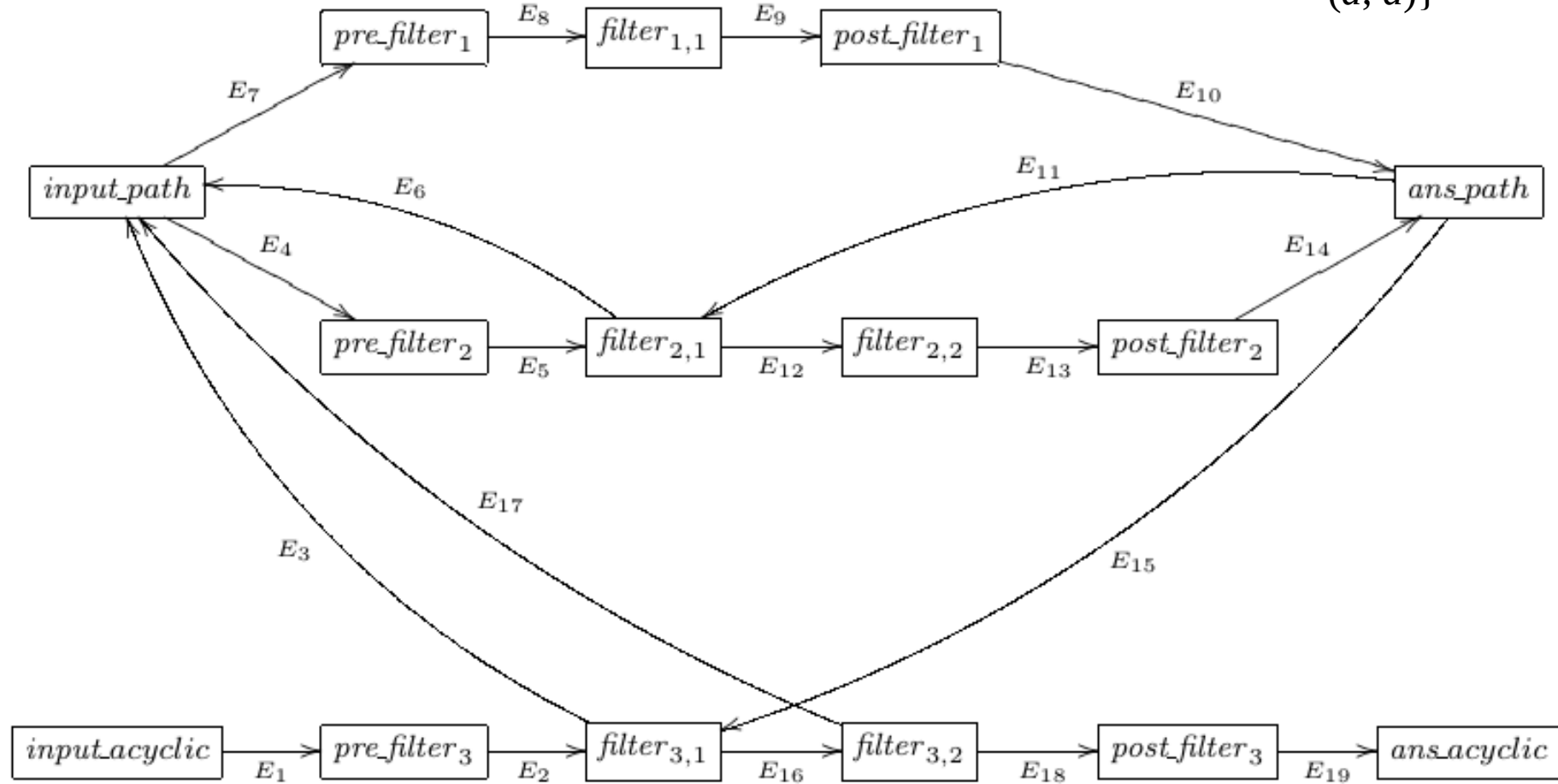
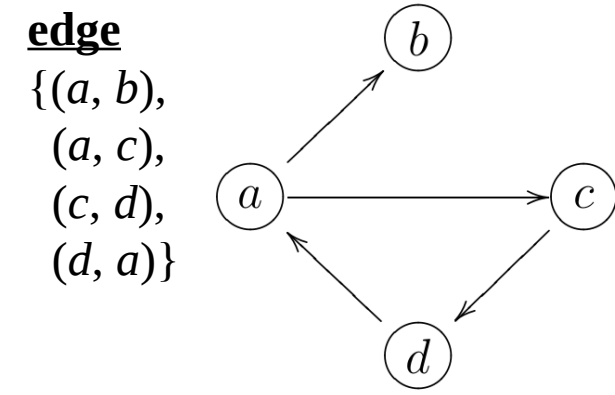


Online Appendix

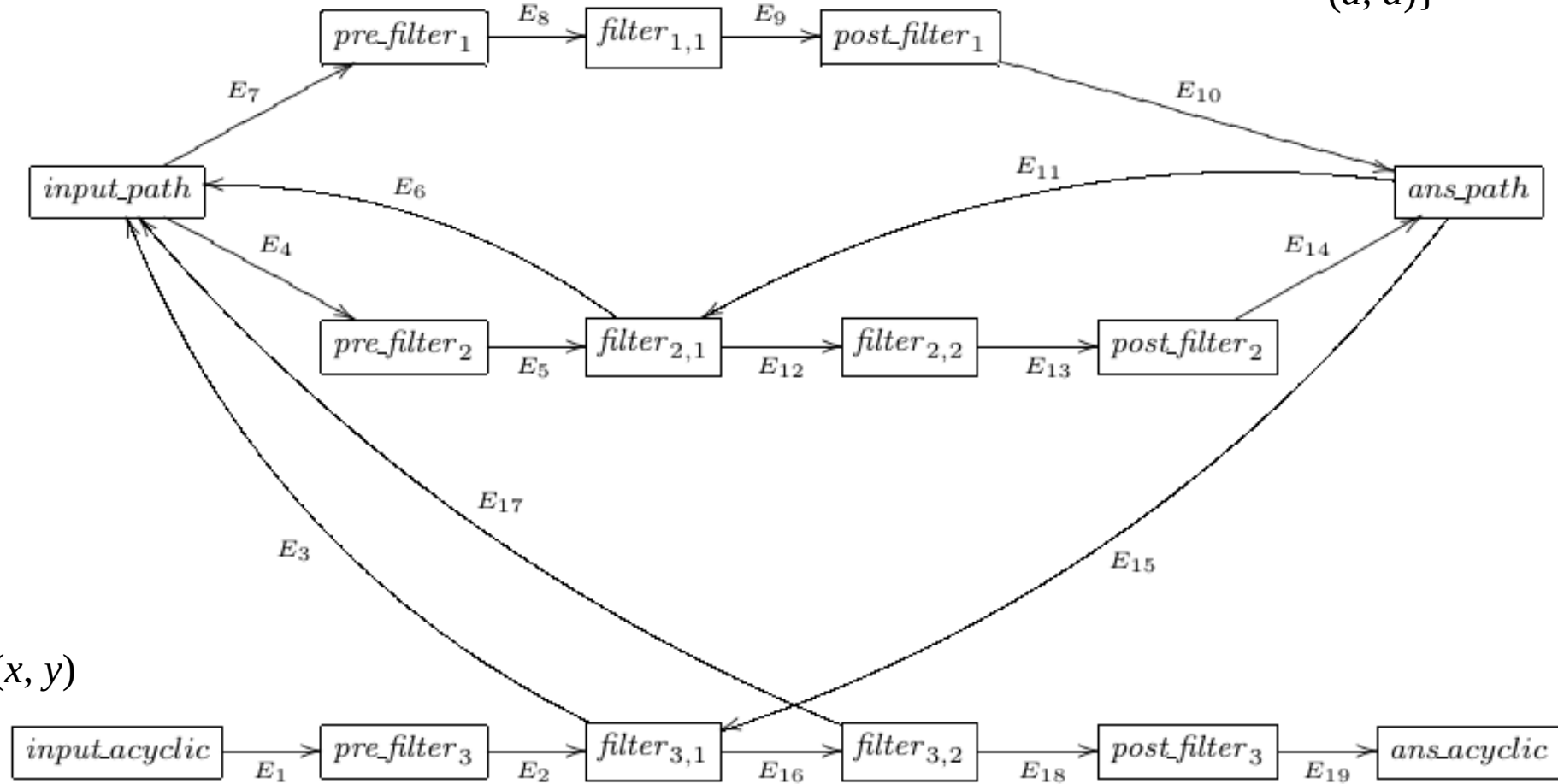
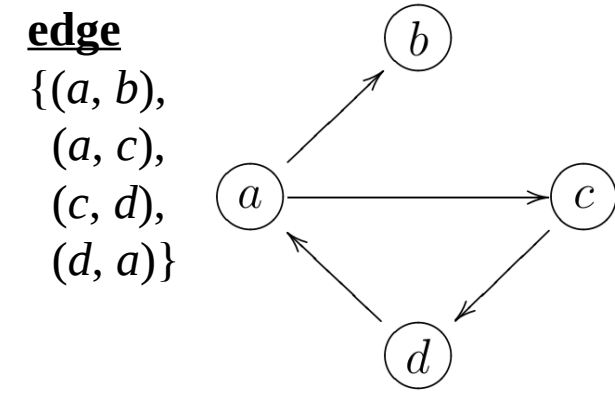
A Demonstration of the QSQN-STR Method for Evaluating Queries to Stratified Datalog[†]

(this should be displayed using the presentation mode)

$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$

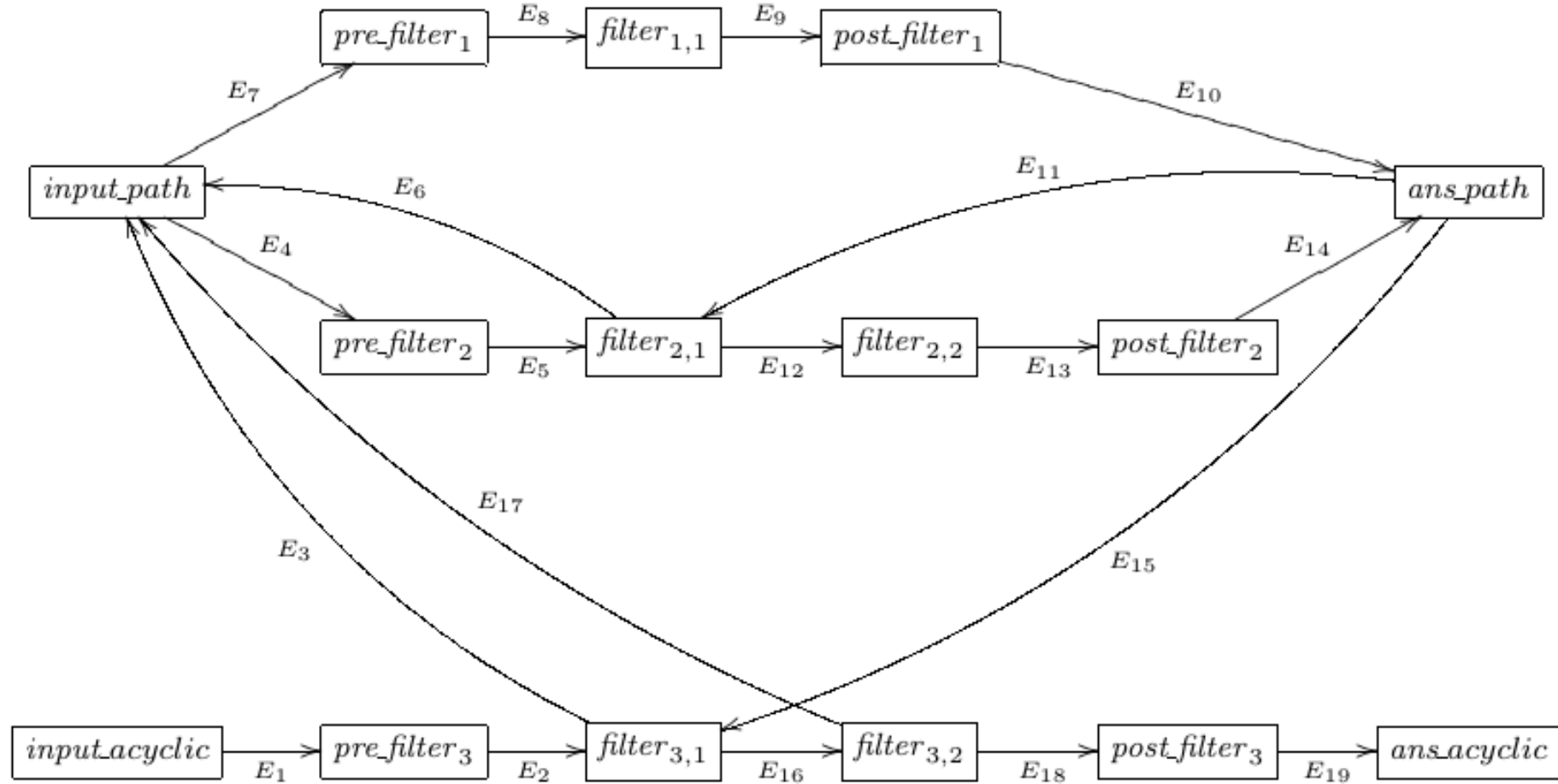
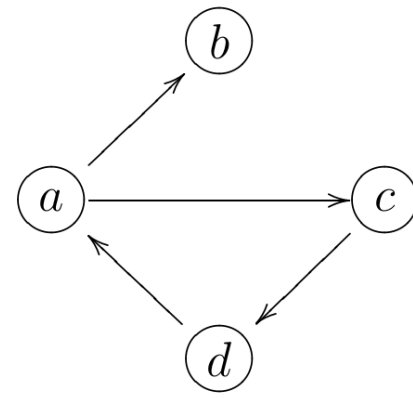


$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



Query: $acyclic(x, y)$

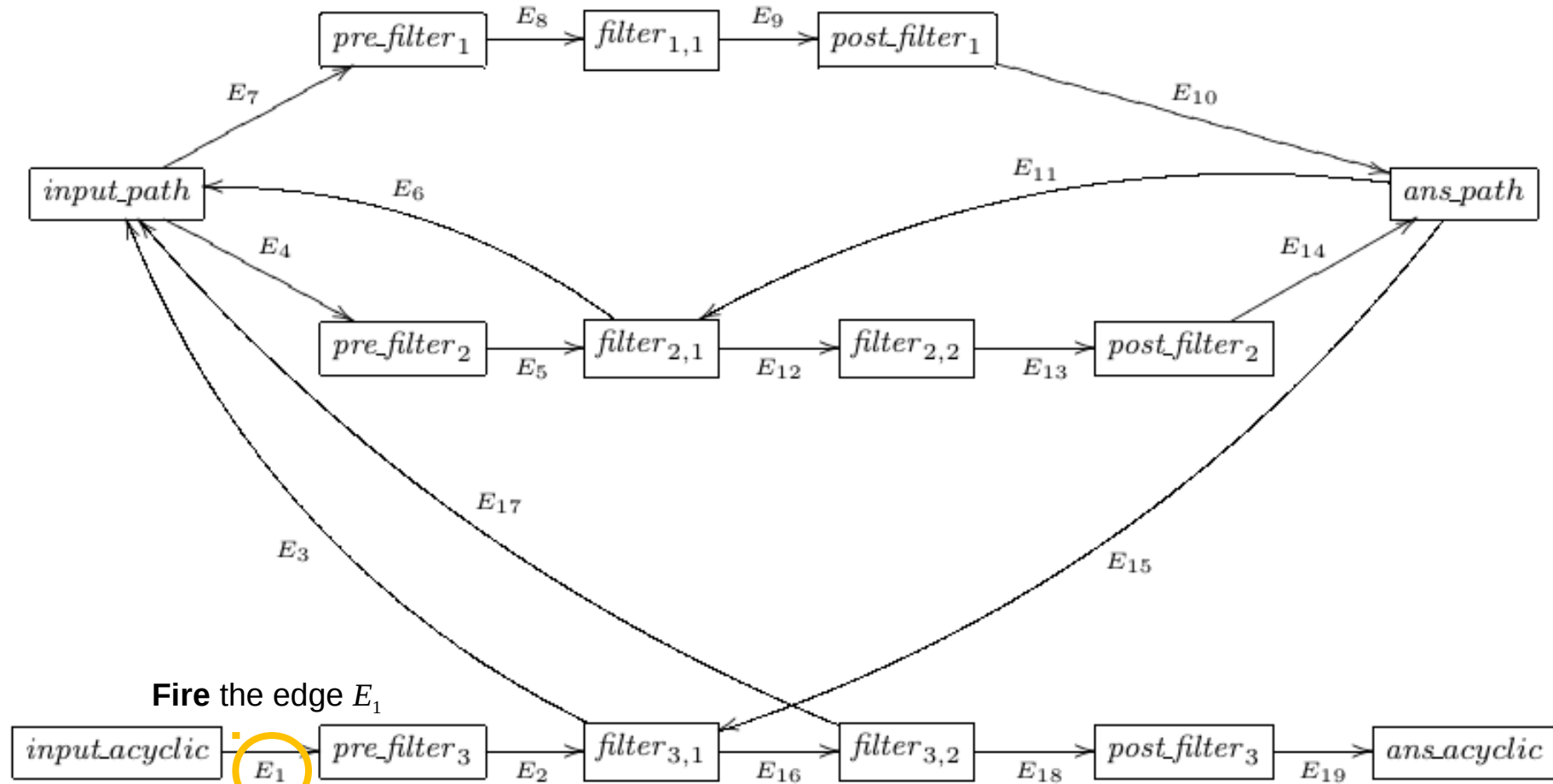
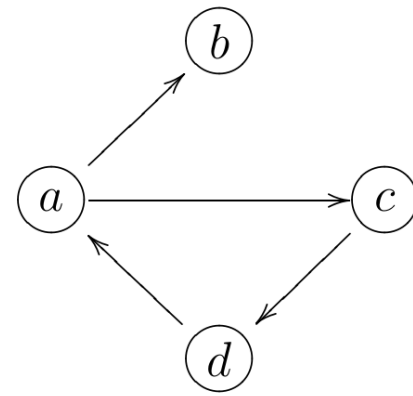
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



tuples

(x_1, y_1)

