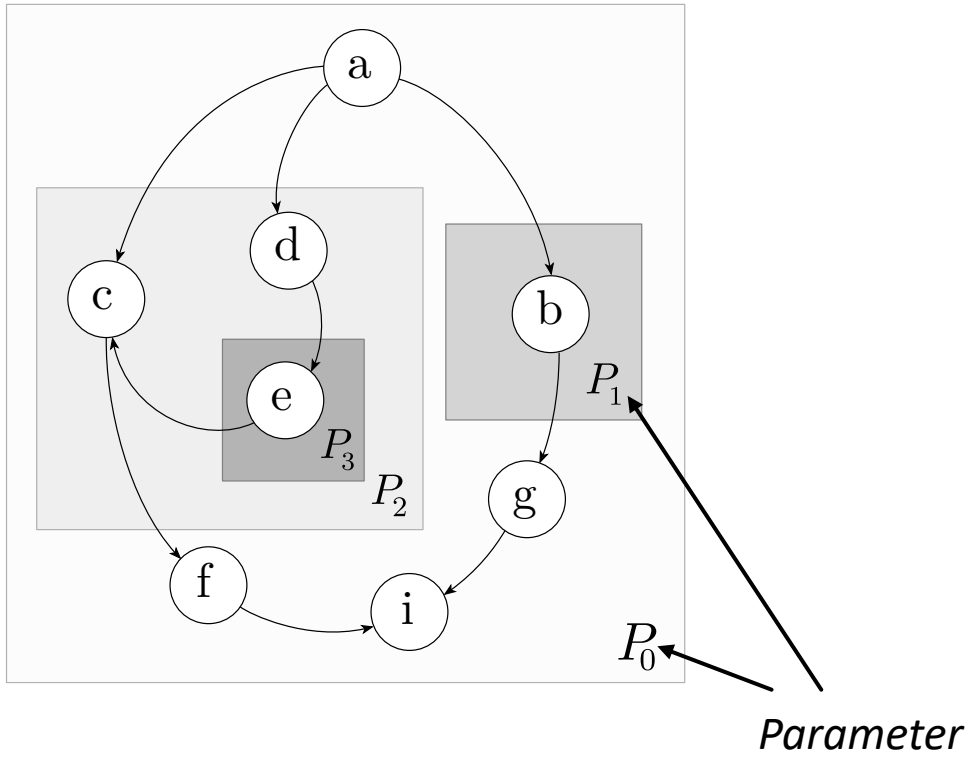
Parametric Graph Template



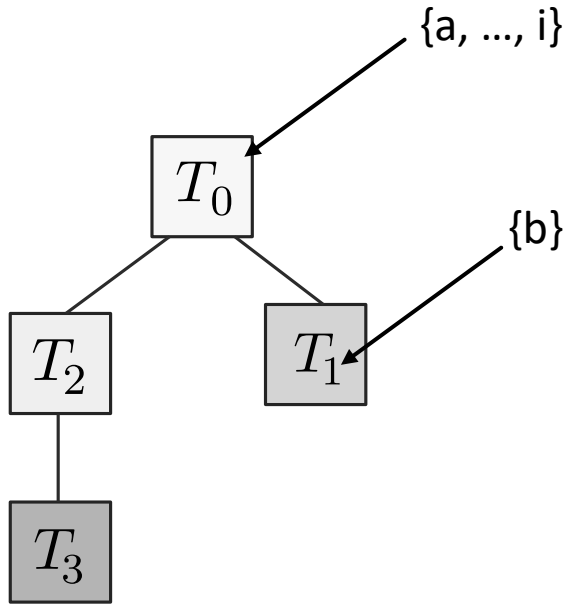
Template Tree

Nested Templates

Template is a subset of vertices

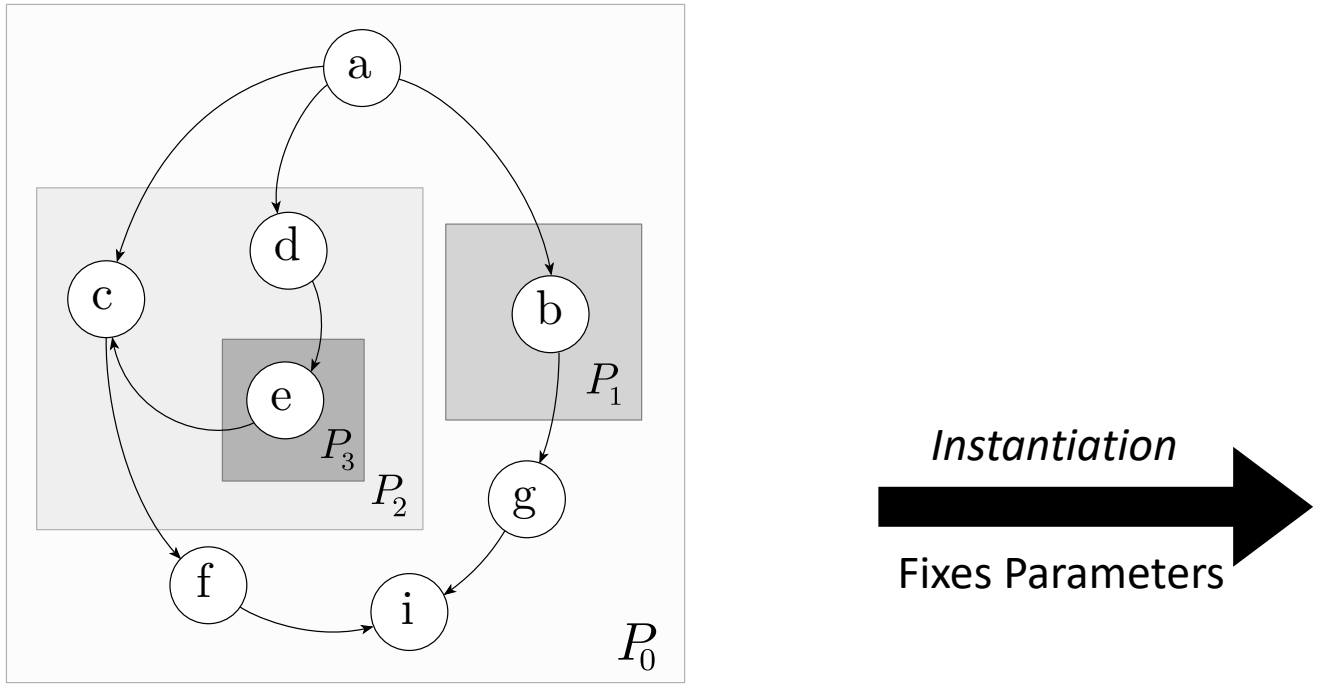


Parametric Graph Template



Template Tree

Instantiation

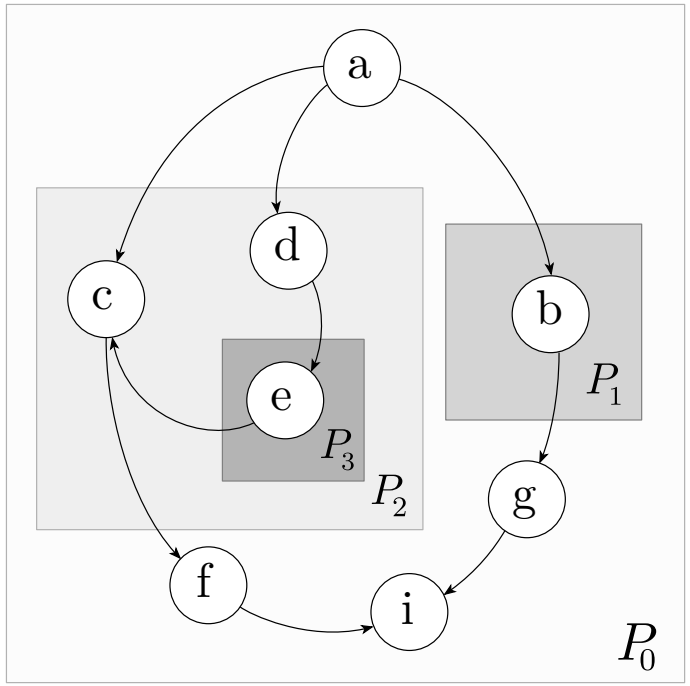


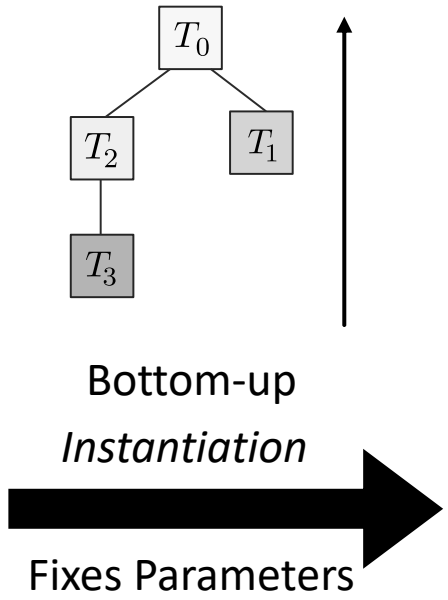
Instantiation

Fixes Parameters

Parametric Graph Template

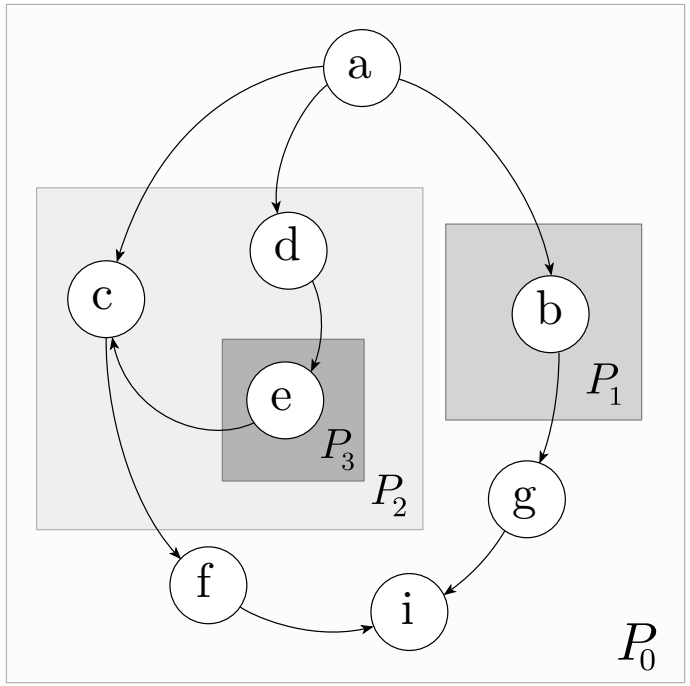
Instantiation

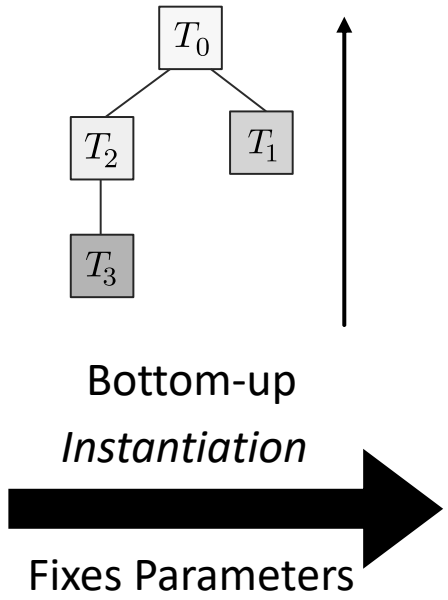