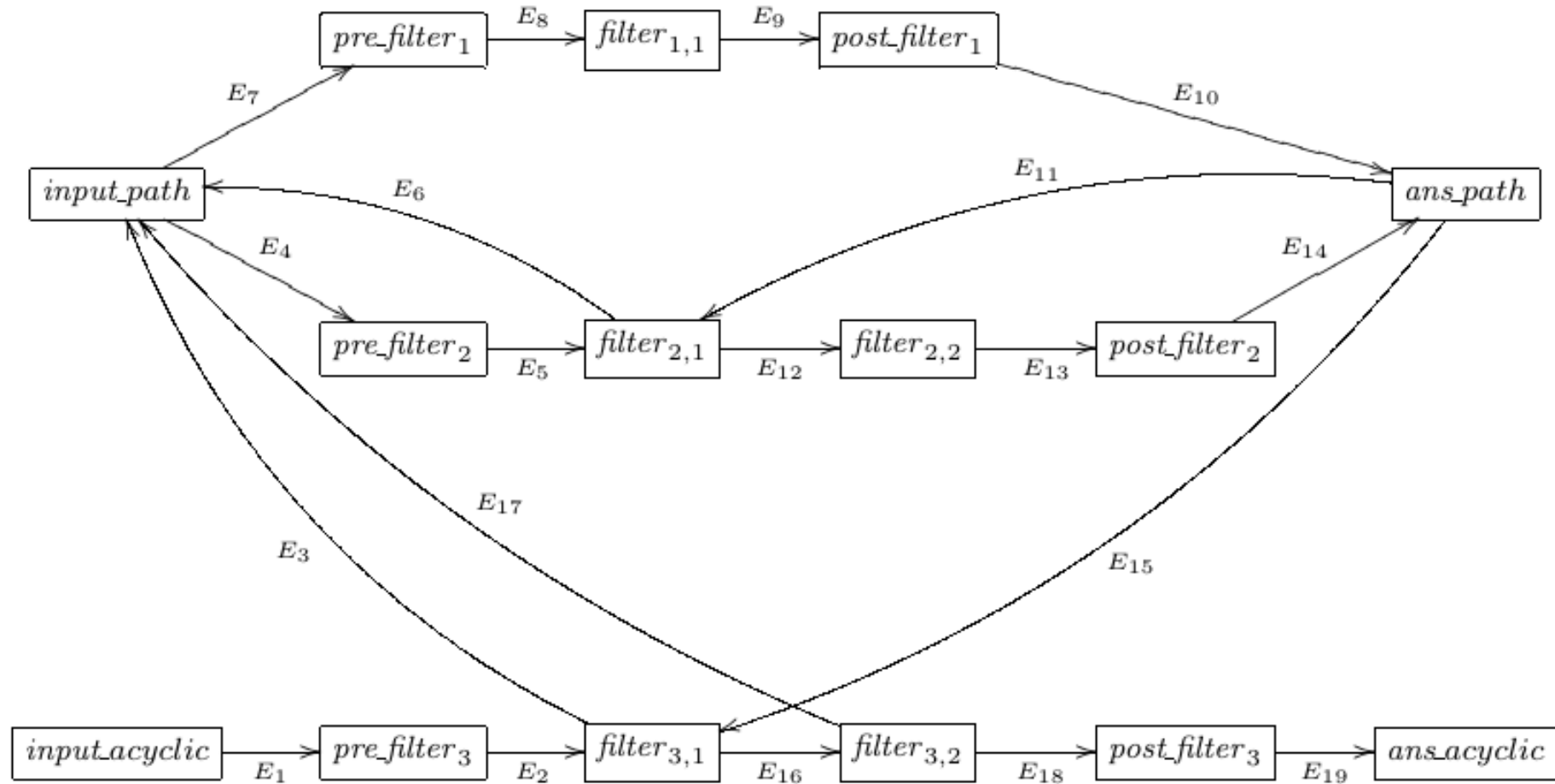
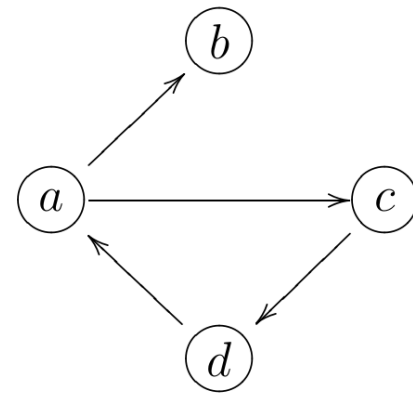
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



Fire the edge E_1

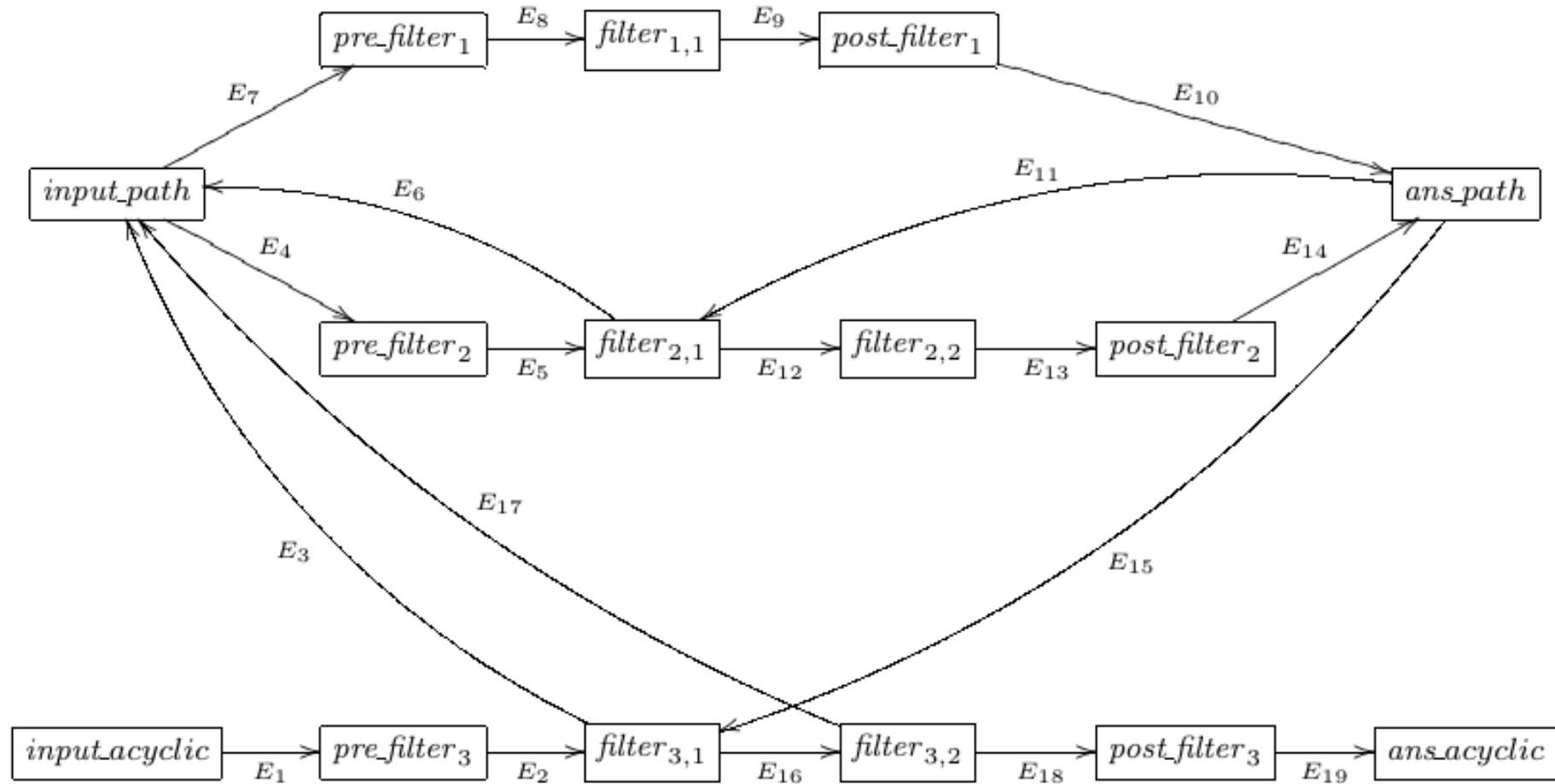
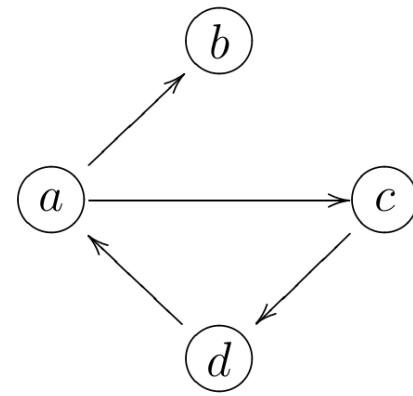
tuples
 (x_1, y_1)

$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



tuples
 $(x_1, y_1) \quad \{((x_1, y_1), \{x/x_1, y/y_1\})\}$

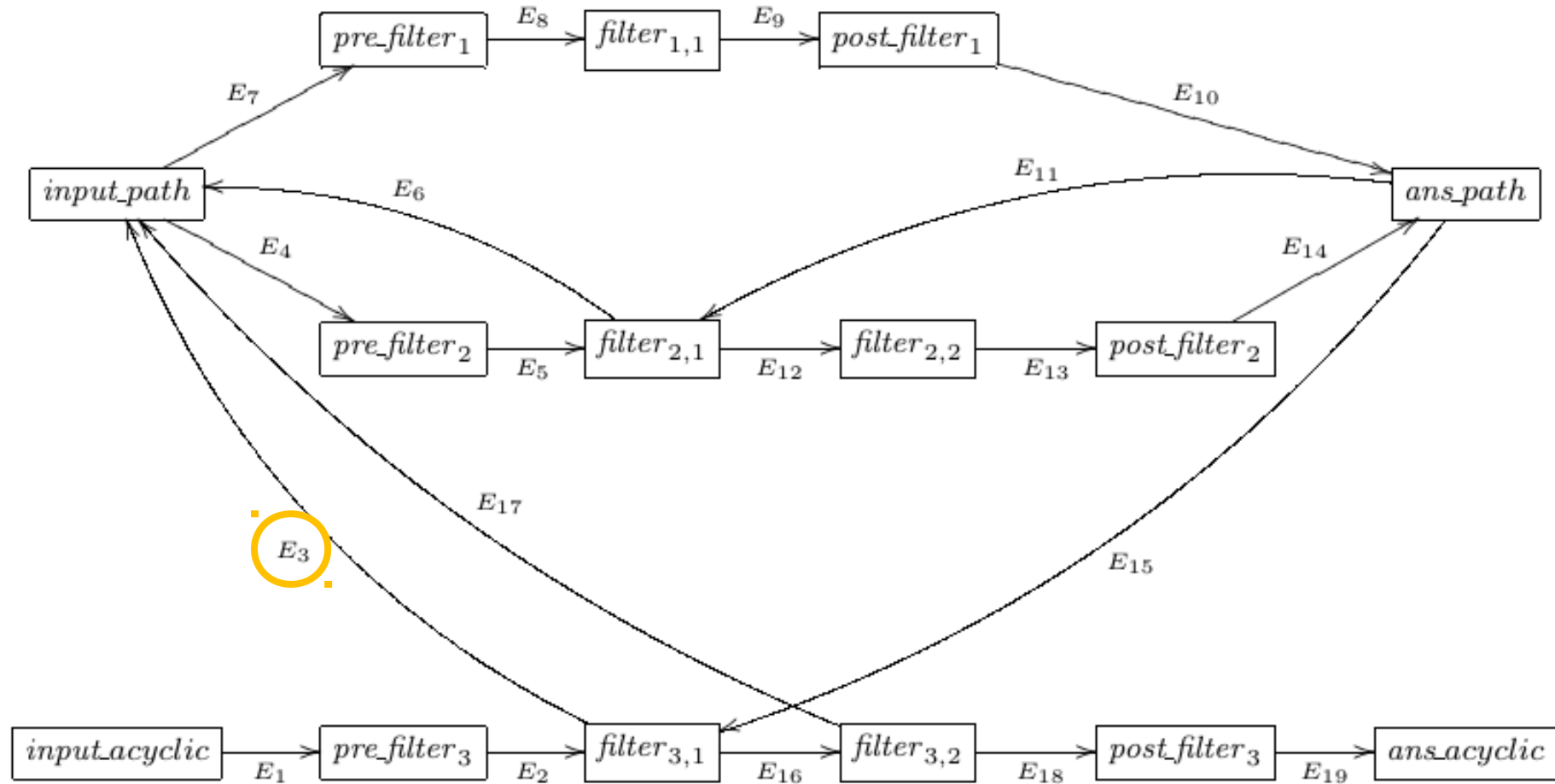
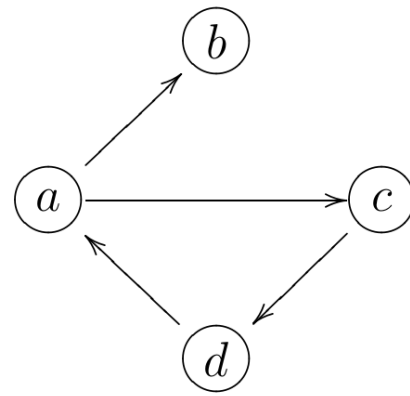
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



tuples
 (x_1, y_1)

subqueries
 $\{((x_1, y_1), \{x/x_1, y/y_1\})\}$

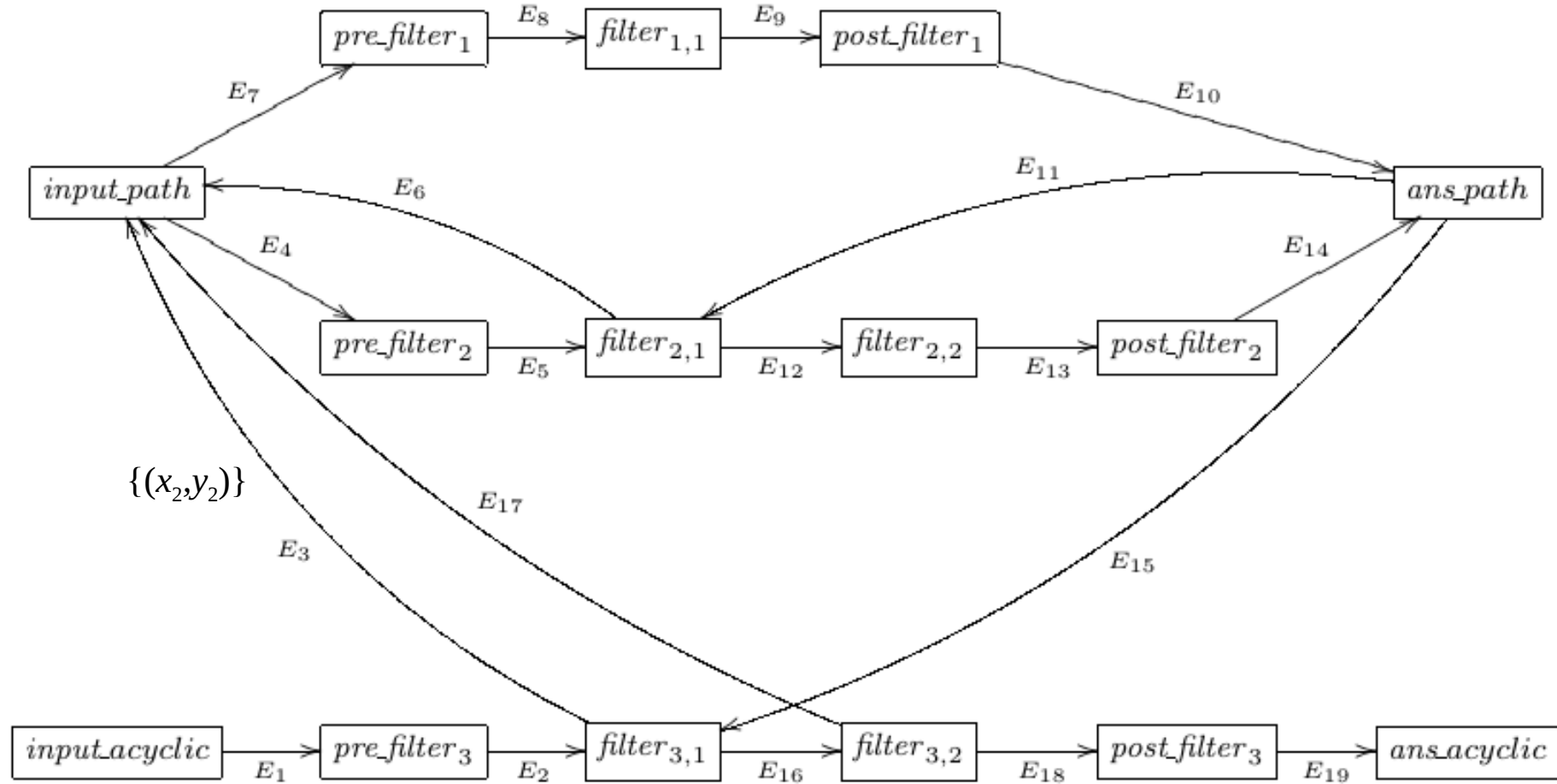
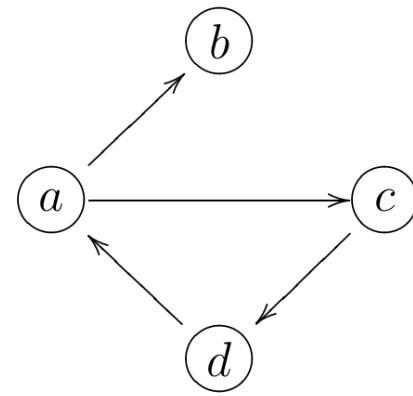
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



tuples
 (x_1, y_1)