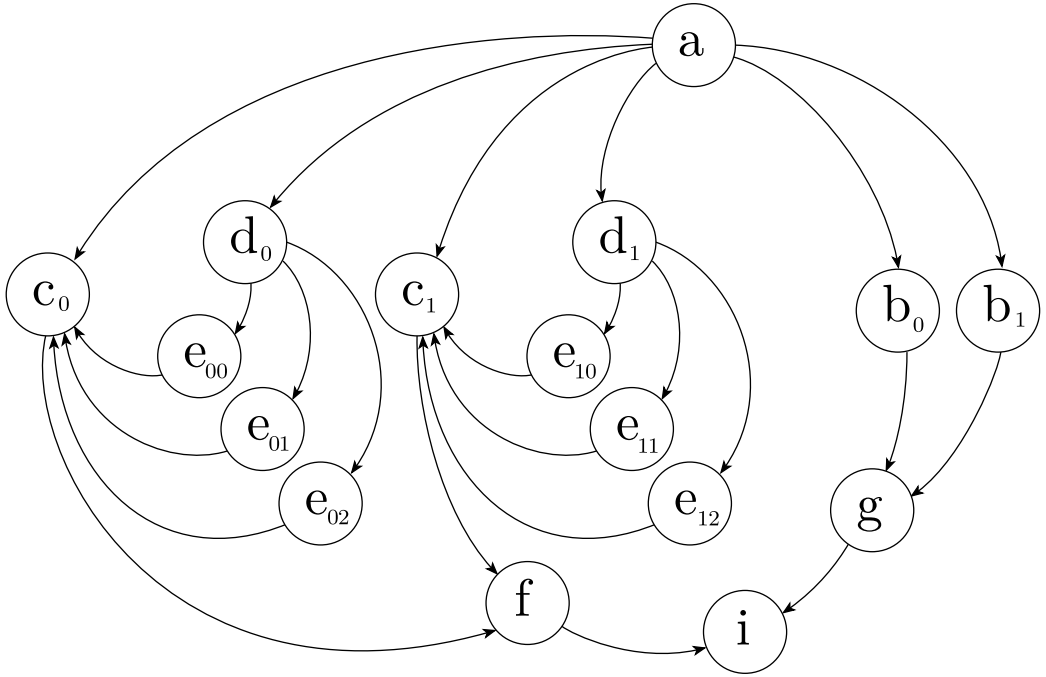


Parametric Graph Template

Instantiation



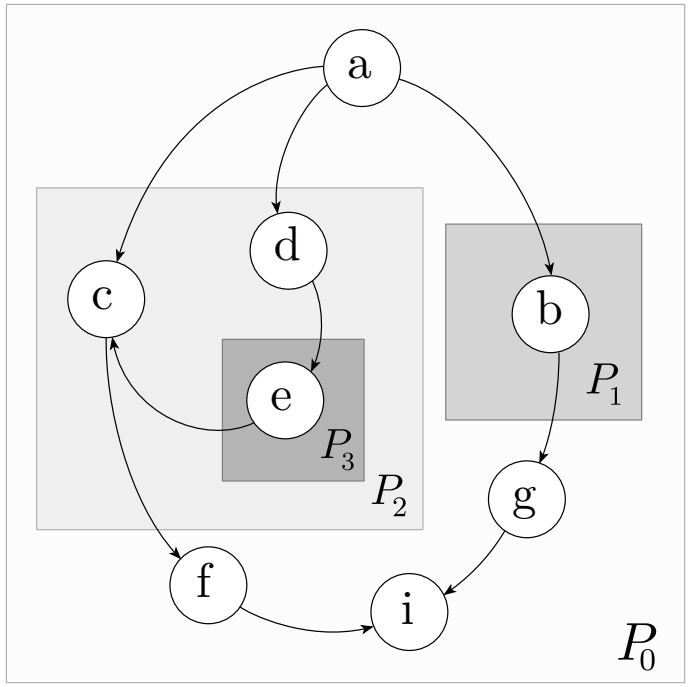


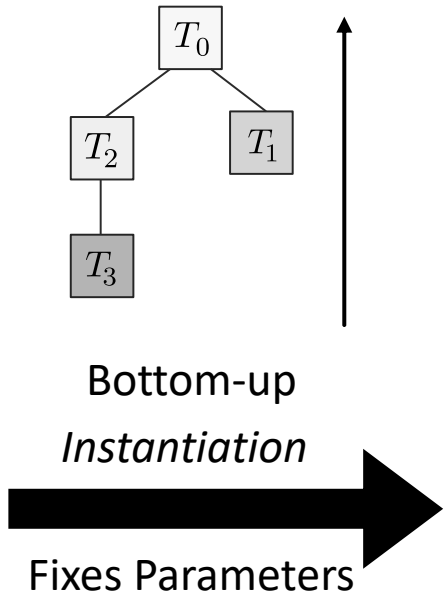


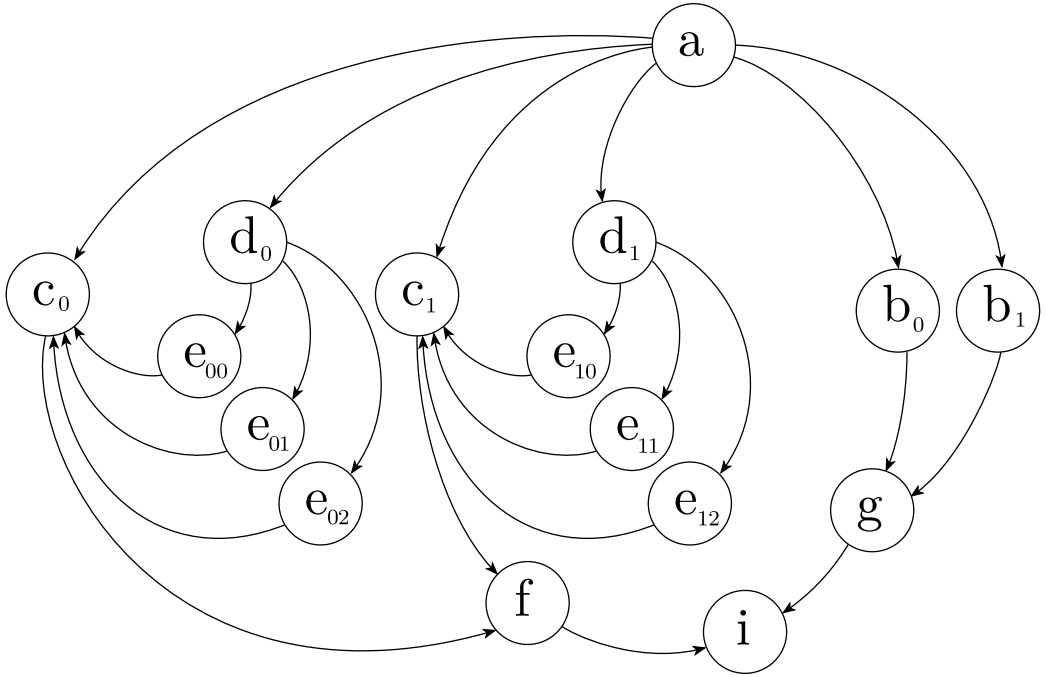
Parametric Graph Template

Instantiated Graph

Instantiation





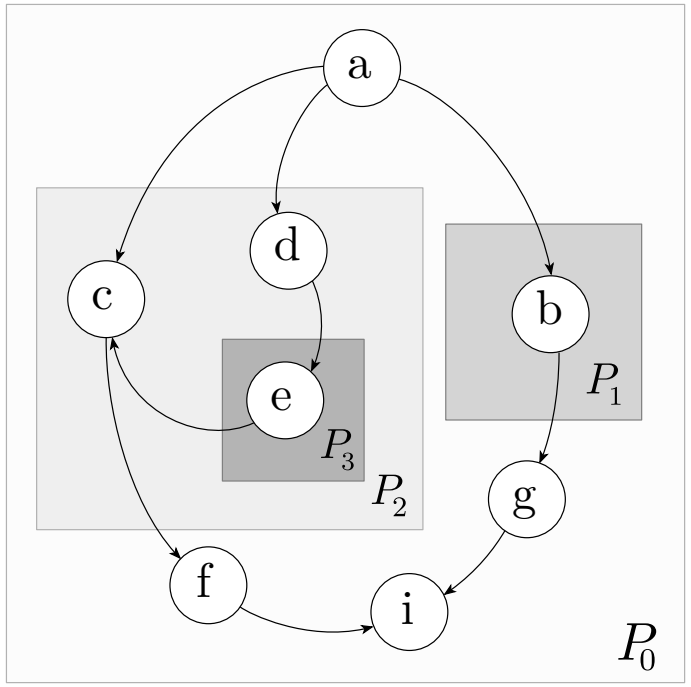


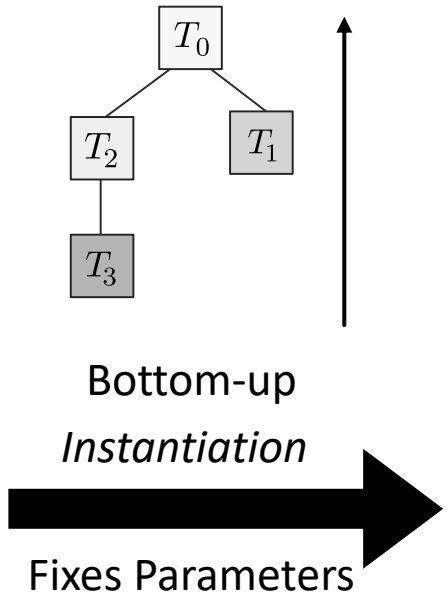
Parametric Graph Template

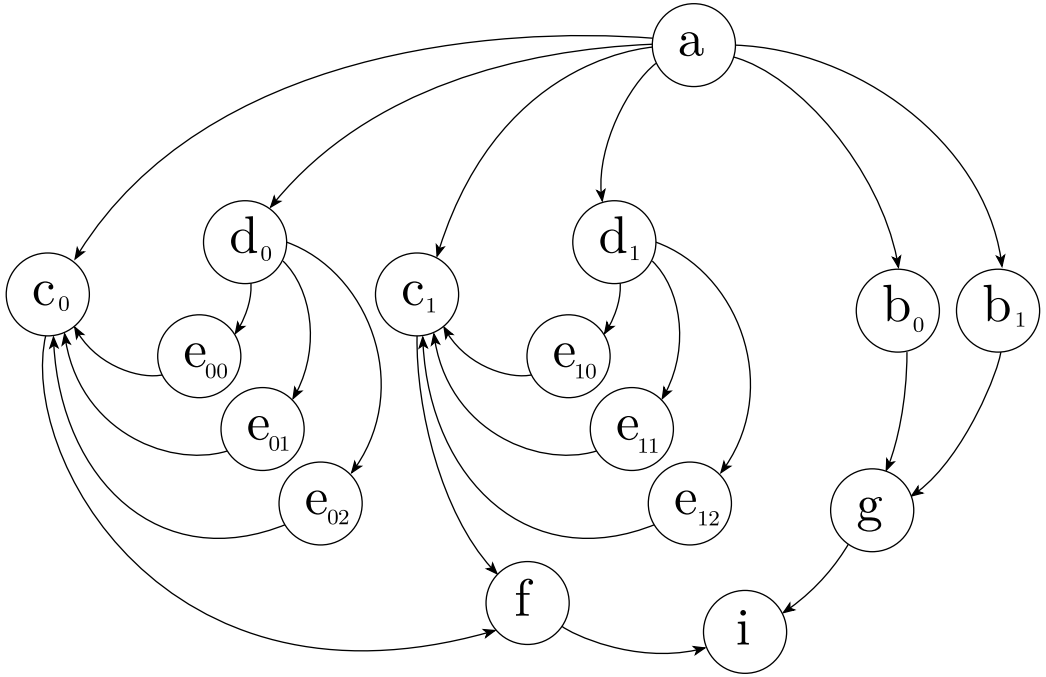
Instantiated Graph

Result can have exponential size in input size!

Instantiation







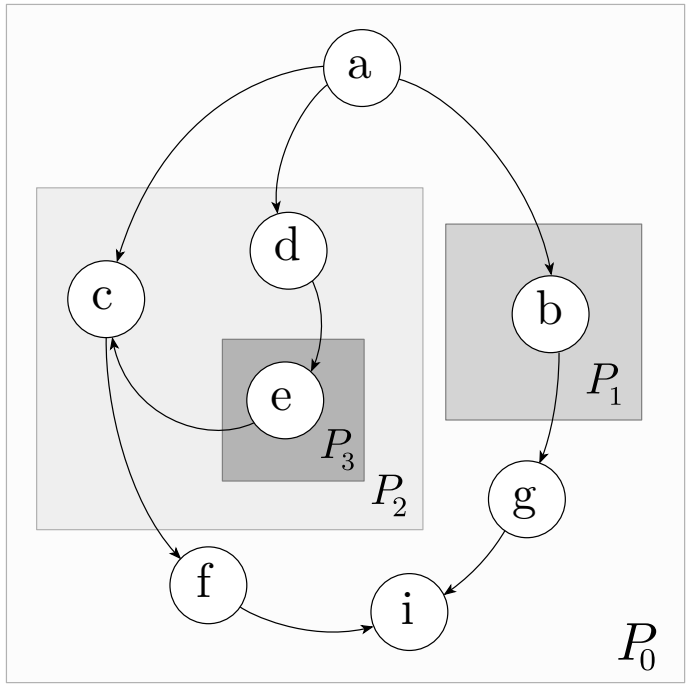
Parametric Graph Template

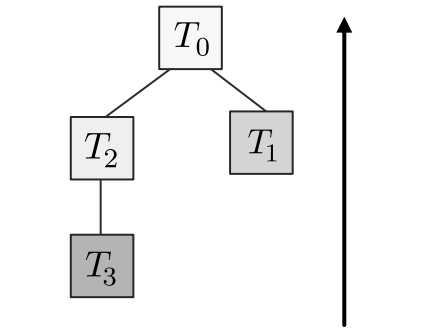
Goal: polynomial time algorithms in template graph's size

Instantiated Graph

Result can have exponential size in input size!

Instantiation

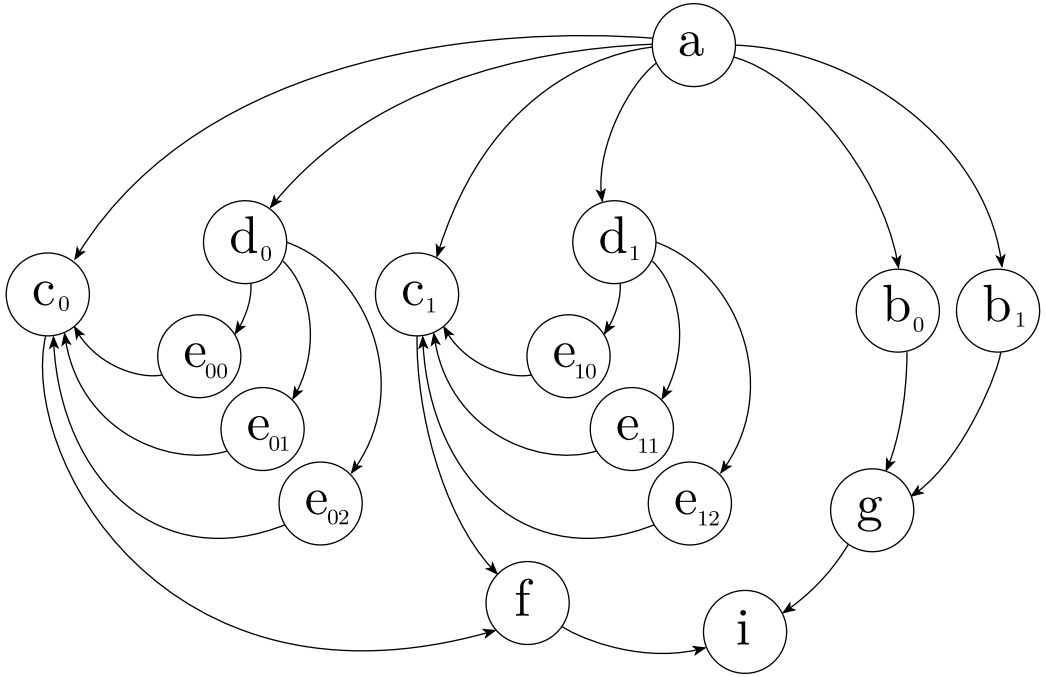




Bottom-up

Instantiation

Fixes Parameters



Parametric Graph Template

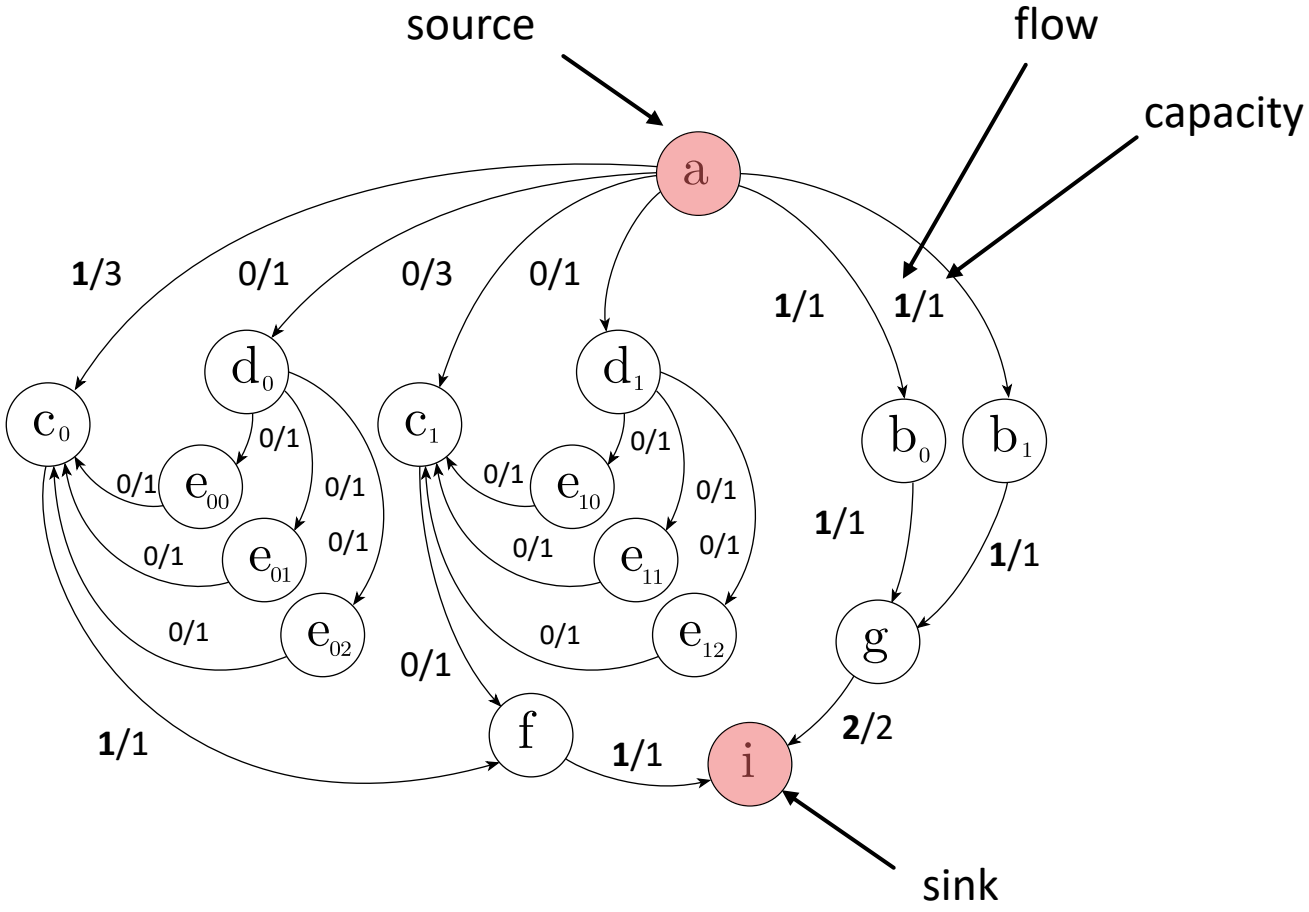
Goal: polynomial time algorithms in template graph's size

We show this for maximum flow

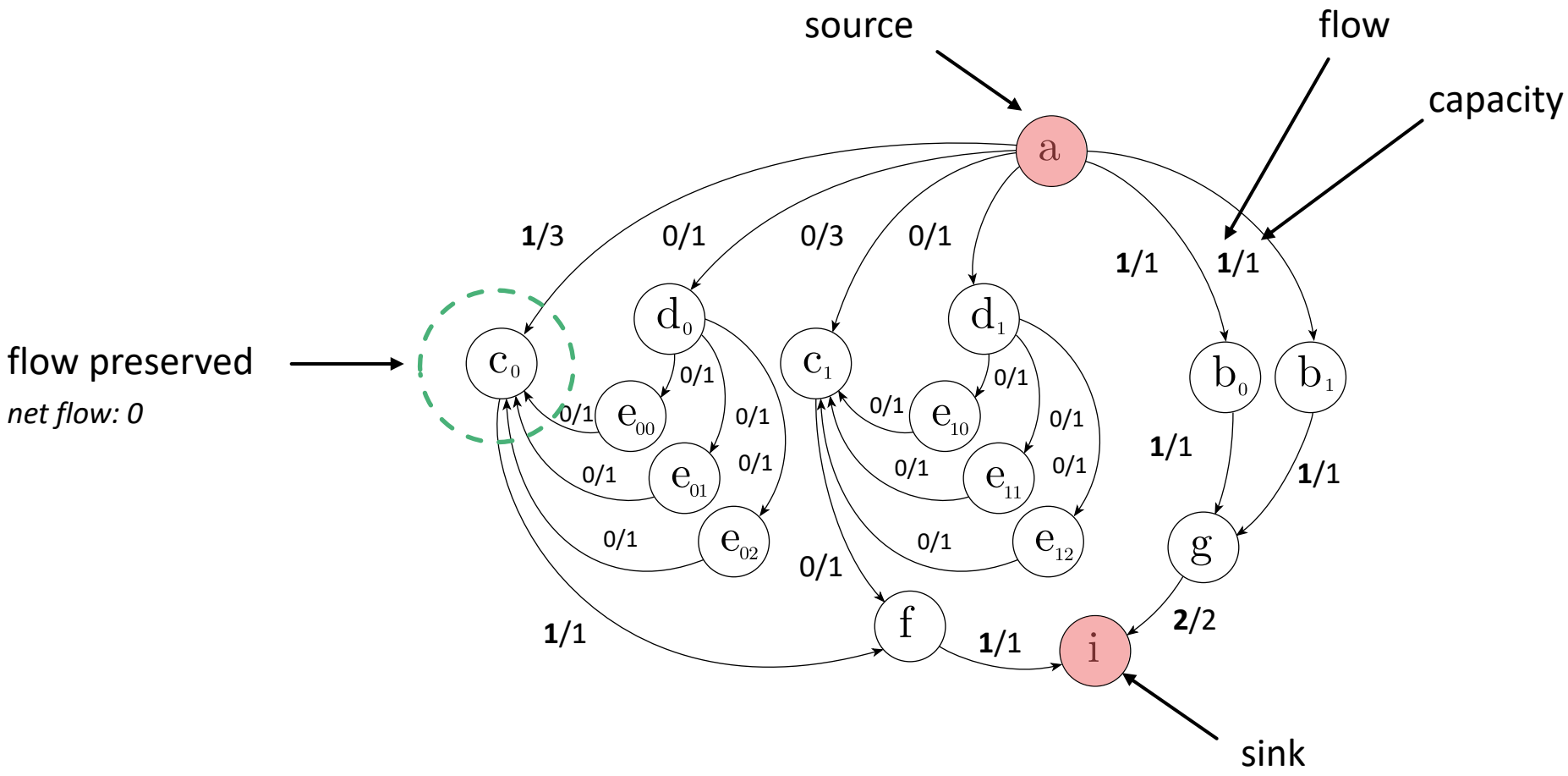
Instantiated Graph

Result can have exponential size in input size!