subqueries
 $\{((x_1, y_1), \{x/x_1, y/y_1\})\}$

$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



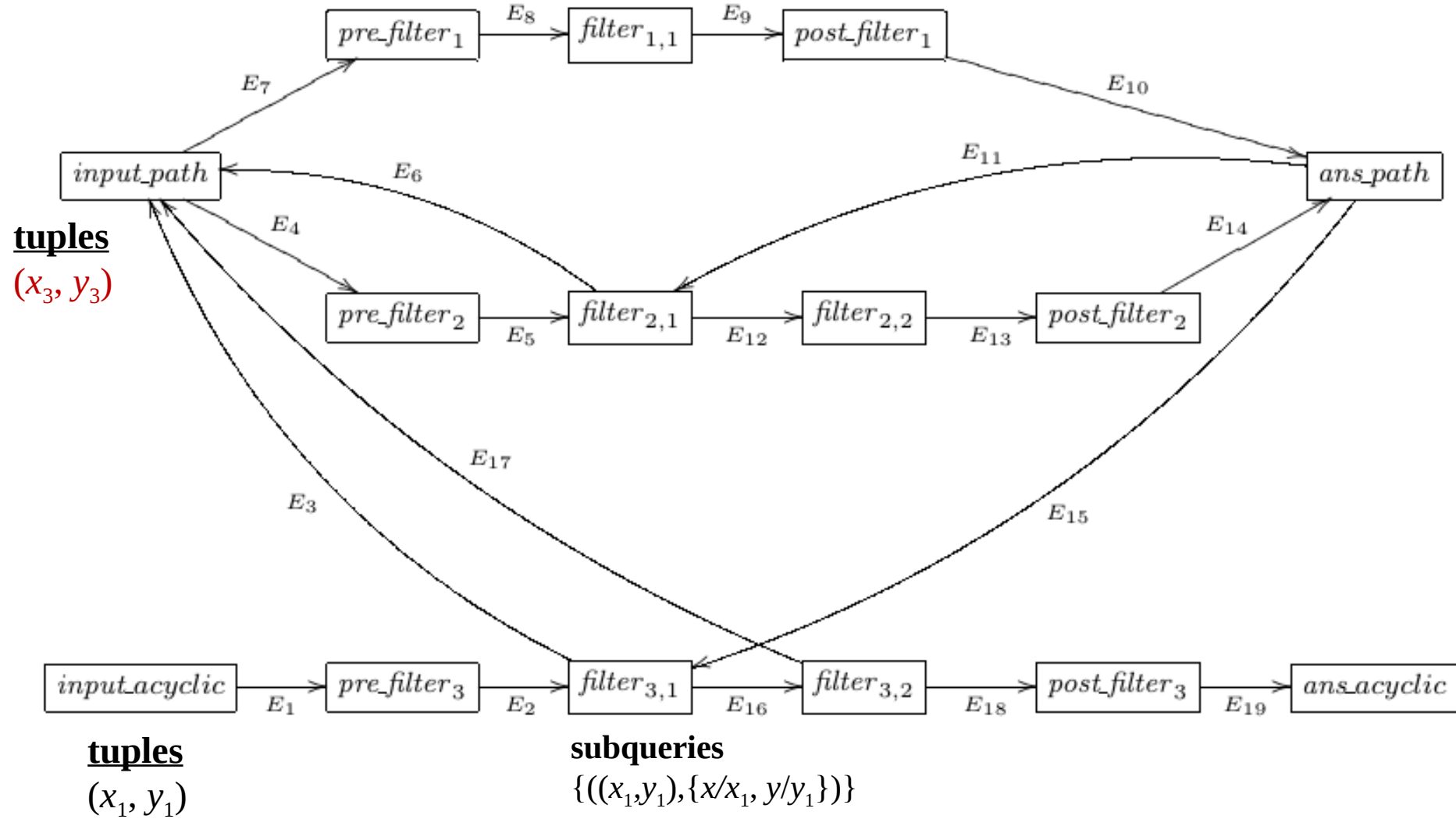
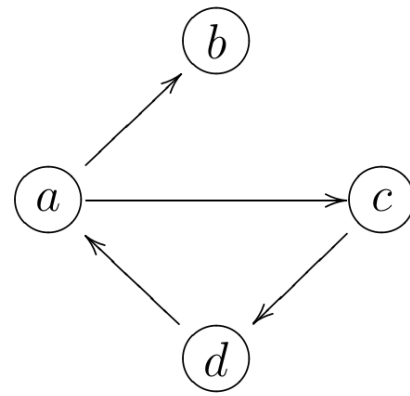
tuples

(x_1, y_1)

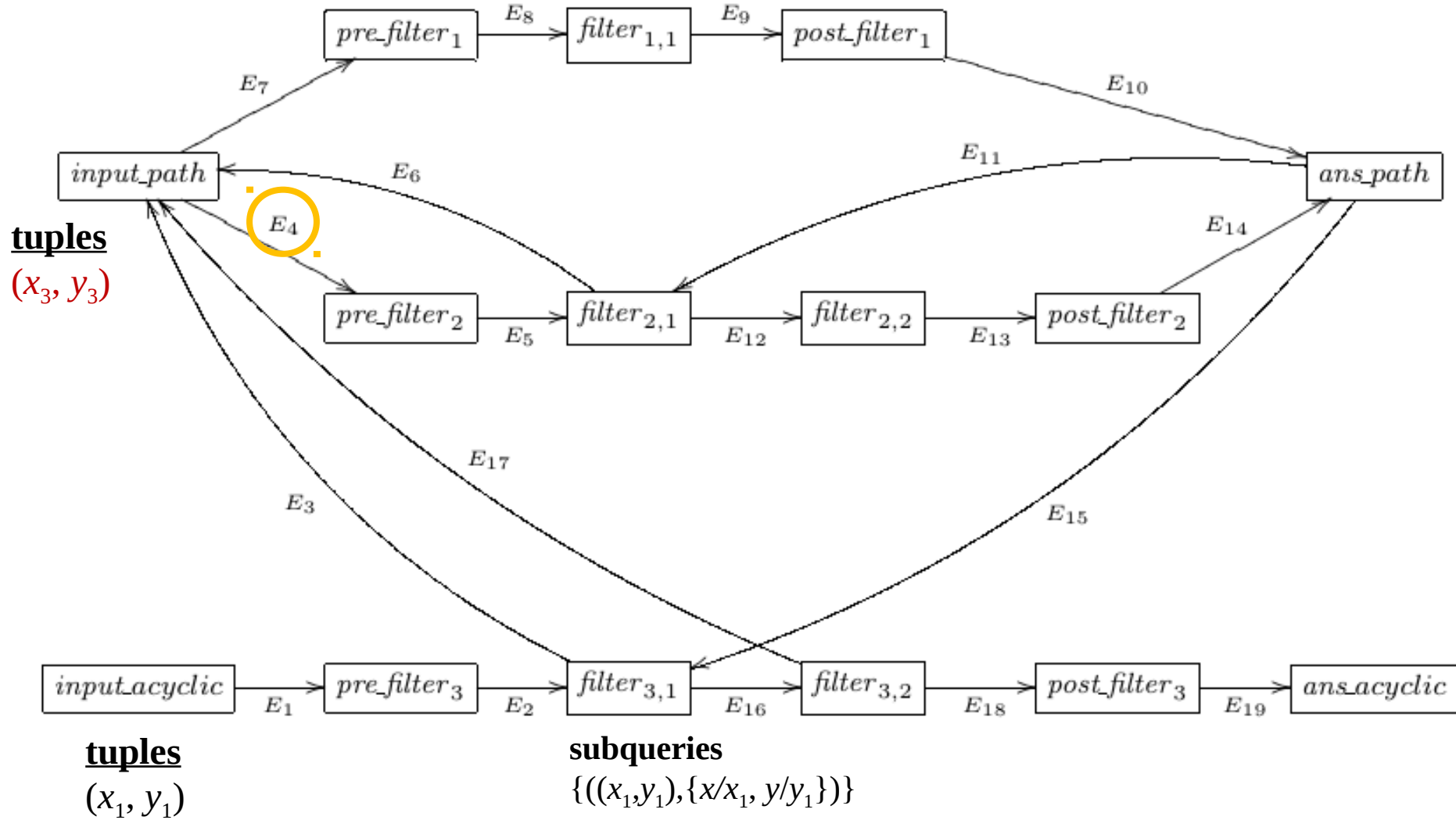
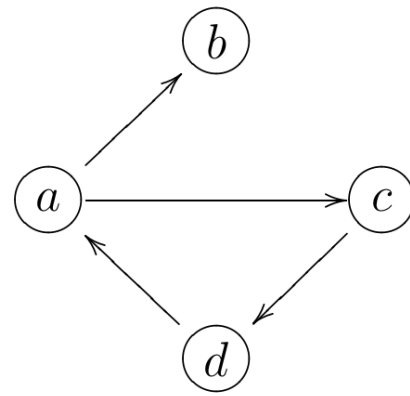
subqueries

$\{((x_1, y_1), \{x/x_1, y/y_1\})\}$

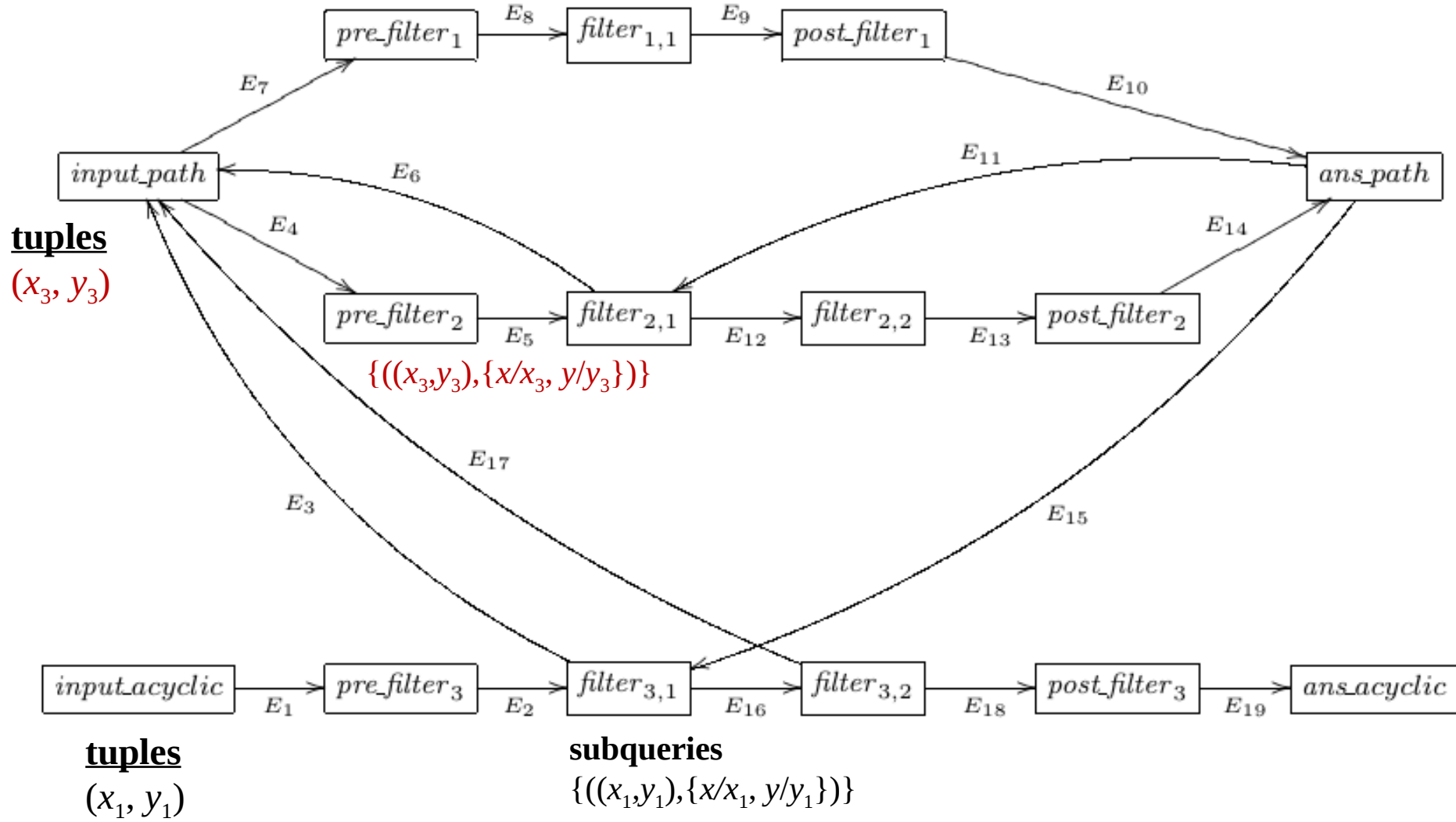
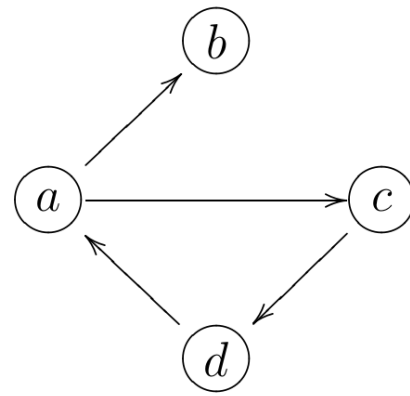
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



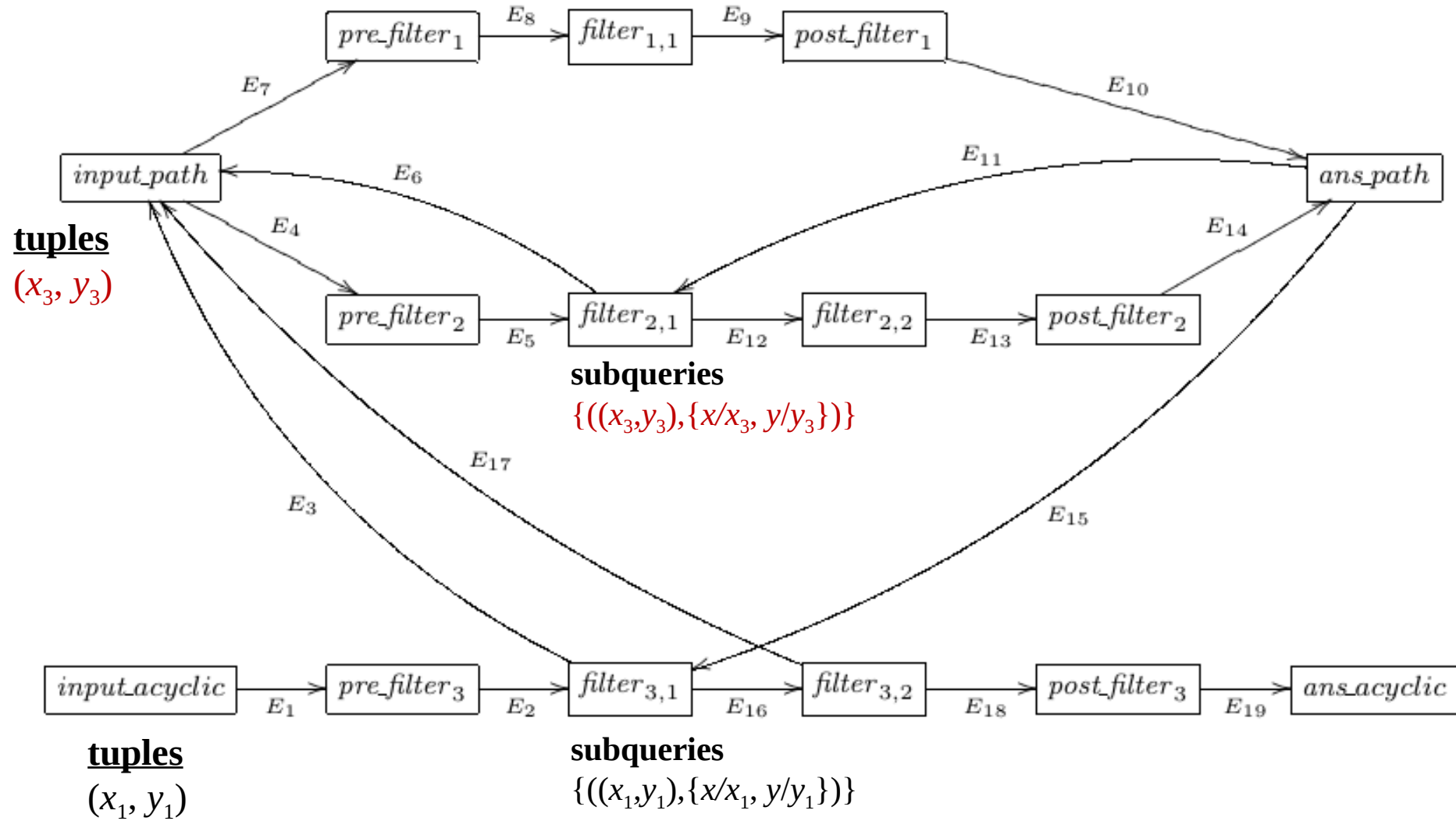
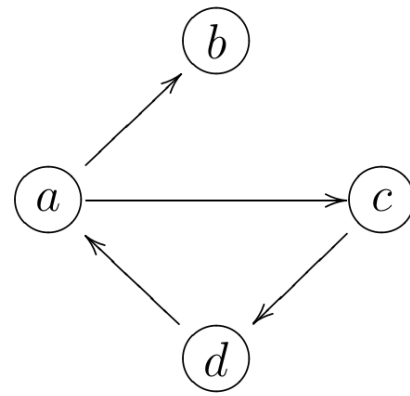
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



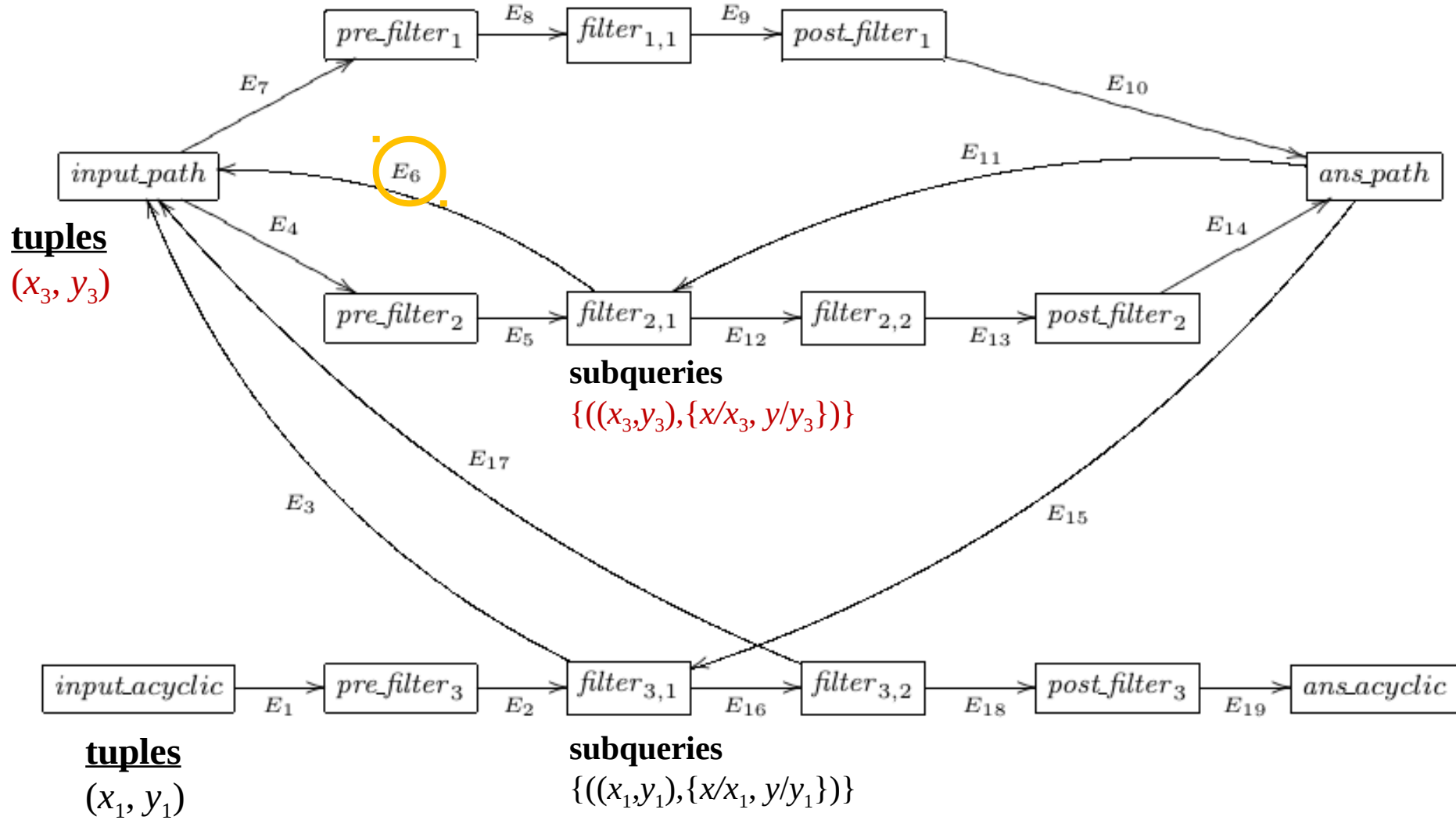
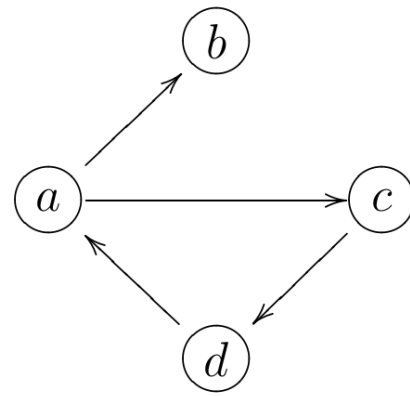
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



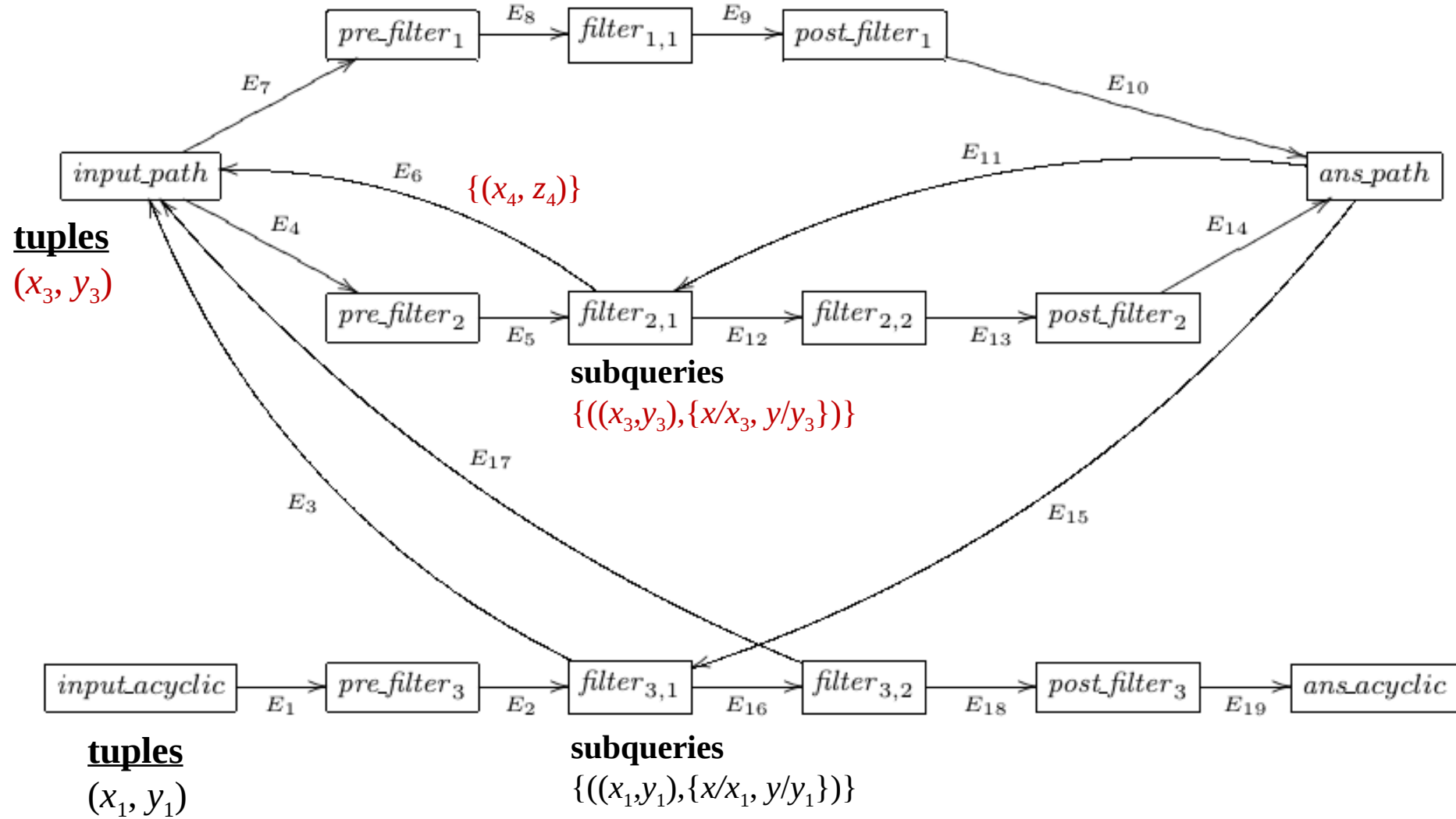
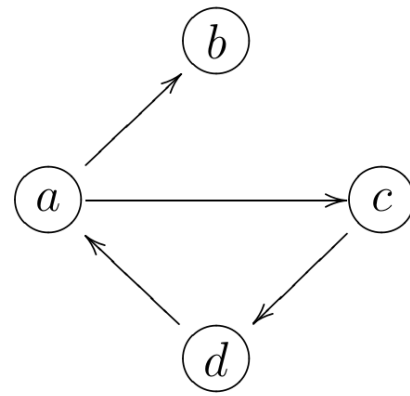
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



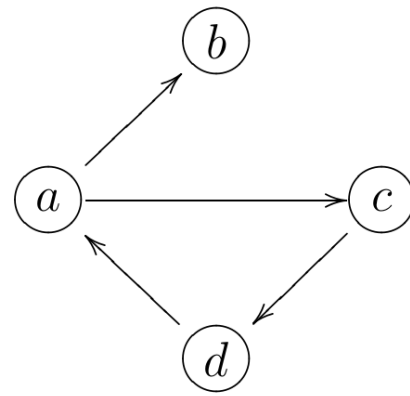
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



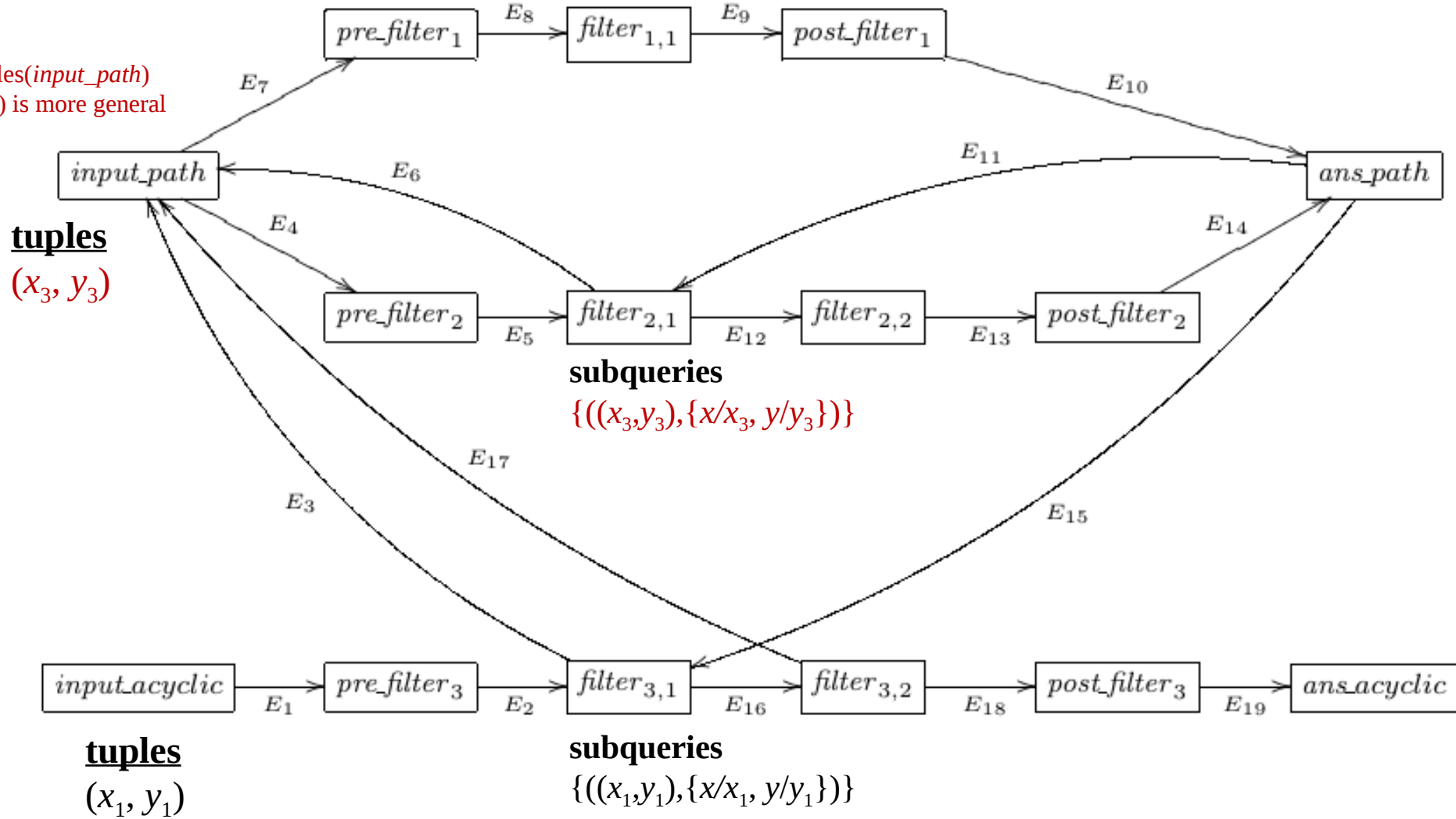
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



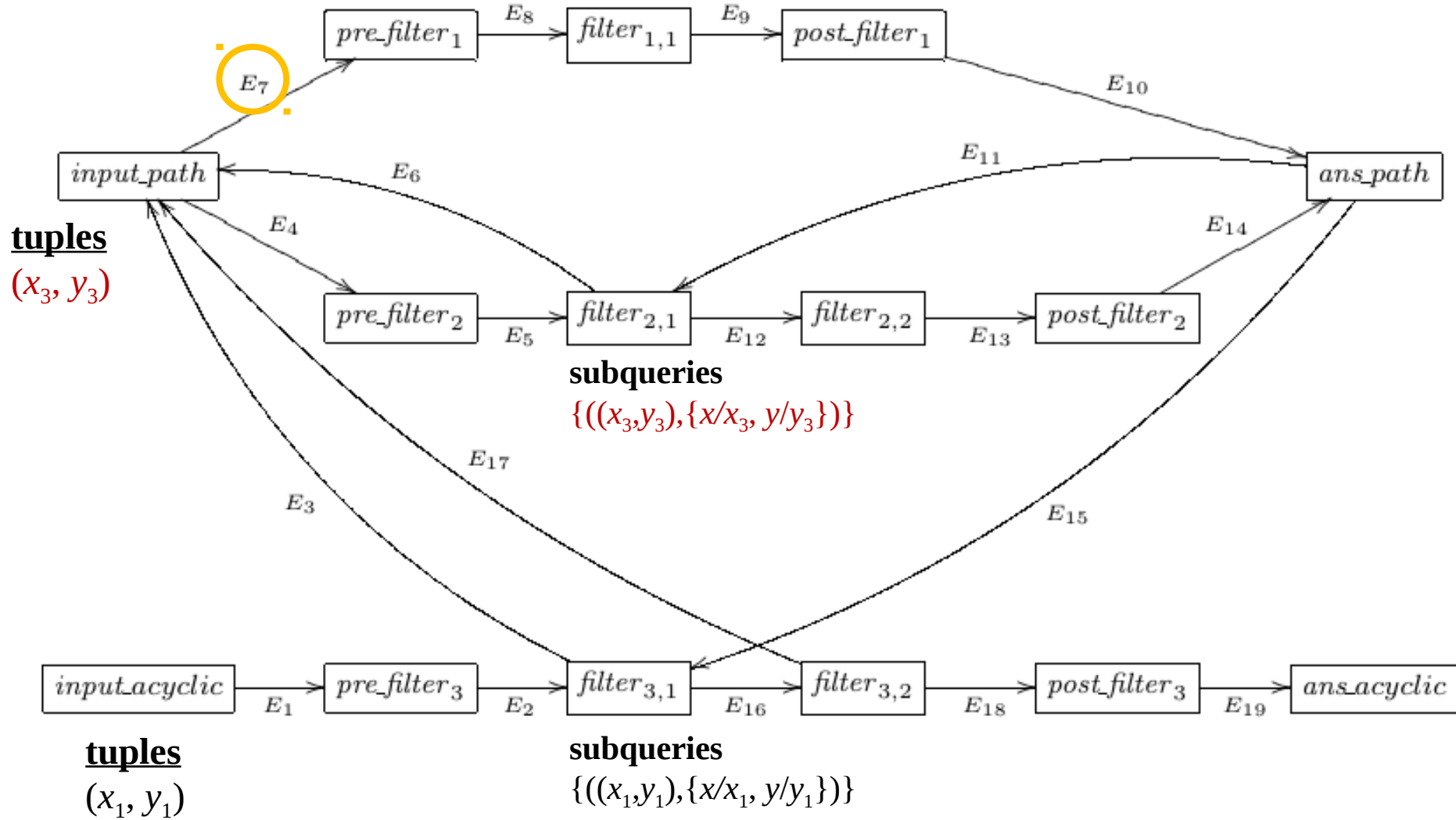
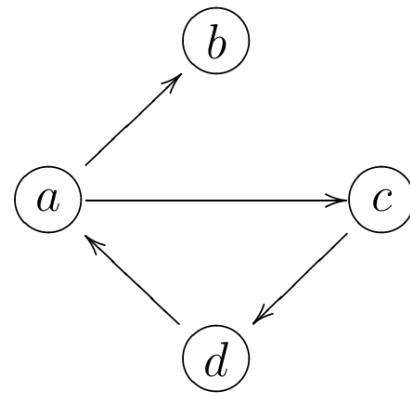
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