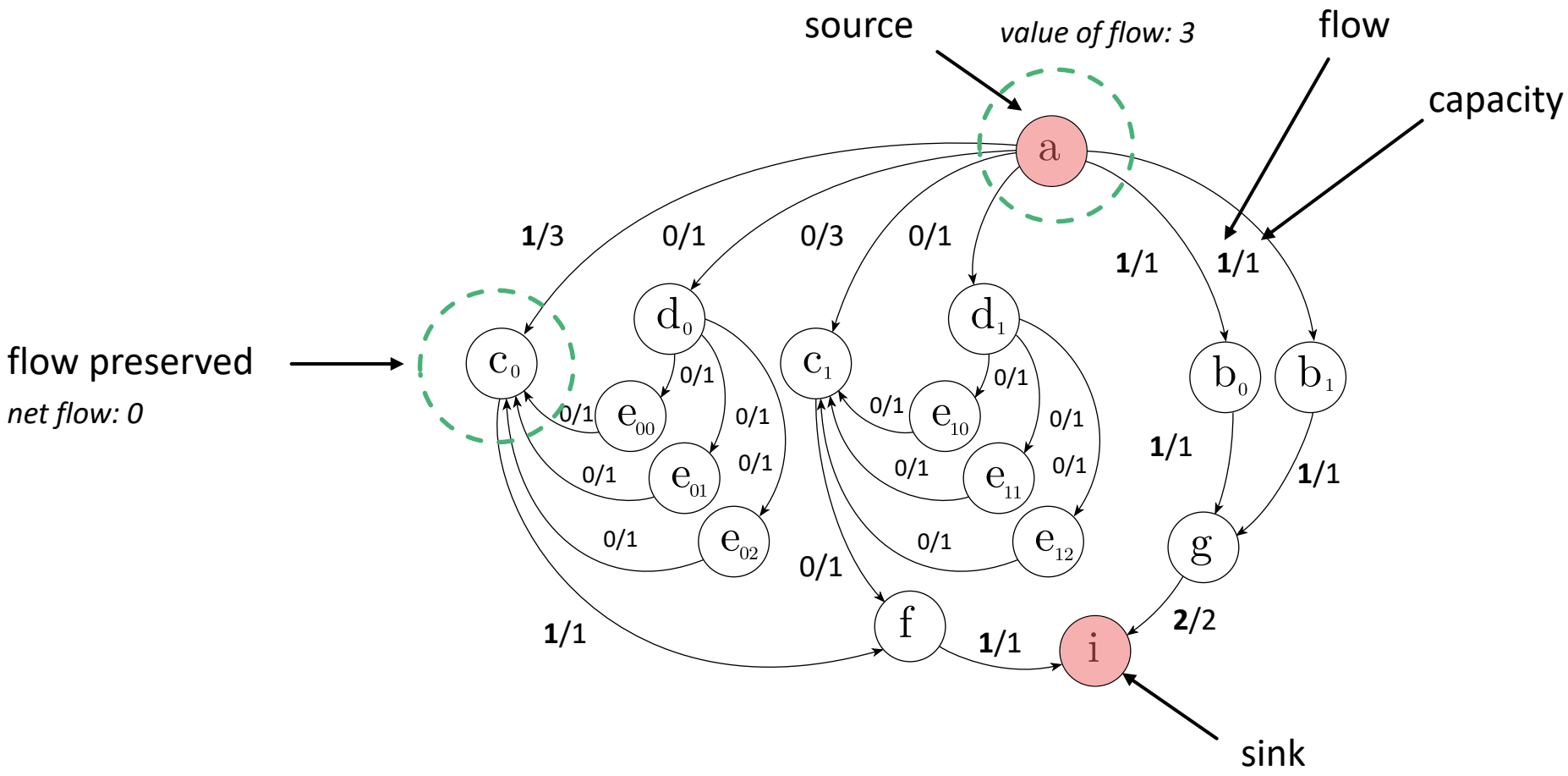
Maximum Flow



Maximum Flow

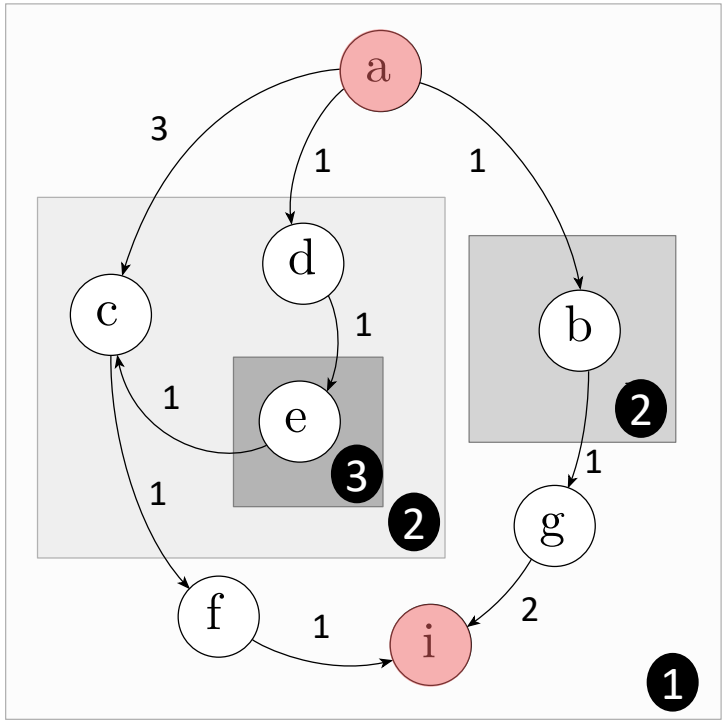


Maximum Flow



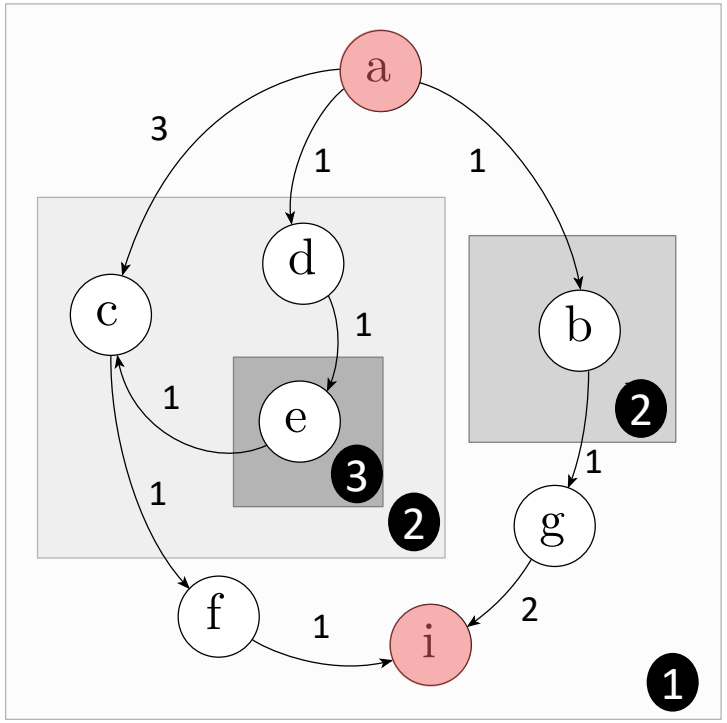
Template Maximum Flow

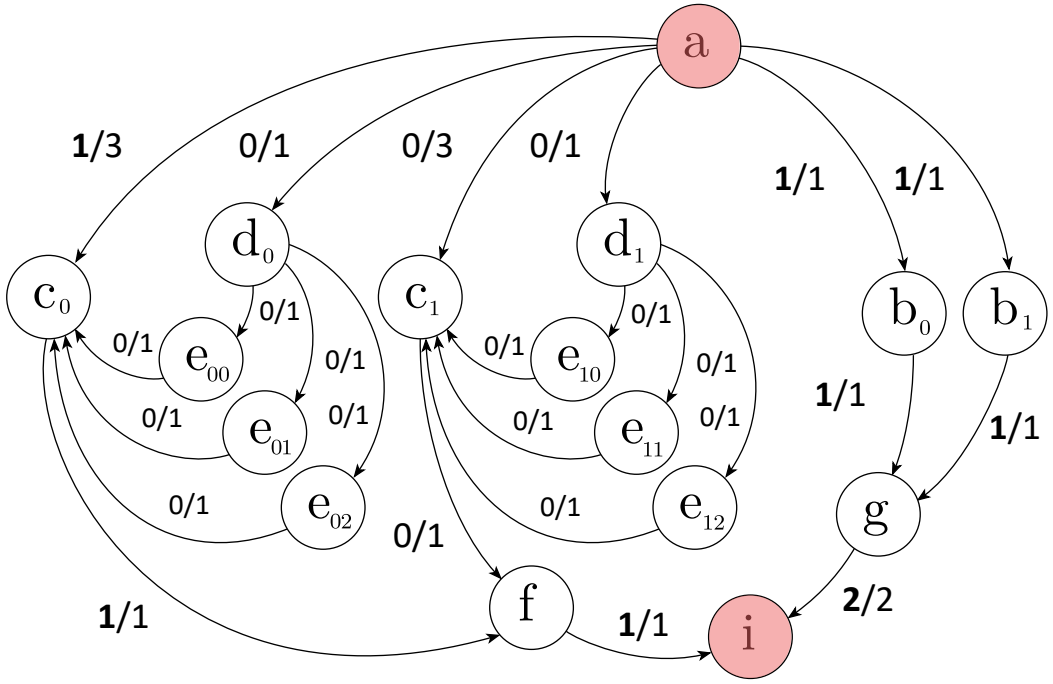
Source and sink are
<u>both</u> in the root template



Template Maximum Flow

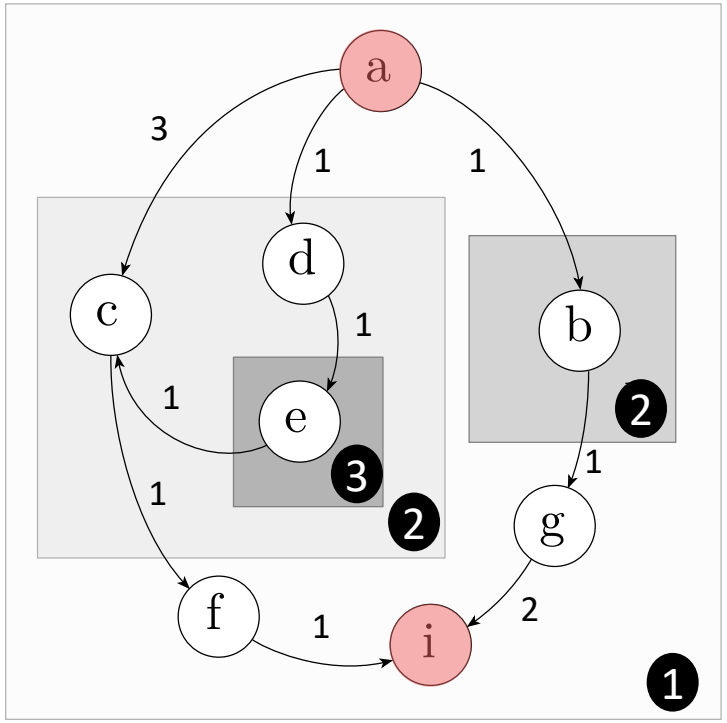
Source and sink are
<u>both</u> in the root template