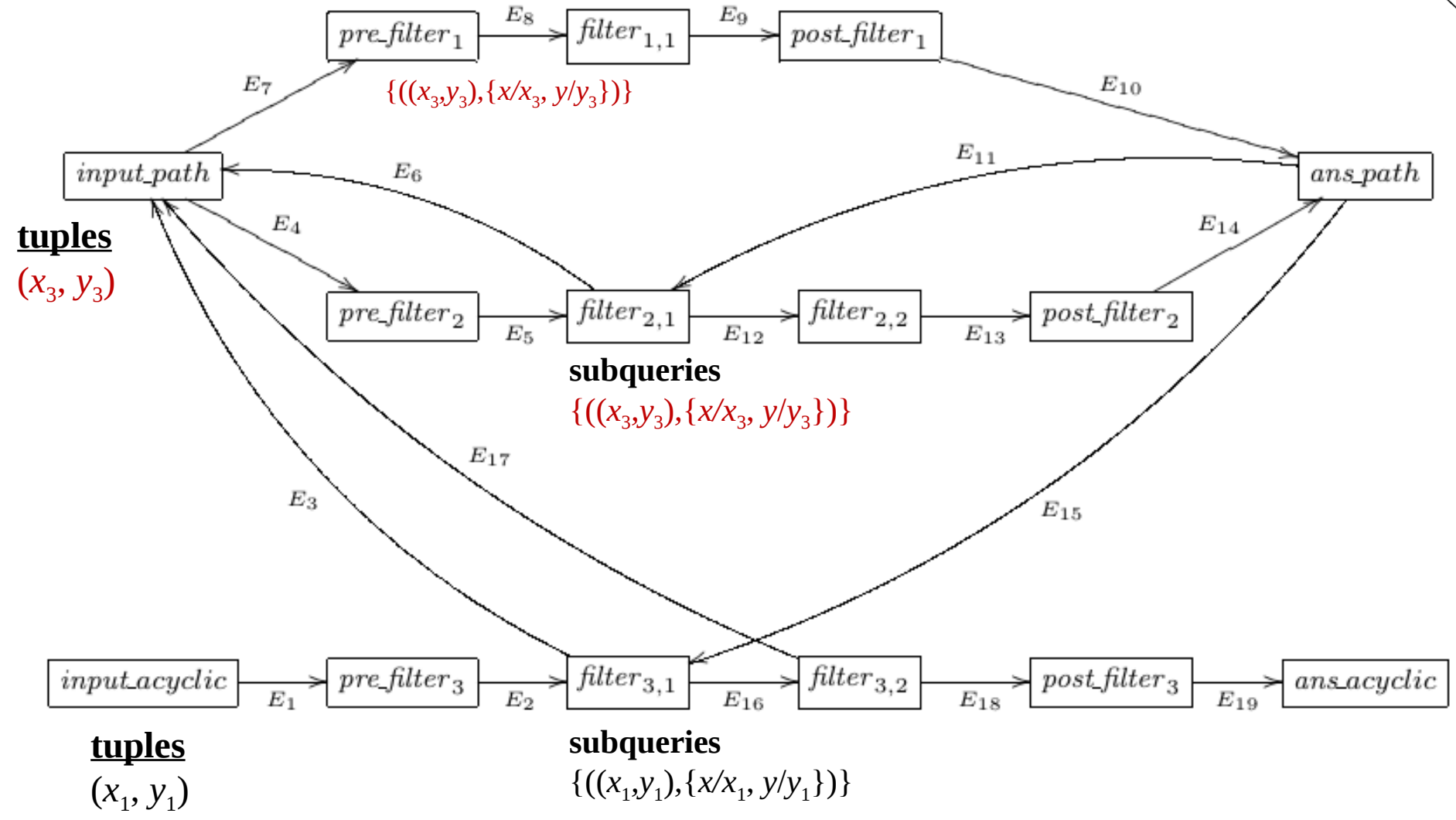
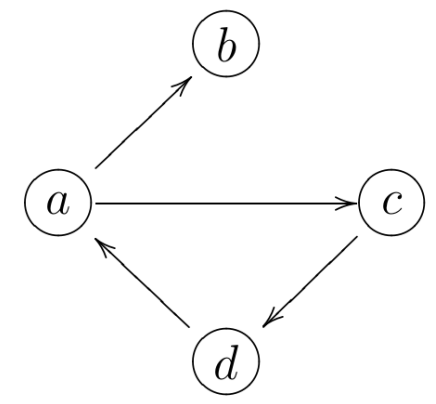
no tuple is added to $tuples(input_path)$ because the tuple (x_3, y_3) is more general than any other tuples



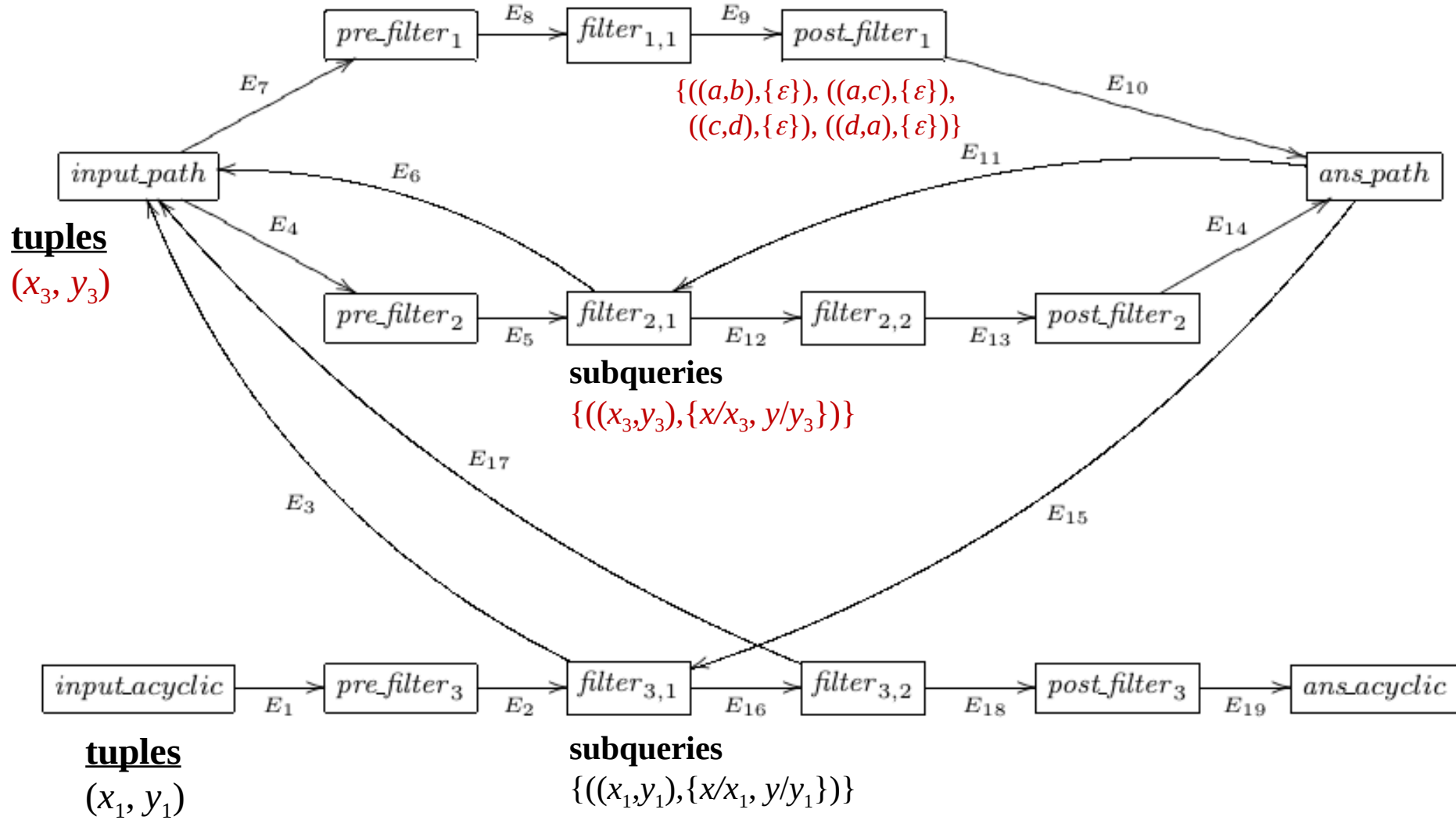
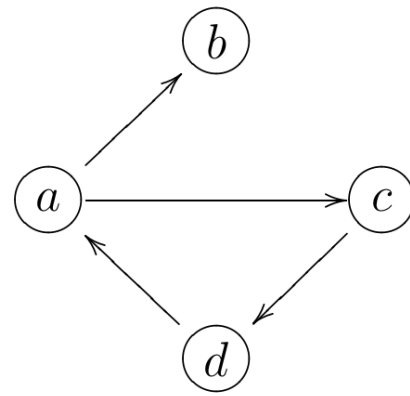
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



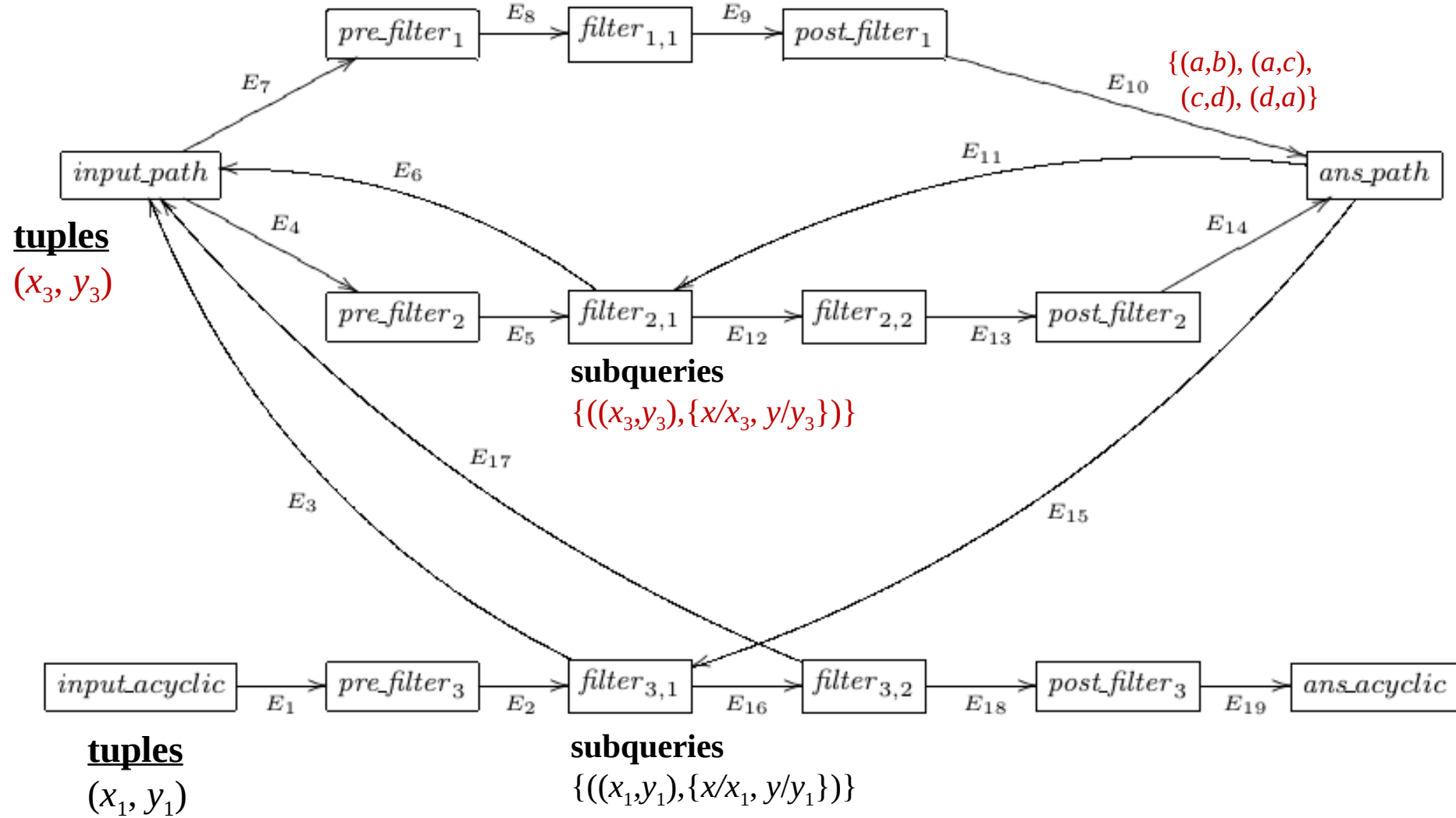
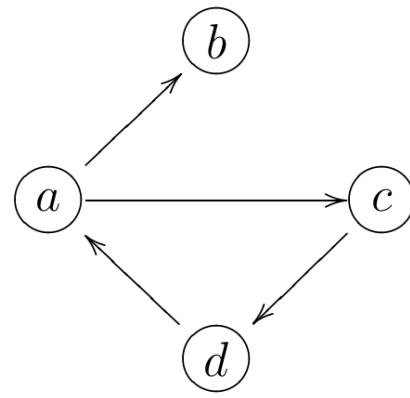
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



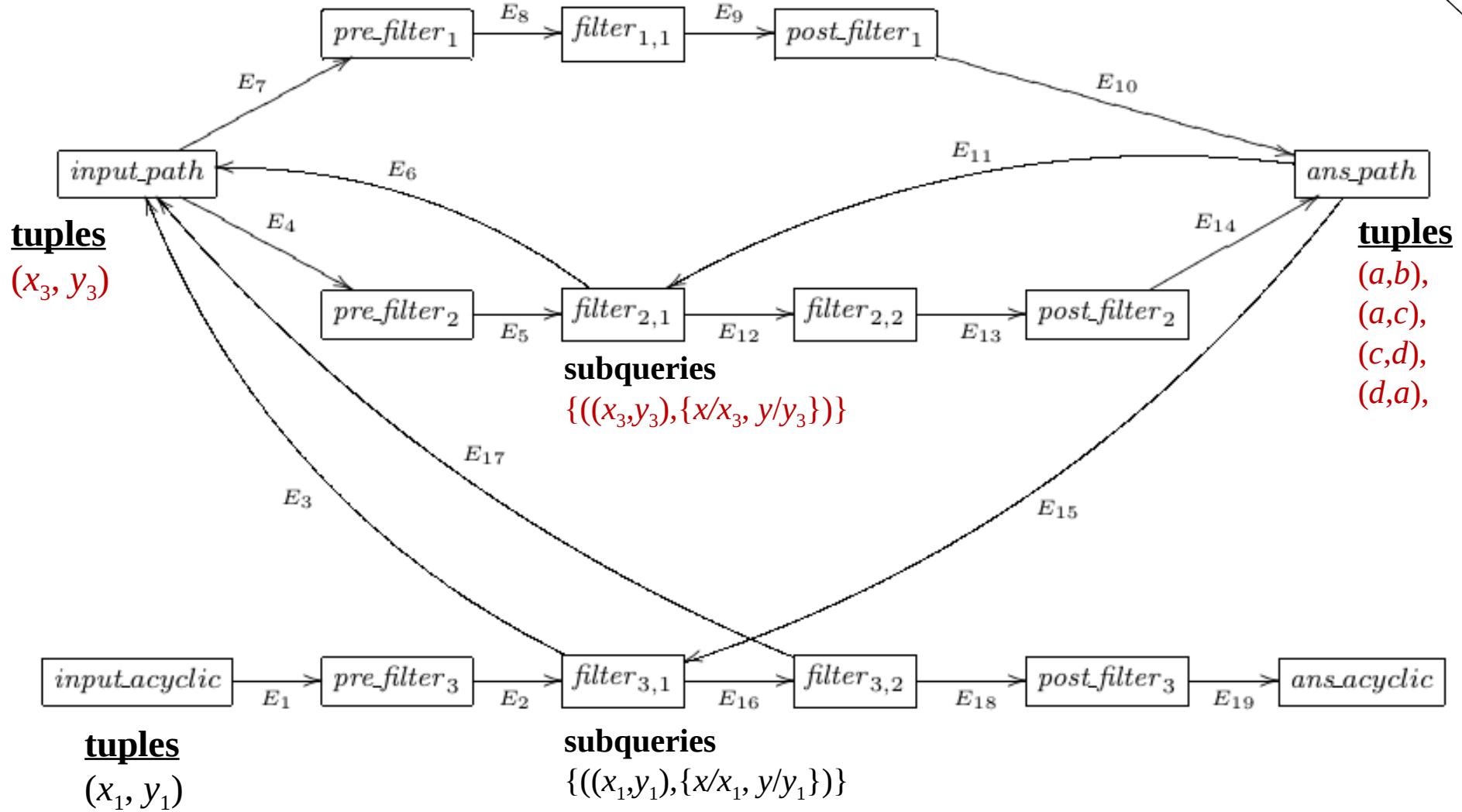
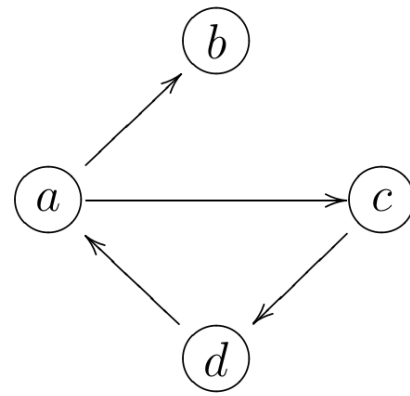
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