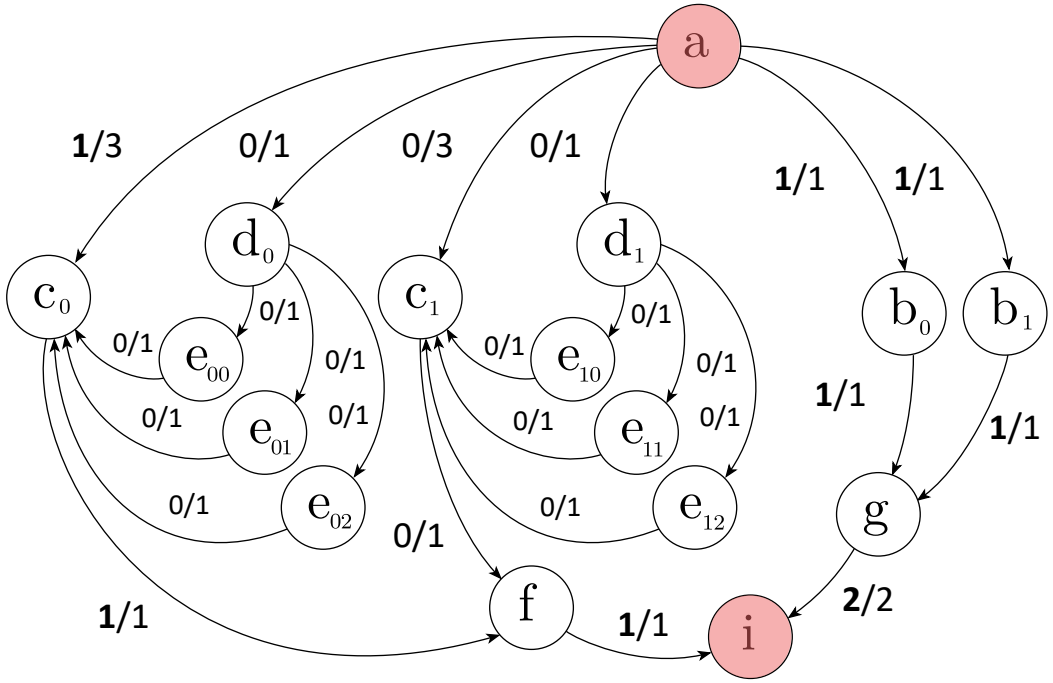




Template Maximum Flow

Source and sink are
<u>both</u> in the root template

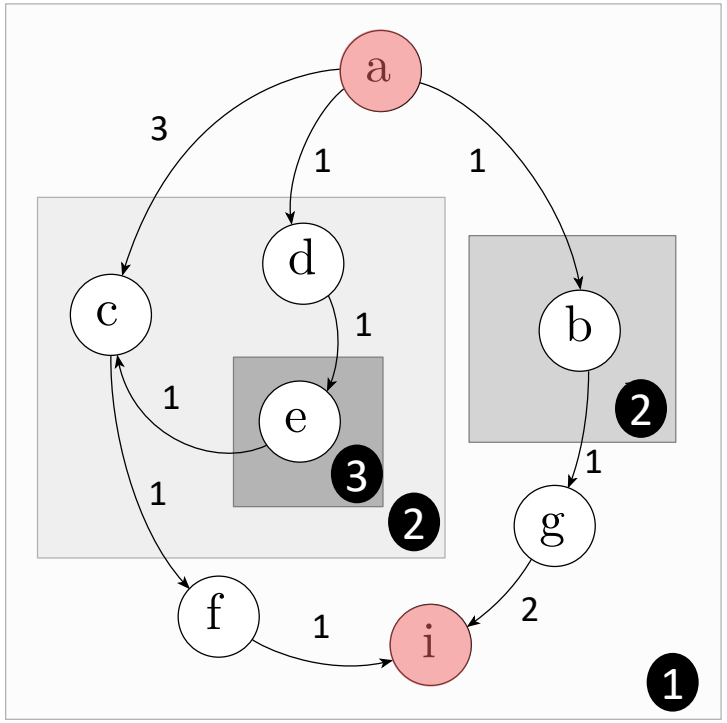




Goal: Solve without fully instantiating

Edge Reweighting

Source and sink are
<u>both</u> in the root template

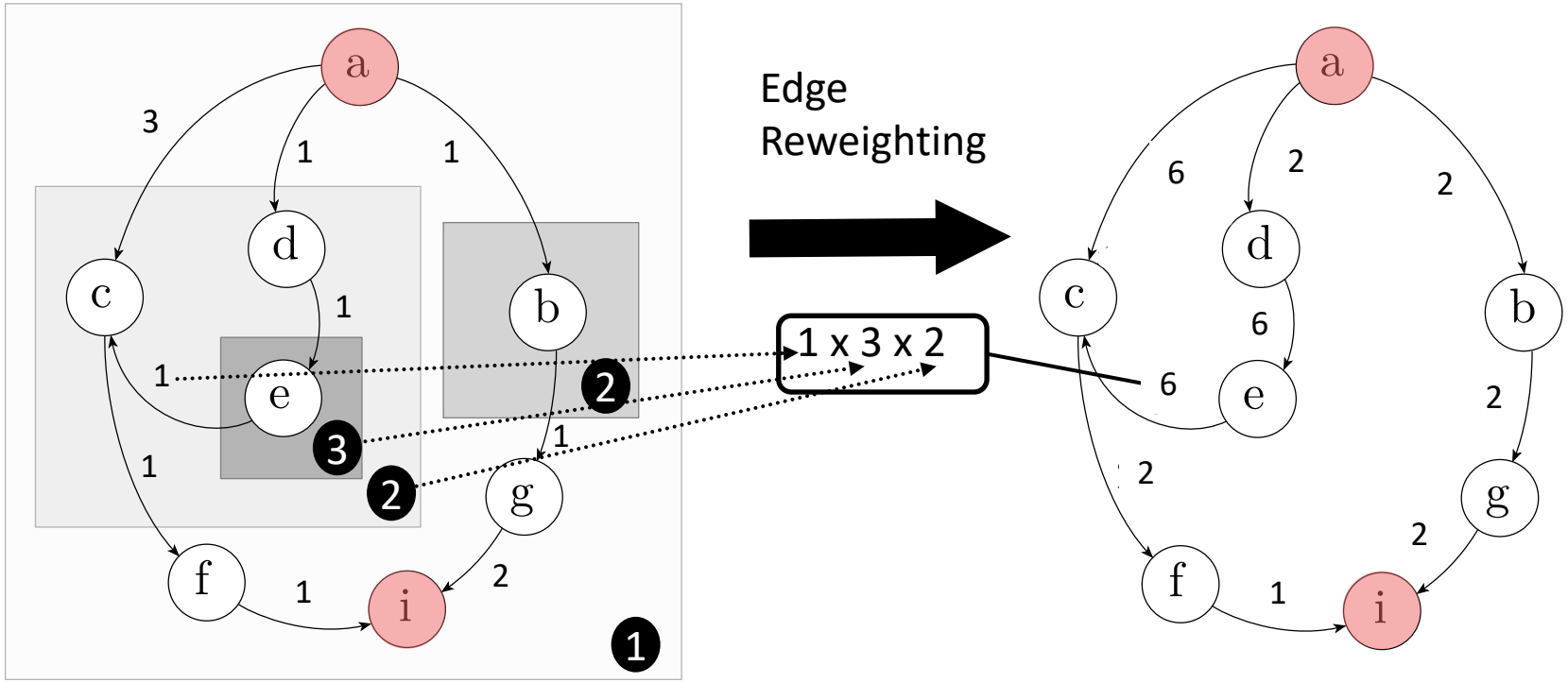


Edge
Reweighting

Intuition:
Capacity ∝ Replications

Edge Reweighting

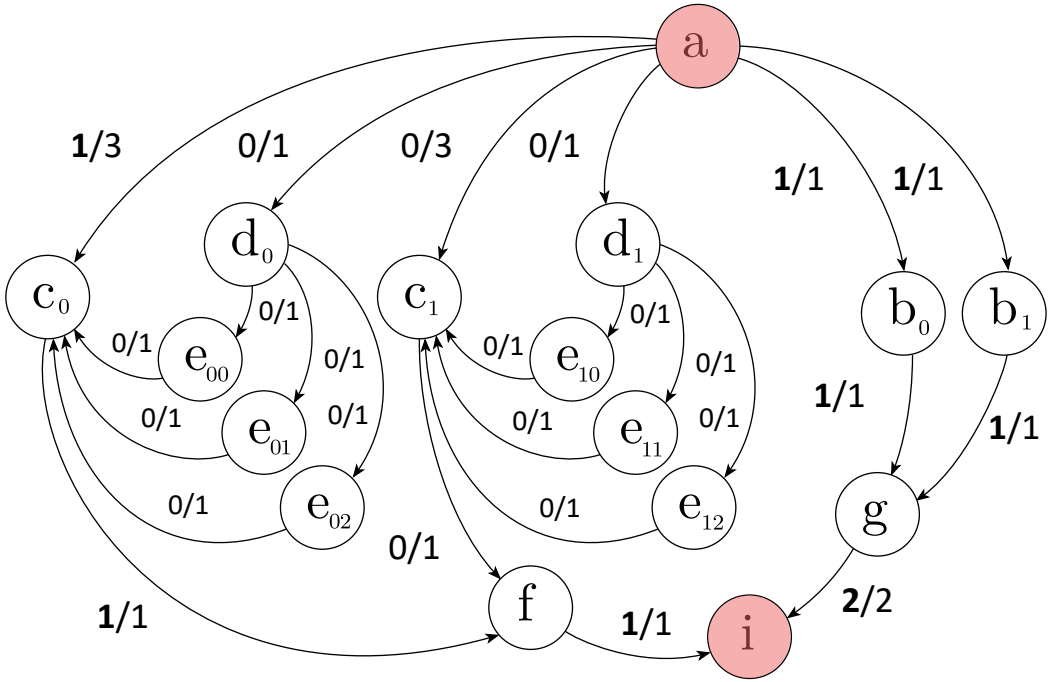
Source and sink are
<u>both</u> in the root template