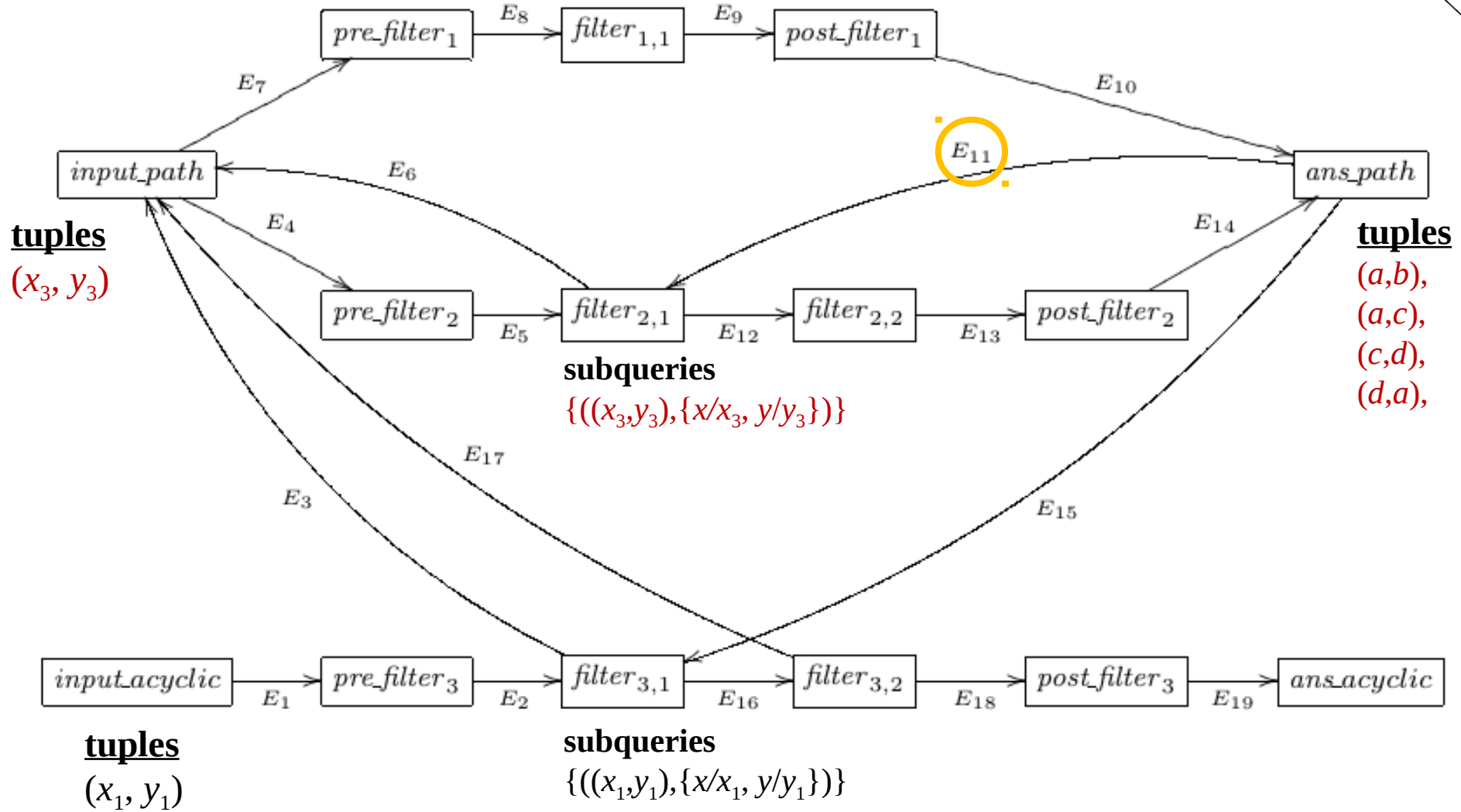
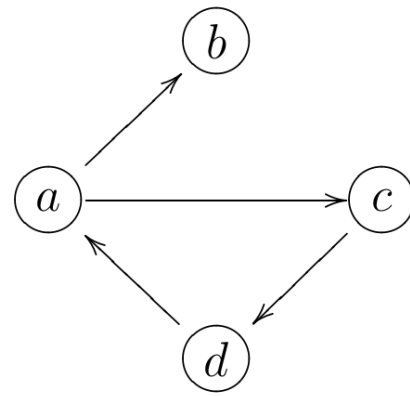
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



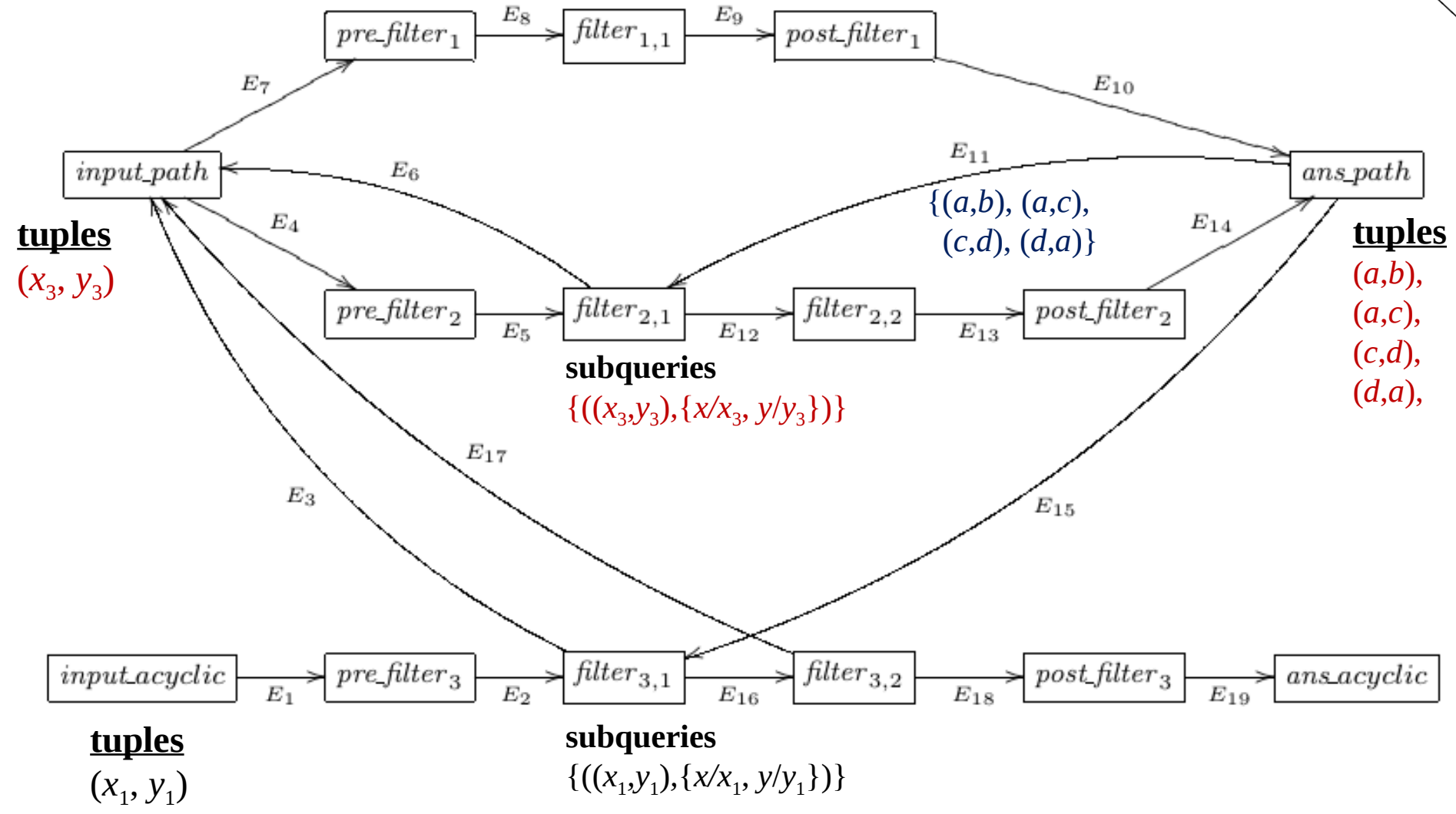
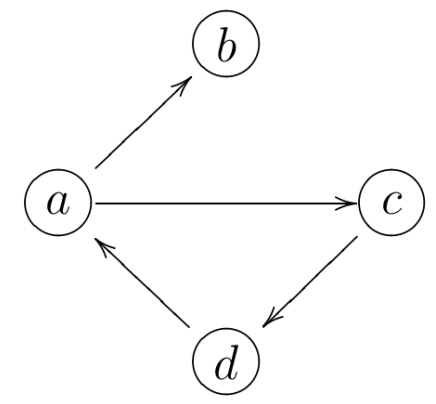
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



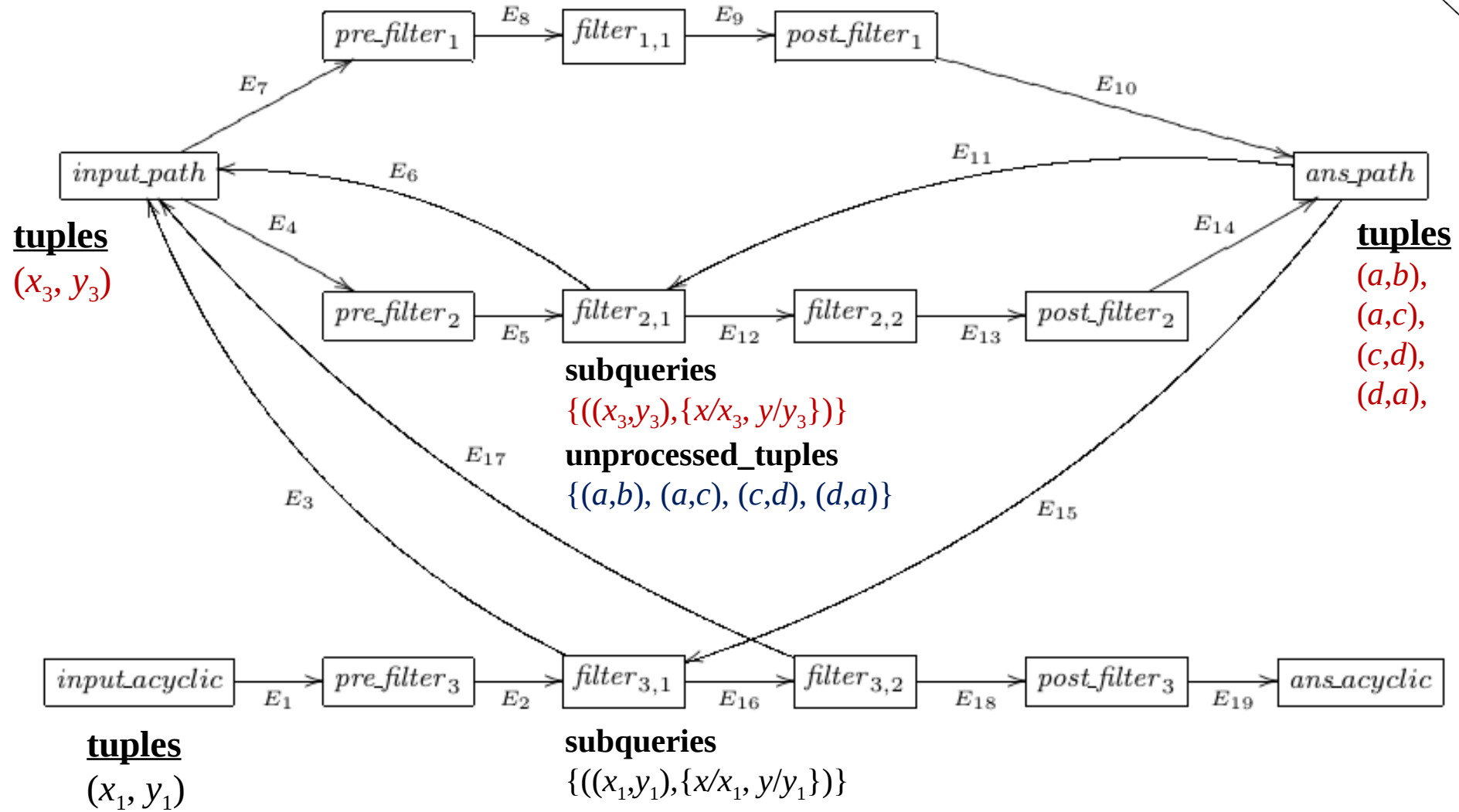
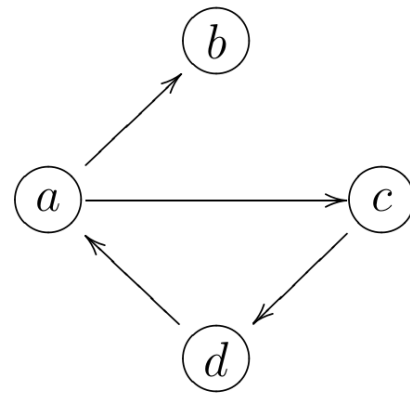
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



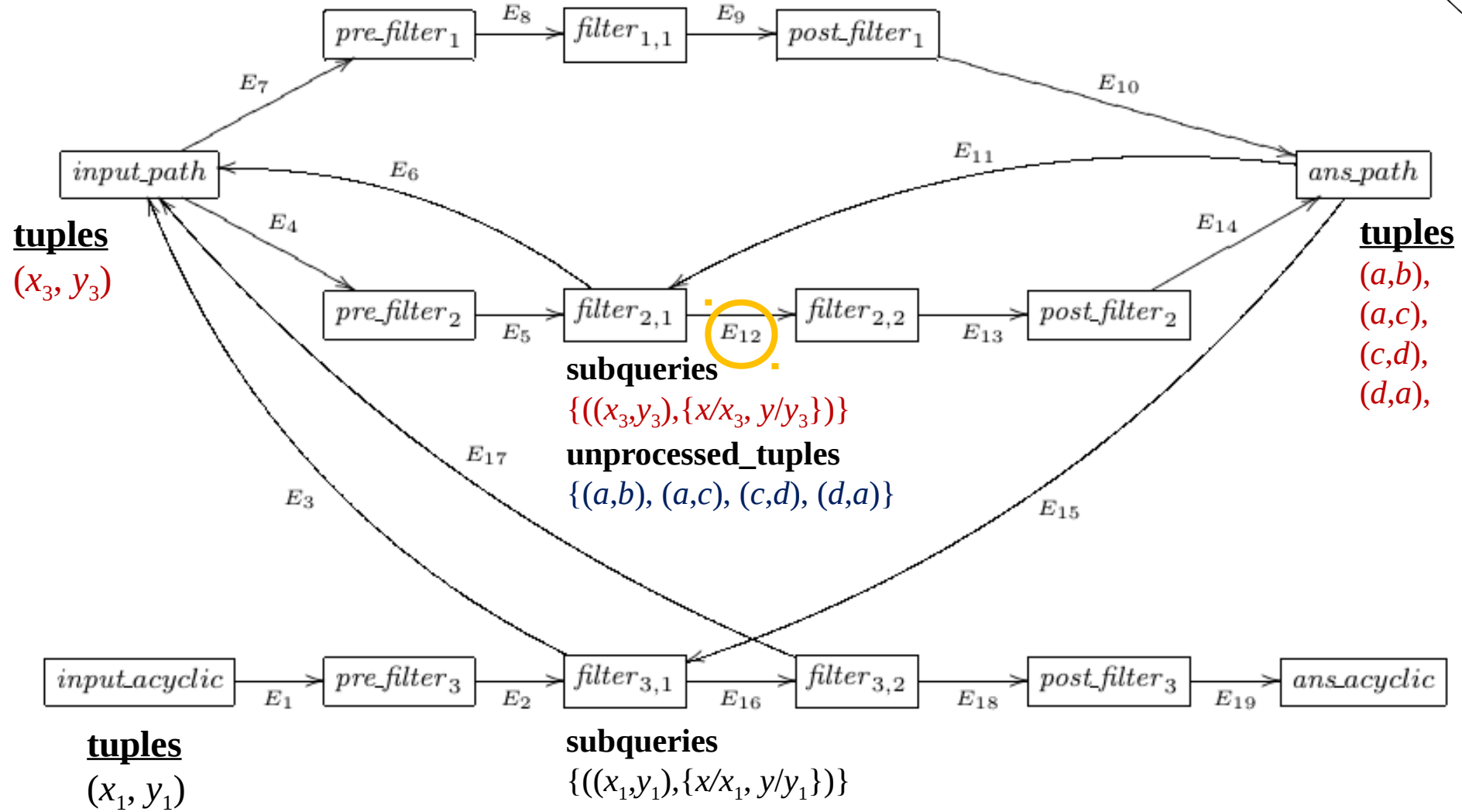
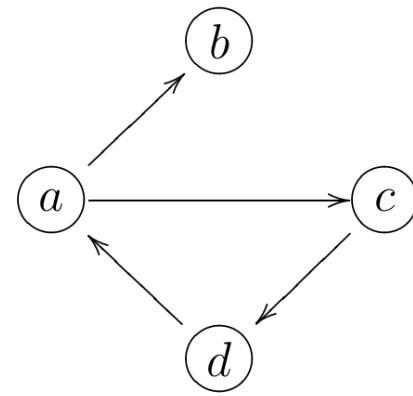
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



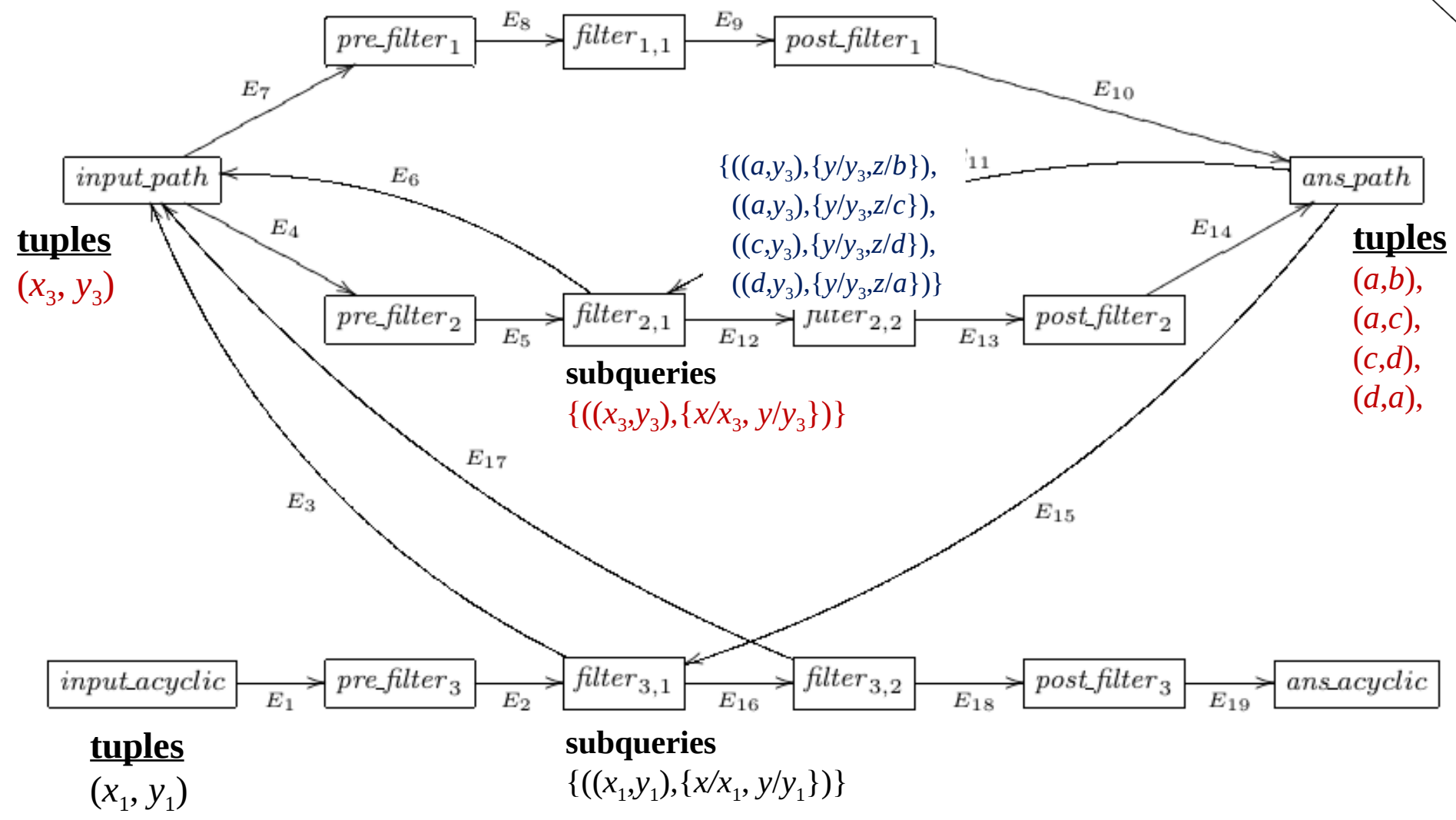
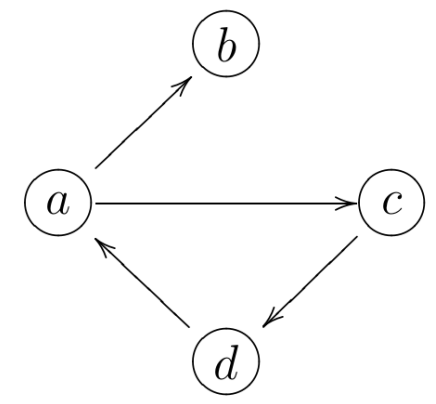
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



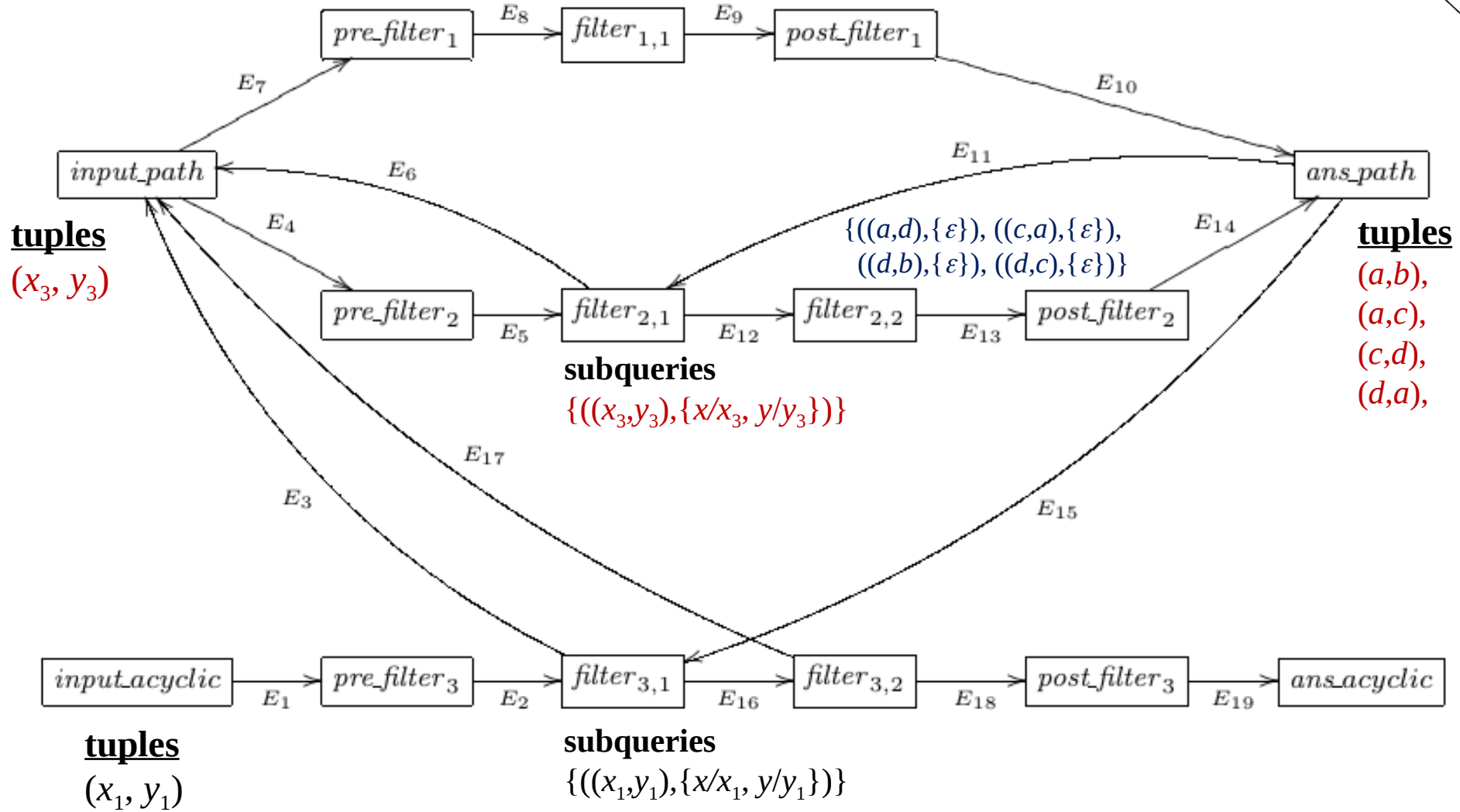
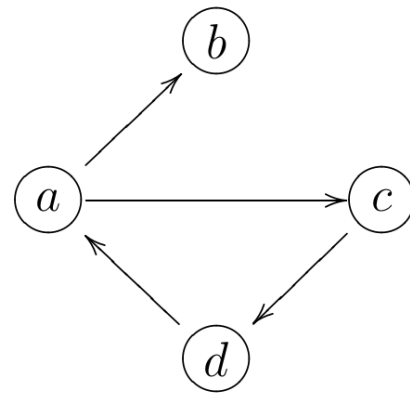
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



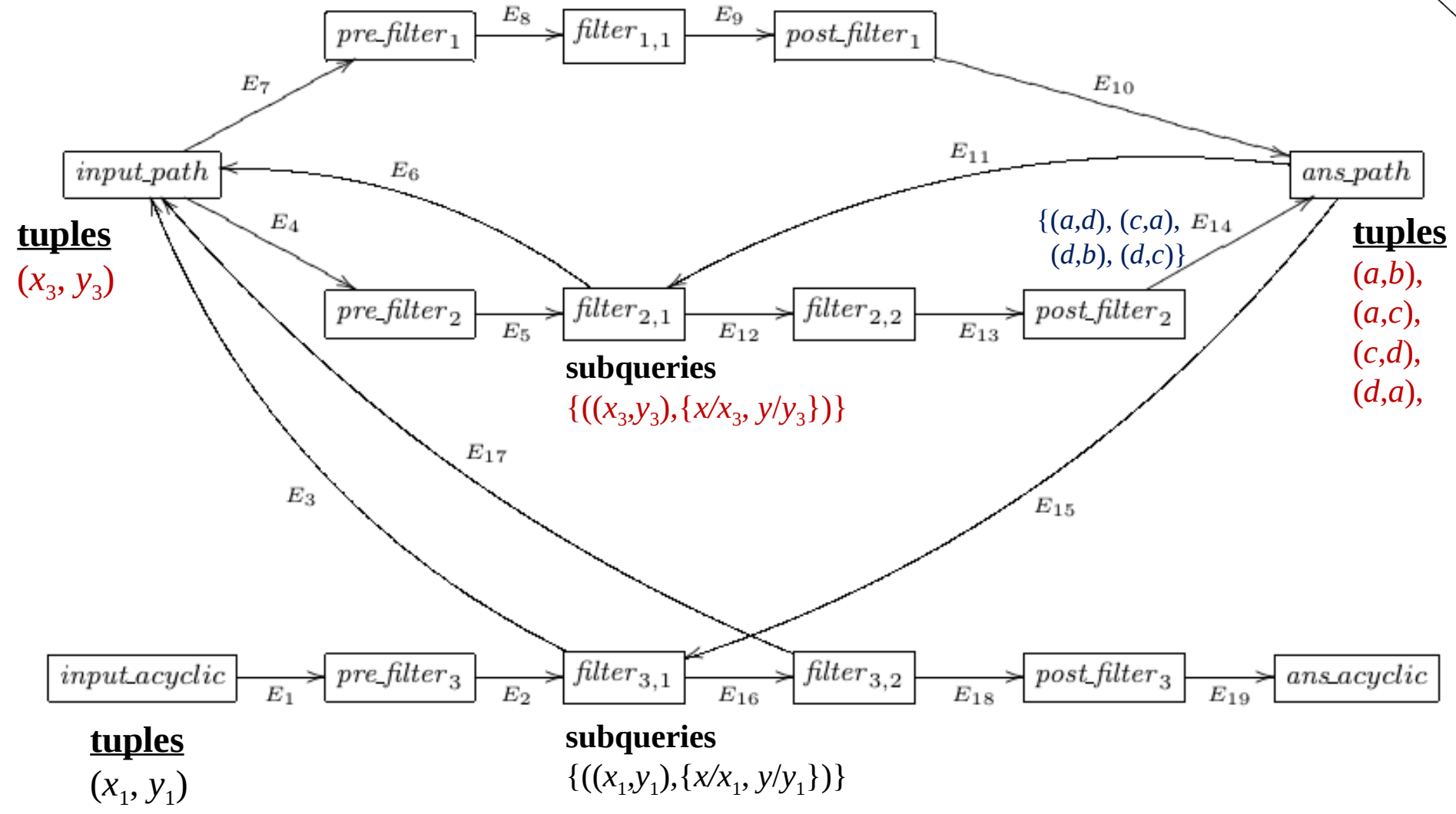
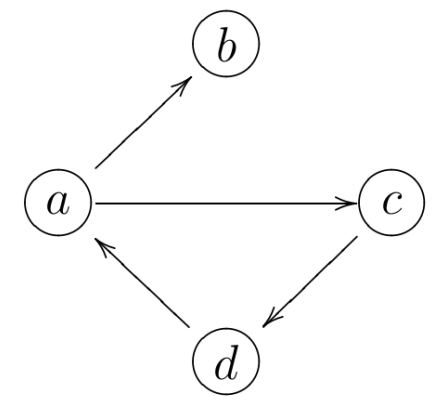
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



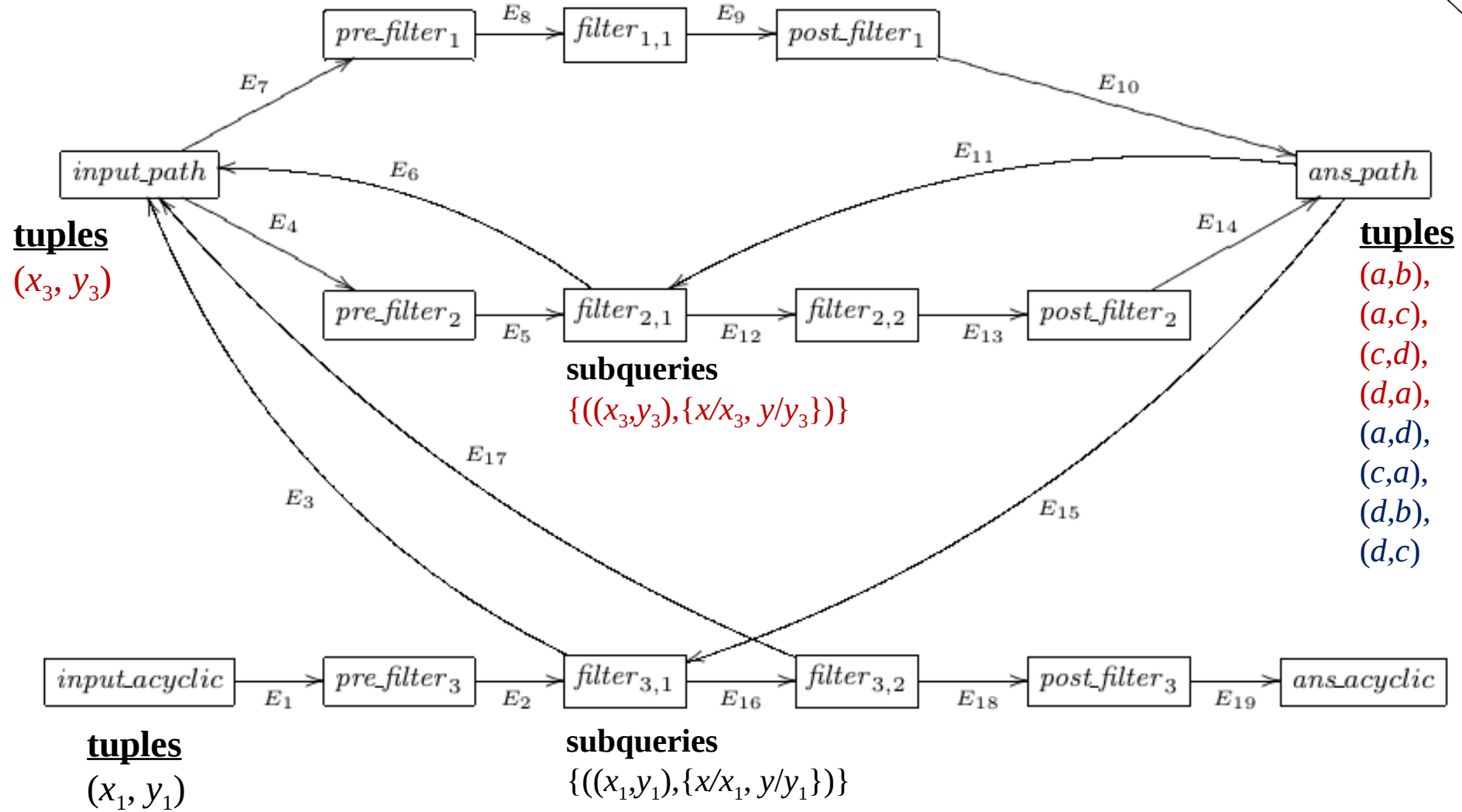
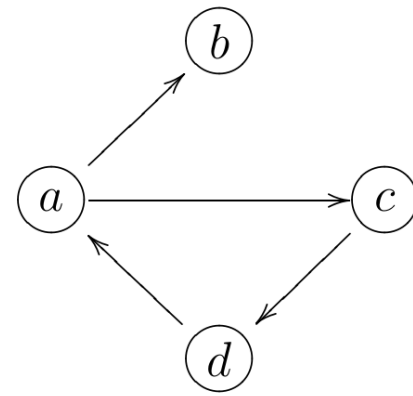
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



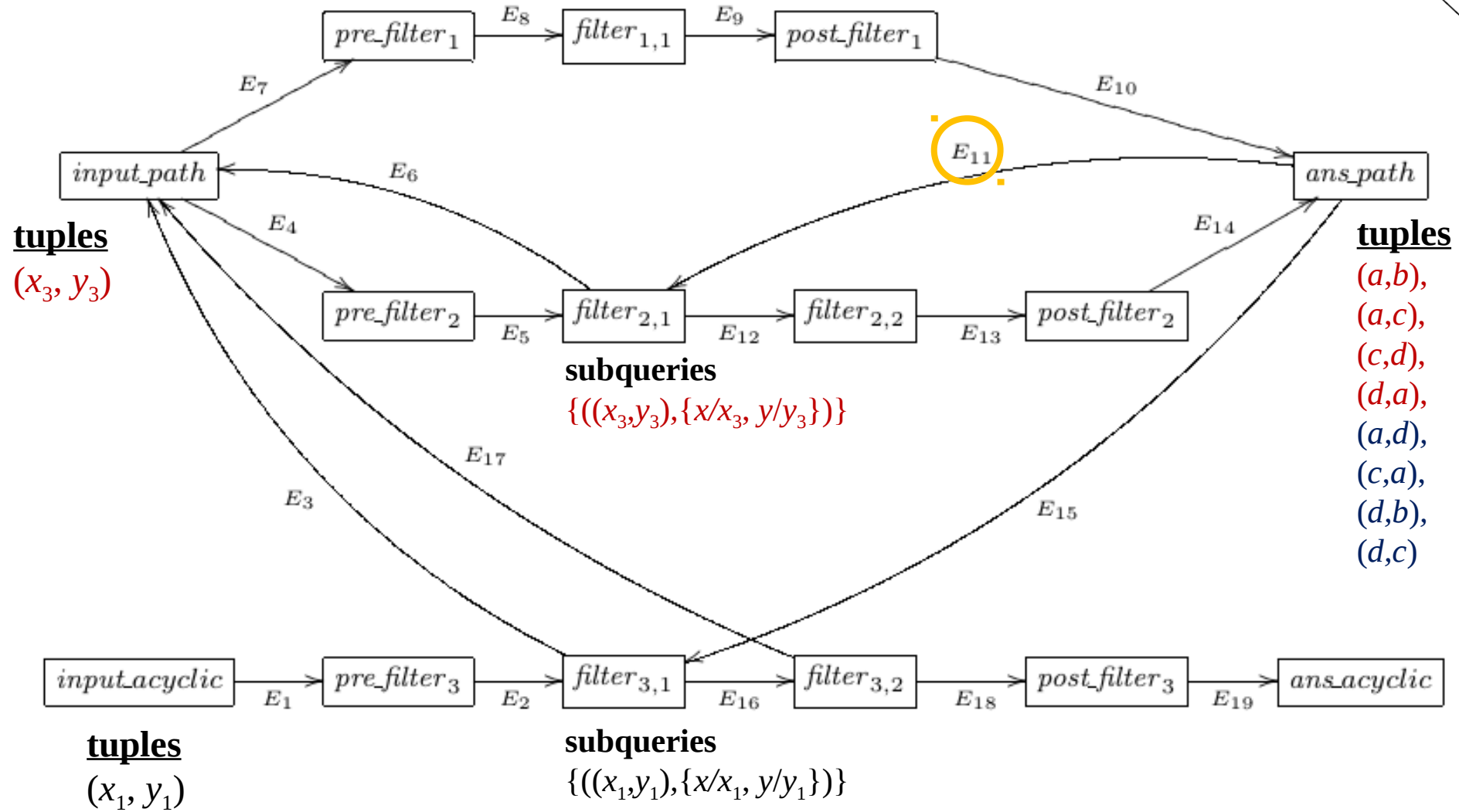
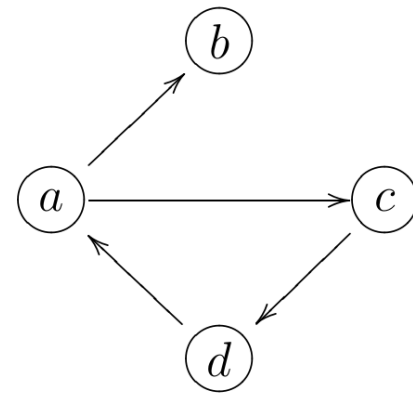
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