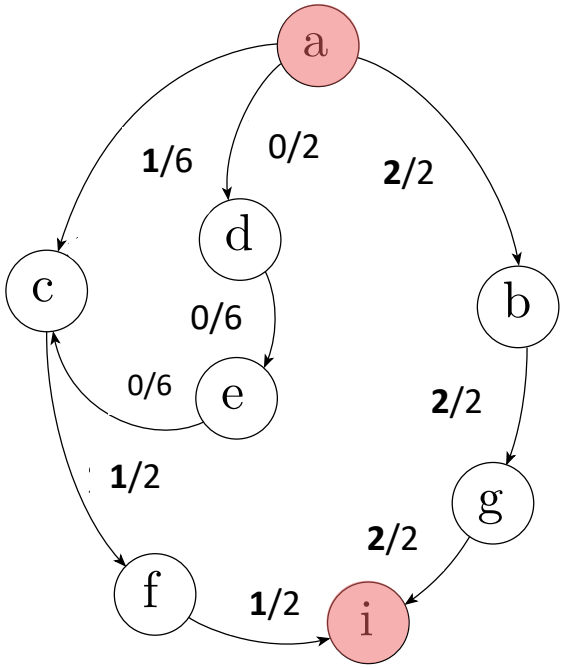


Intuition:
Capacity ∝ Replications

Edge Reweighting

Source and sink are
<u>both</u> in the root template



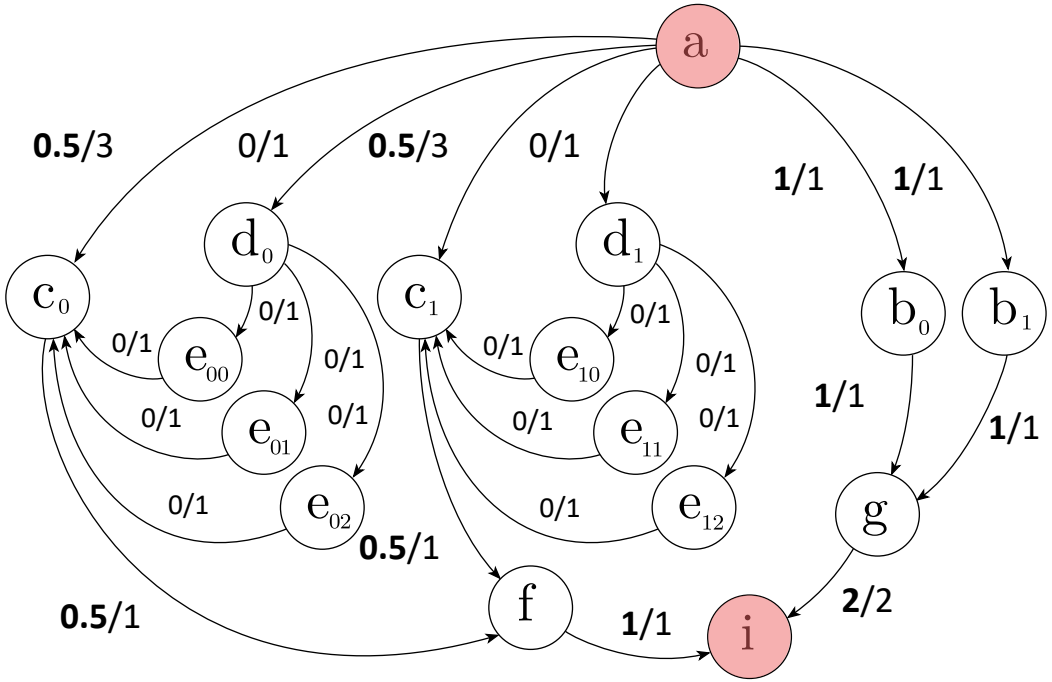


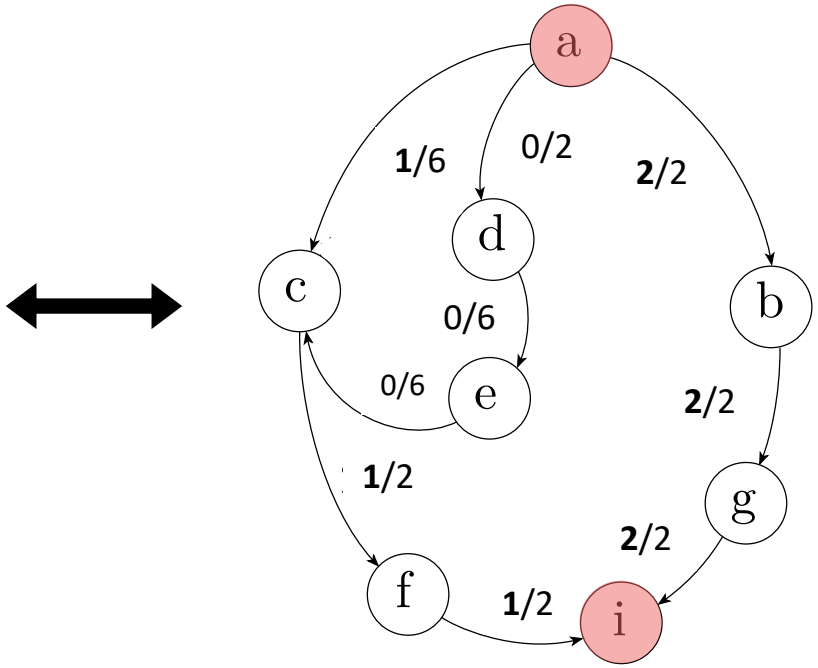
Maximum Flow is preserved

Proof uses strong duality

Edge Reweighting

Source and sink are
<u>both</u> in the root template



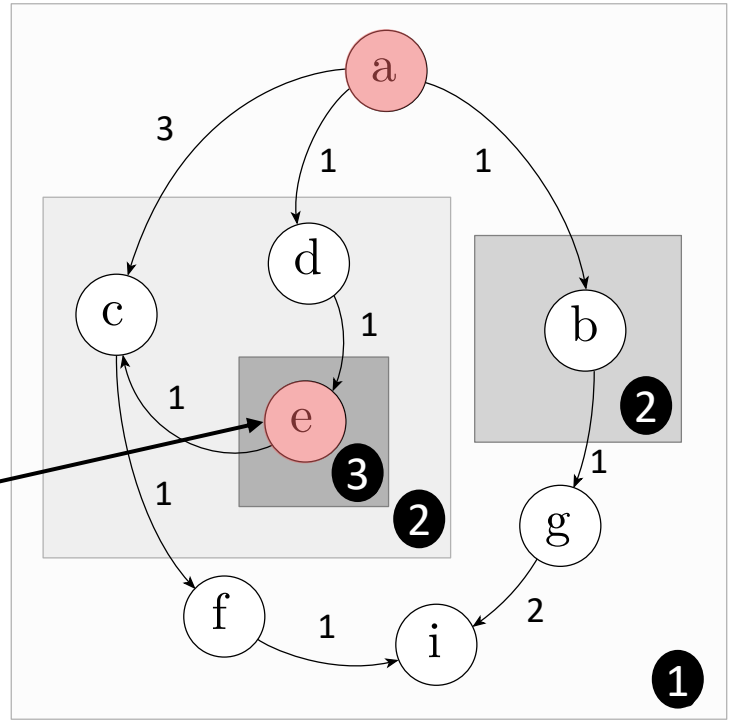


Maximum Flow is preserved

Proof uses strong duality

General Case

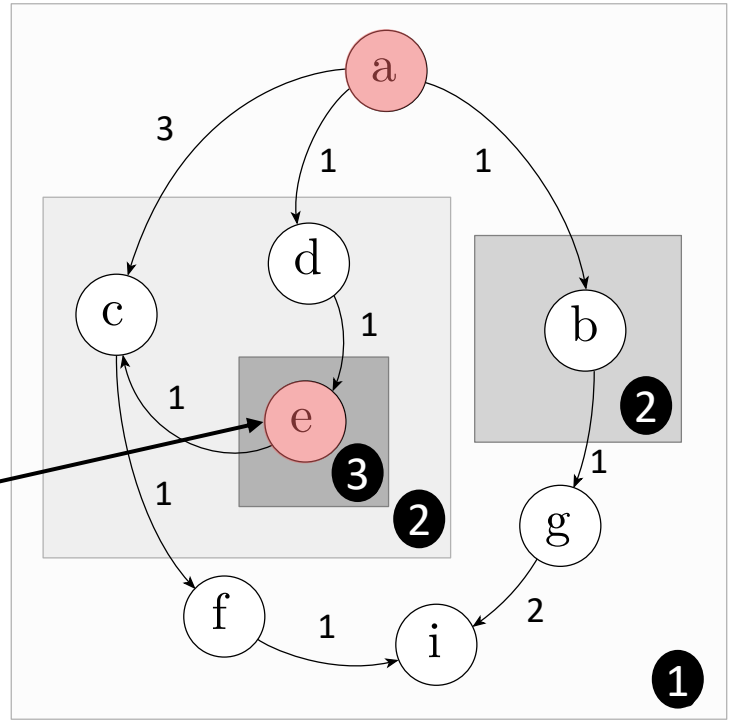
Source and sink
not both in the root template?

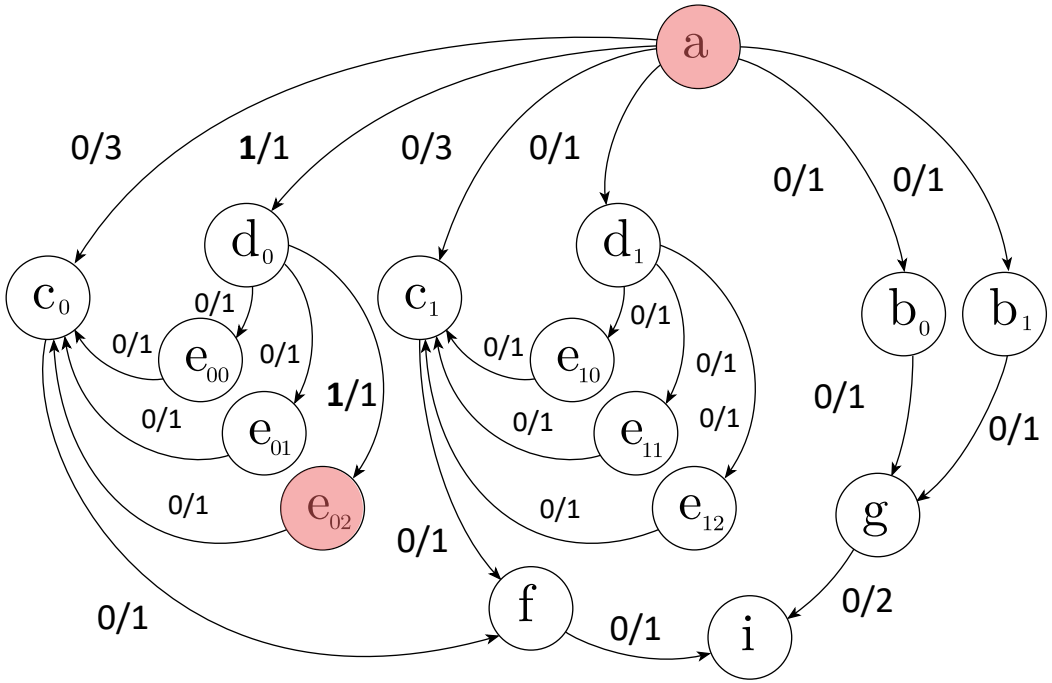


Sink is *one* instance

General Case

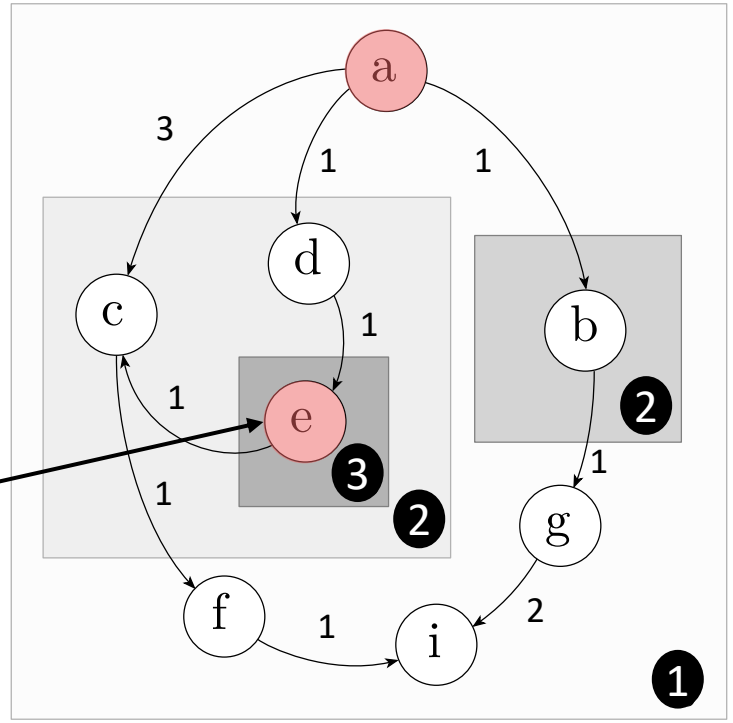
Source and sink
not both in the root template?



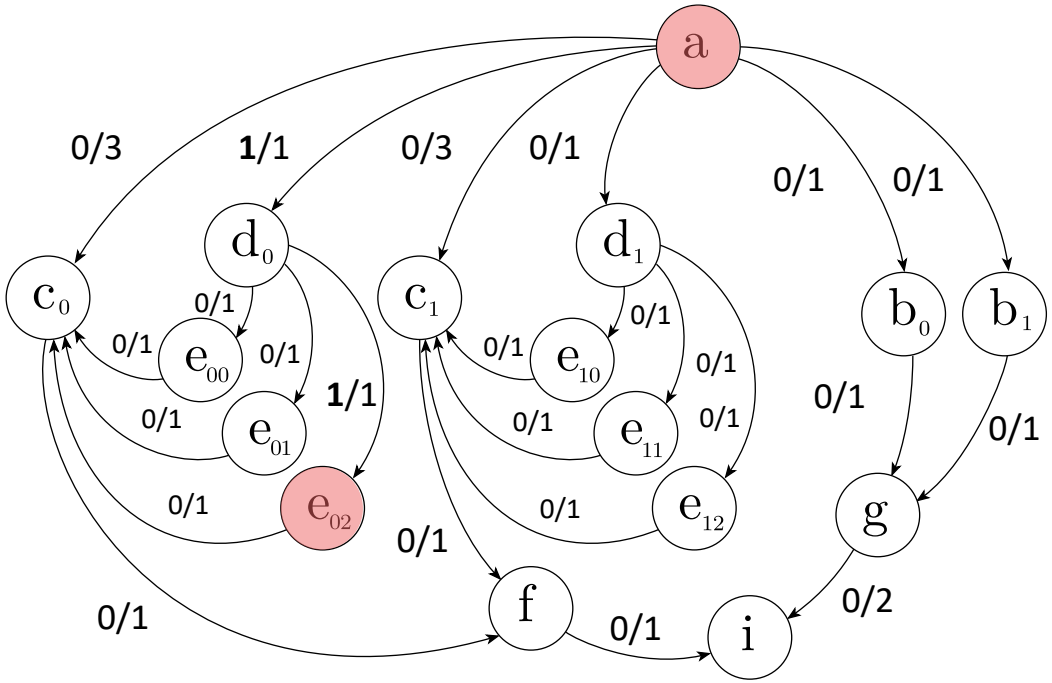


General Case

Source and sink
not both in the root template?

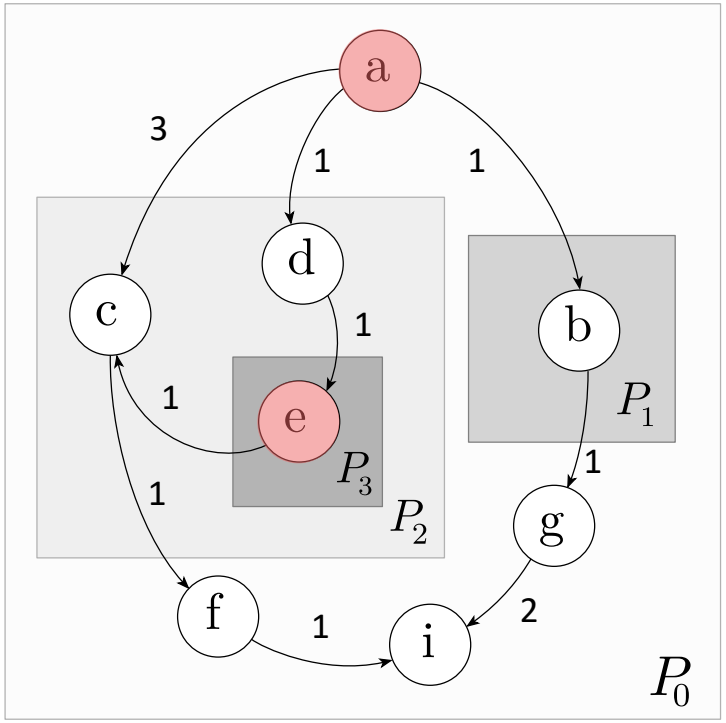


Sink is *one* instance



Capacity & Replications

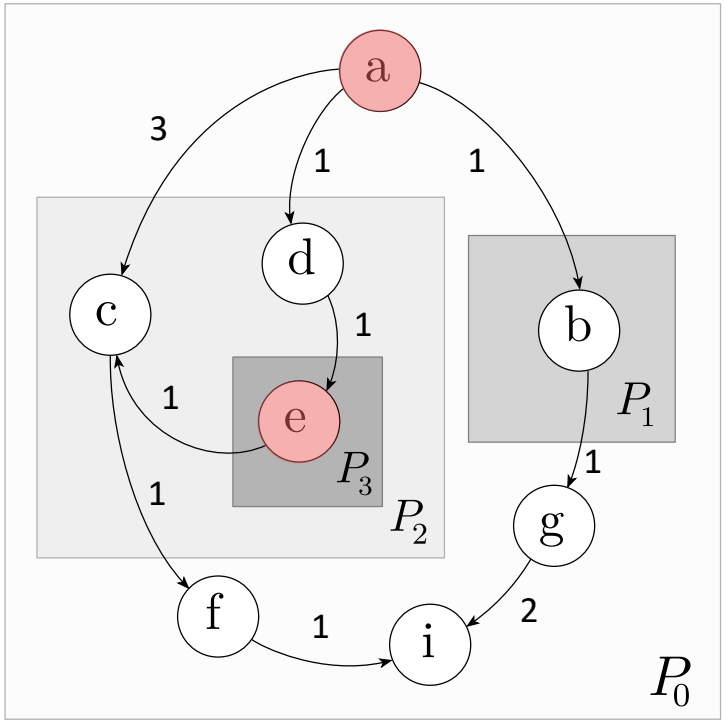
Partial Instantiation

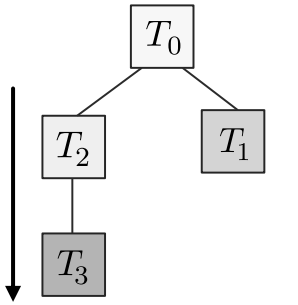


Partial Instantiation

Reduce to situation where source and sink are in the root

Partial Instantiation



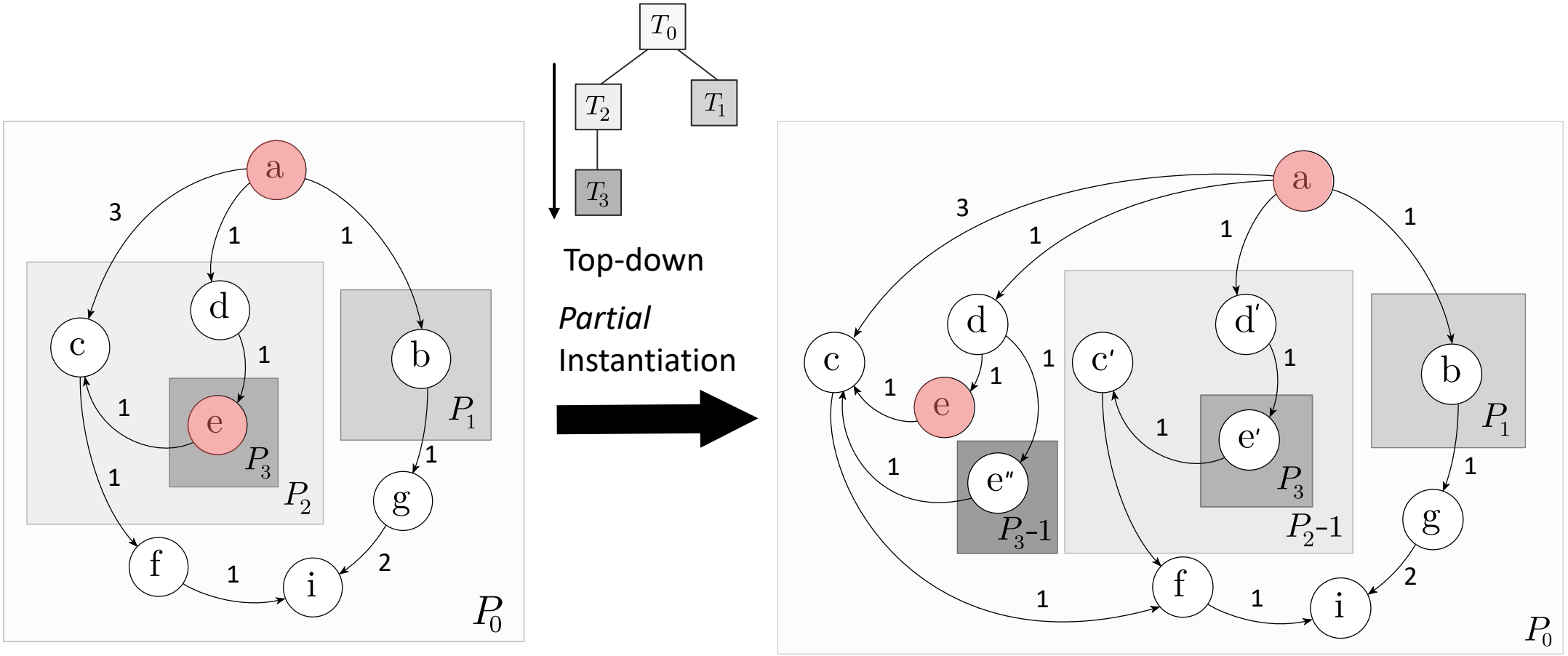


Top-down

Partial Instantiation

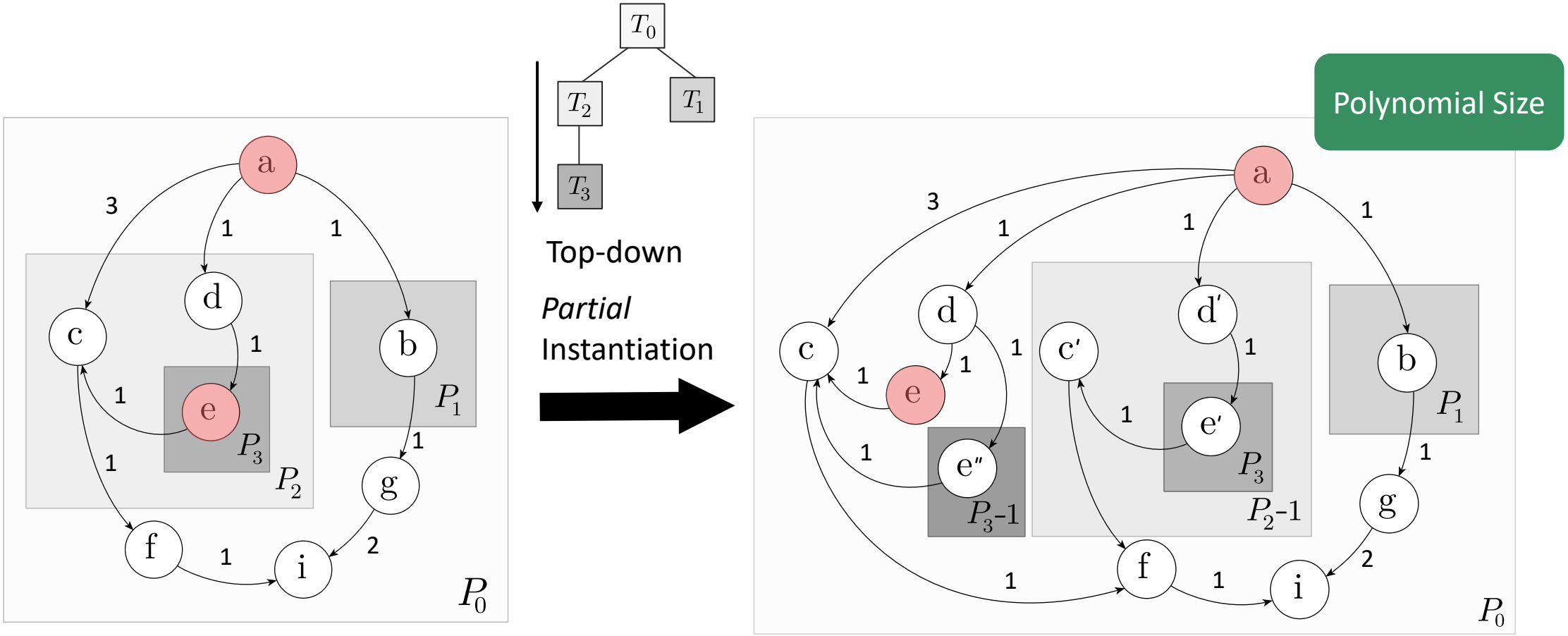
Reduce to situation where source and sink are in the root

Partial Instantiation



Reduce to situation where source and sink are in the root

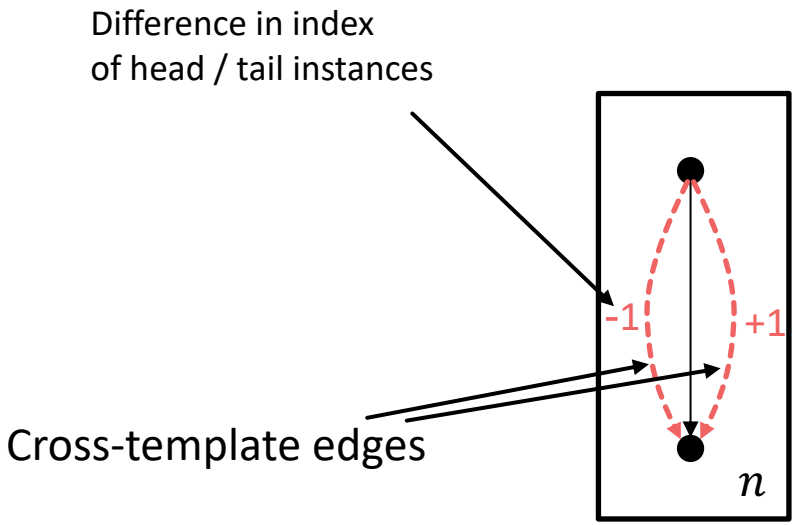
Partial Instantiation

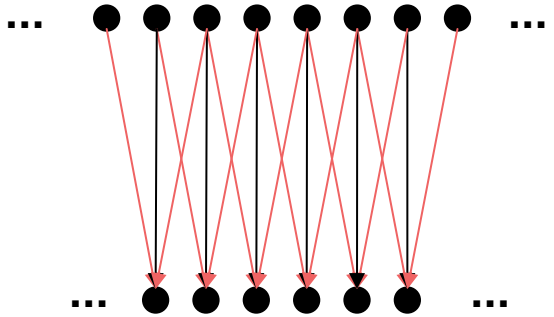


Reduce to situation where source and sink are in the root

Cross-Template Edges

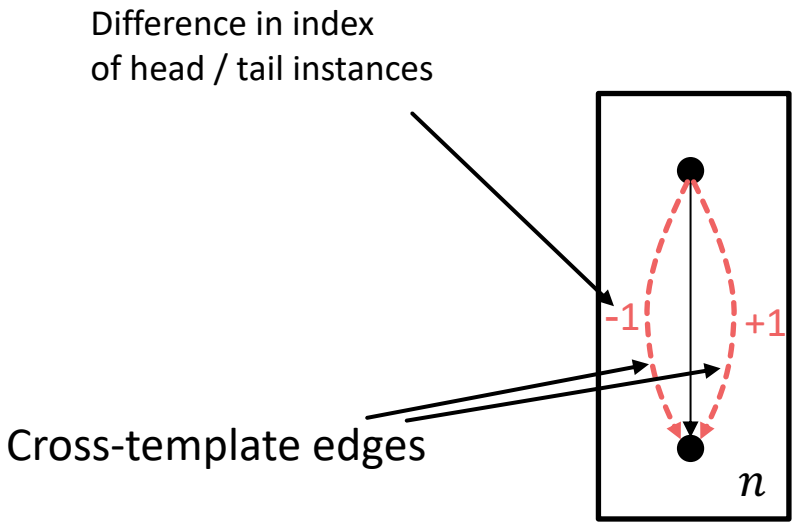
Example: 1-Dimensional Stencil

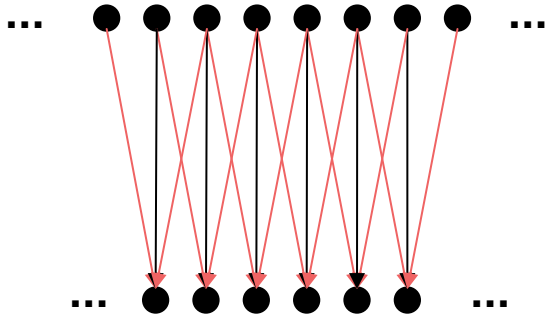




Cross-Template Edges

Example: 1-Dimensional Stencil

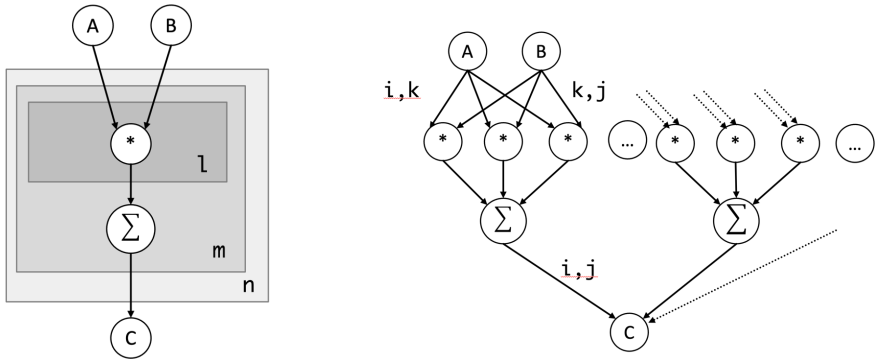




Maximum flow results still apply

Parametric Graph Templates

Nested Dataflow Representation



youtube.com/@spcl 150+ Talks

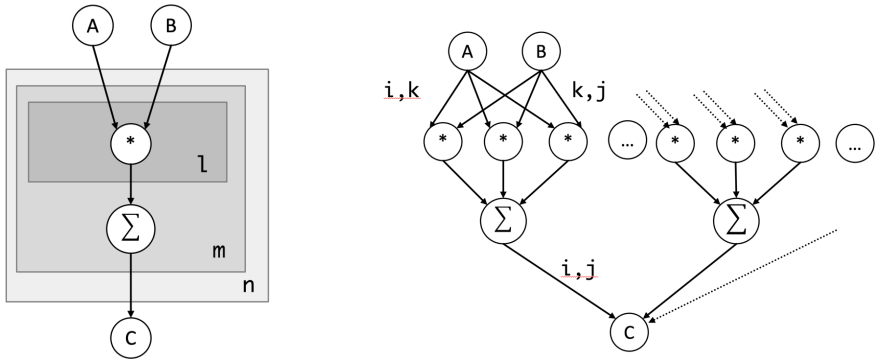
twitter.com/spcl_eth 1.2K+ Followers

github.com/spcl 2K+ Stars

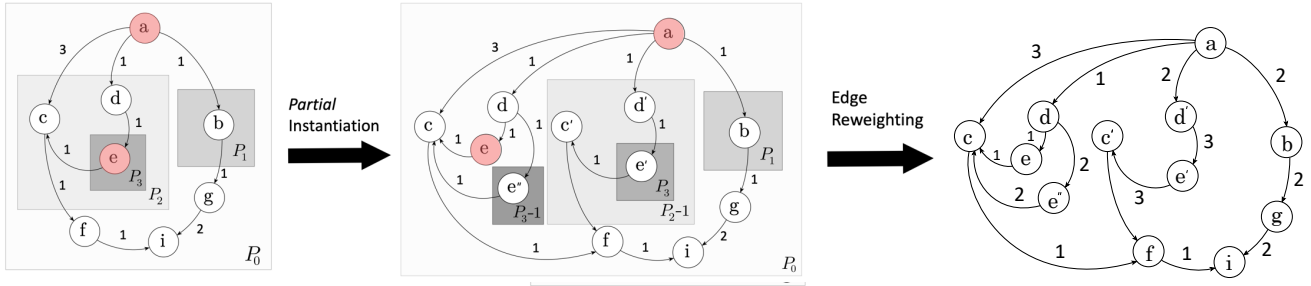
... or spcl.inf.ethz.ch

Parametric Graph Templates

Nested Dataflow Representation



Efficient Maximum Flow Algorithm



youtube.com/@spcl 150+ Talks

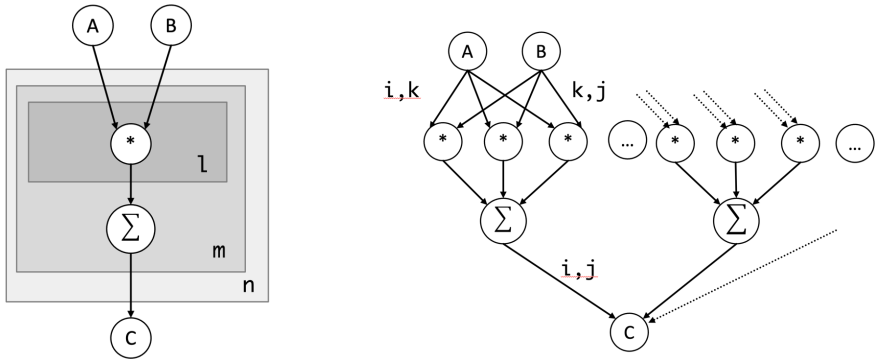
twitter.com/spcl_eth 1.2K+ Followers

github.com/spcl 2K+ Stars

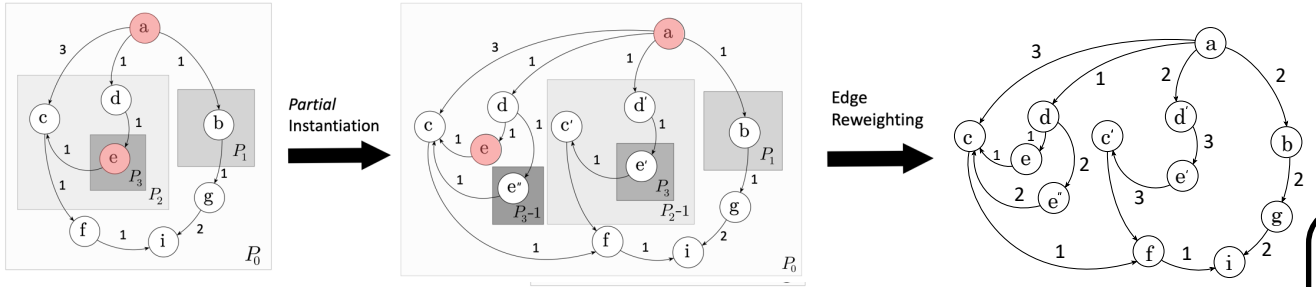
… or spcl.inf.ethz.ch

Parametric Graph Templates

Nested Dataflow Representation



Efficient Maximum Flow Algorithm



youtube.com/@spcl 150+ Talks

twitter.com/spcl_eth 1.2K+ Followers

github.com/spcl 2K+ Stars

… or spcl.inf.ethz.ch

… other problems?