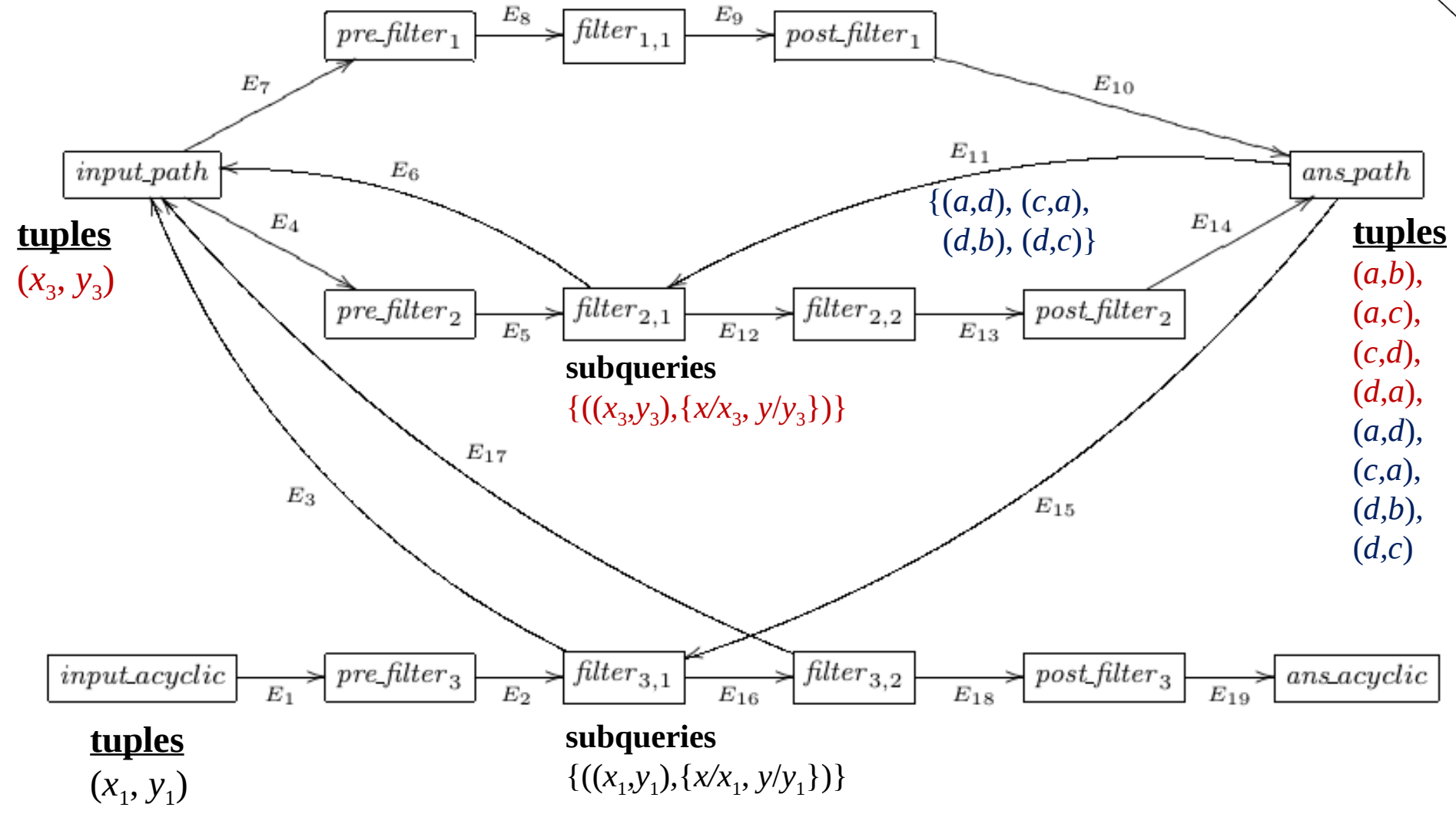
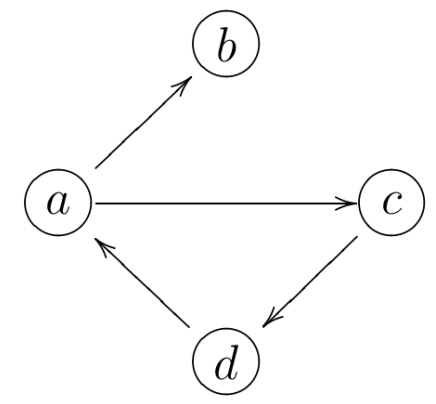
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



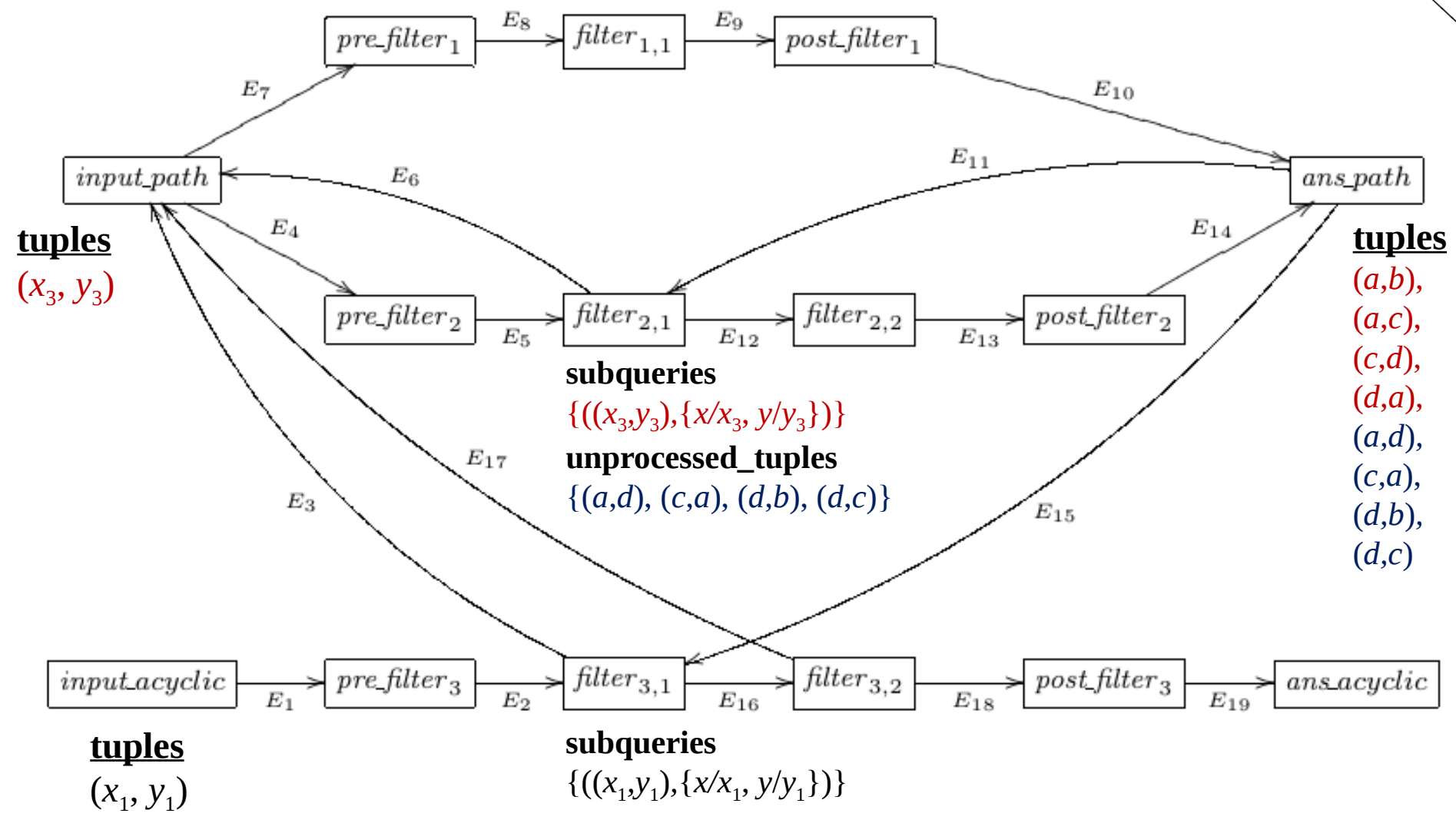
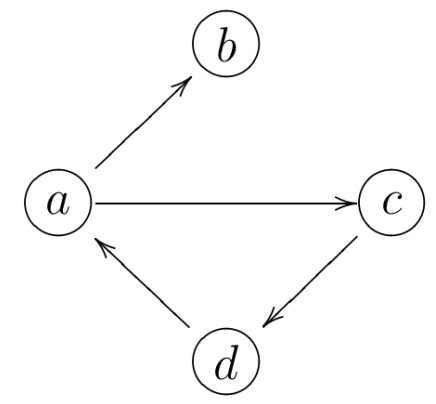
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



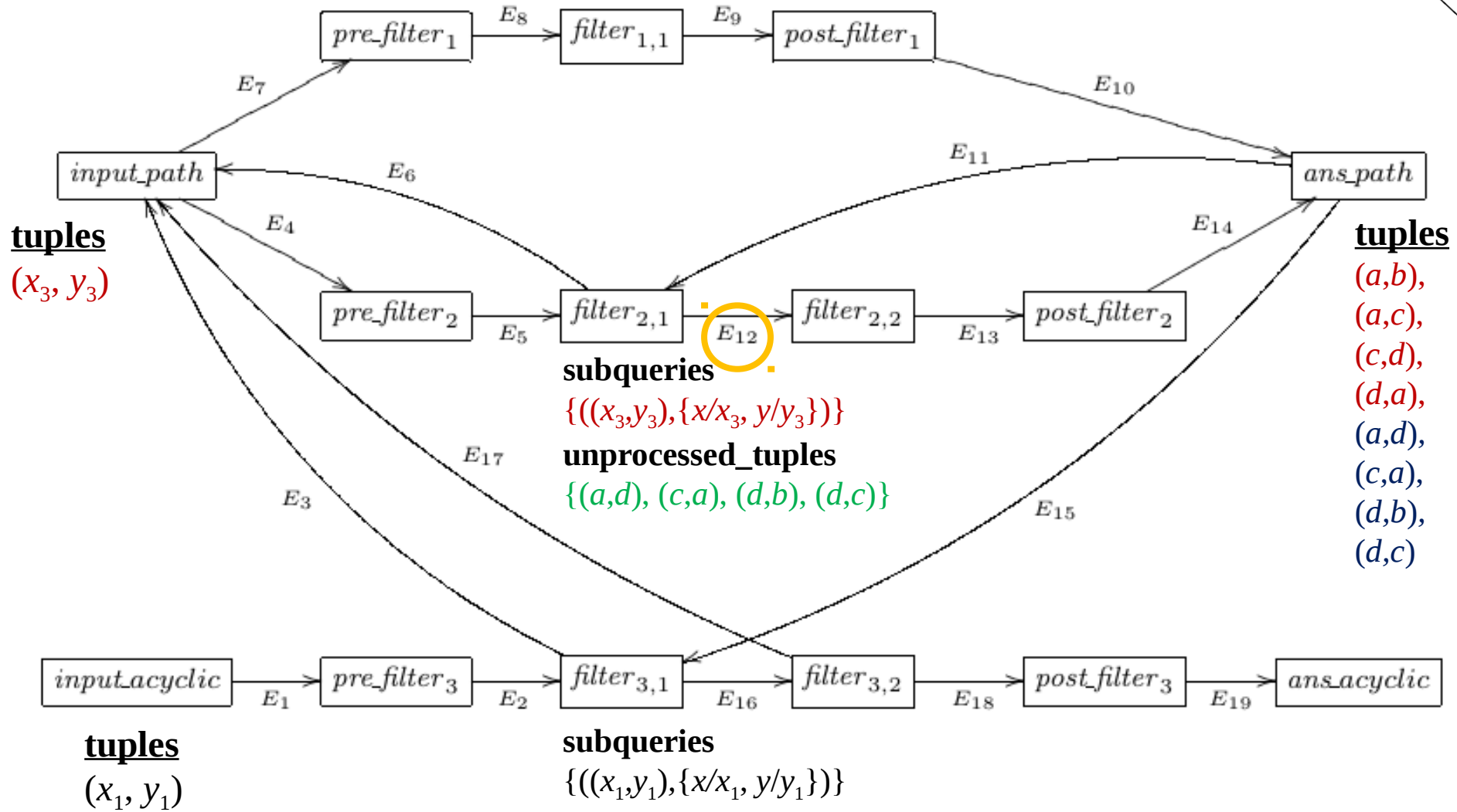
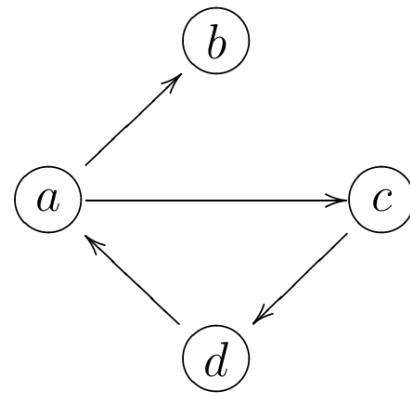
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



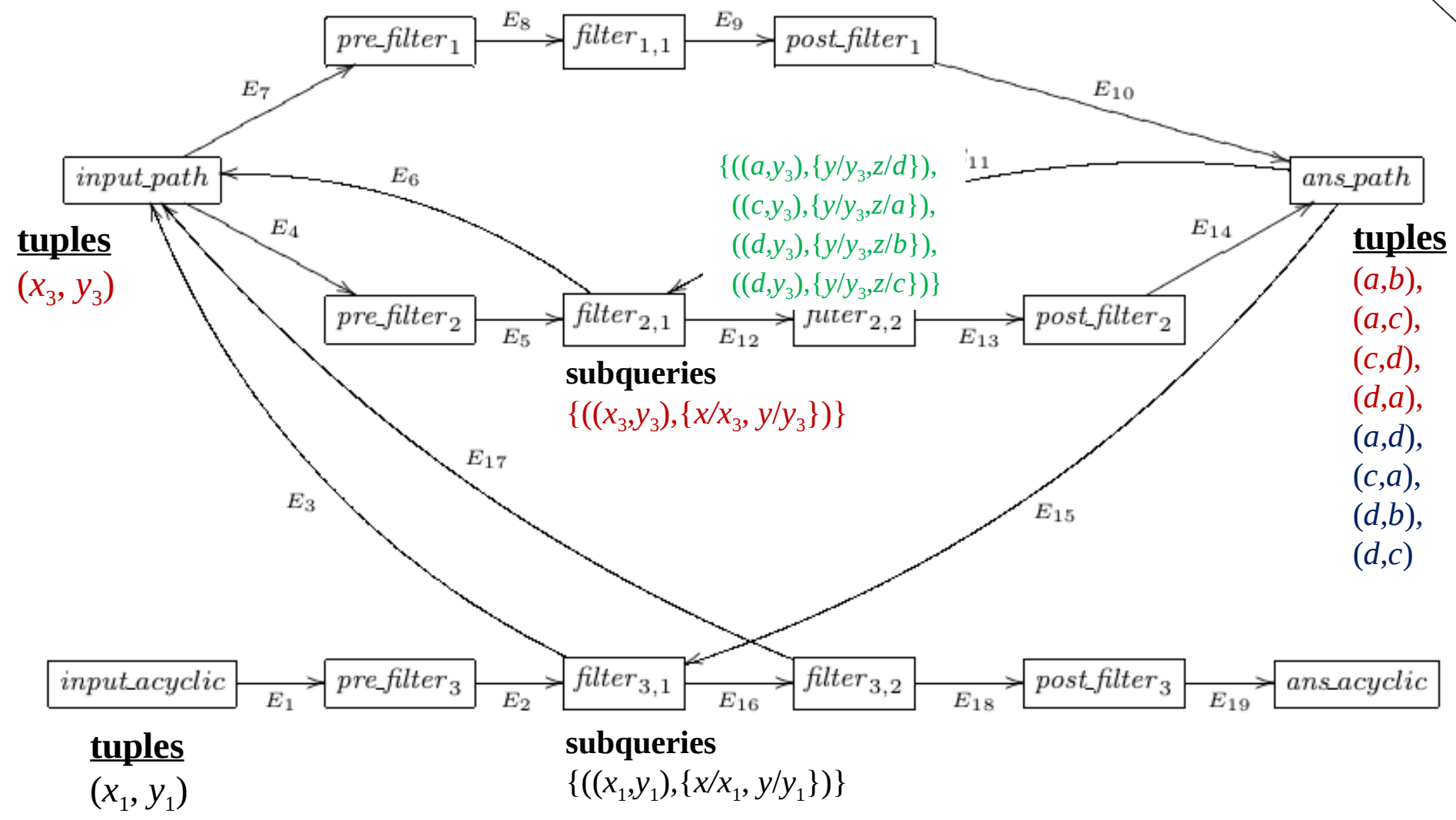
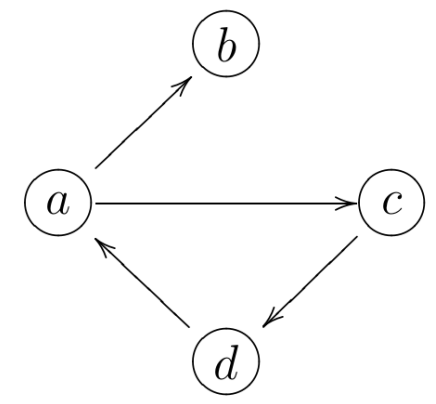
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



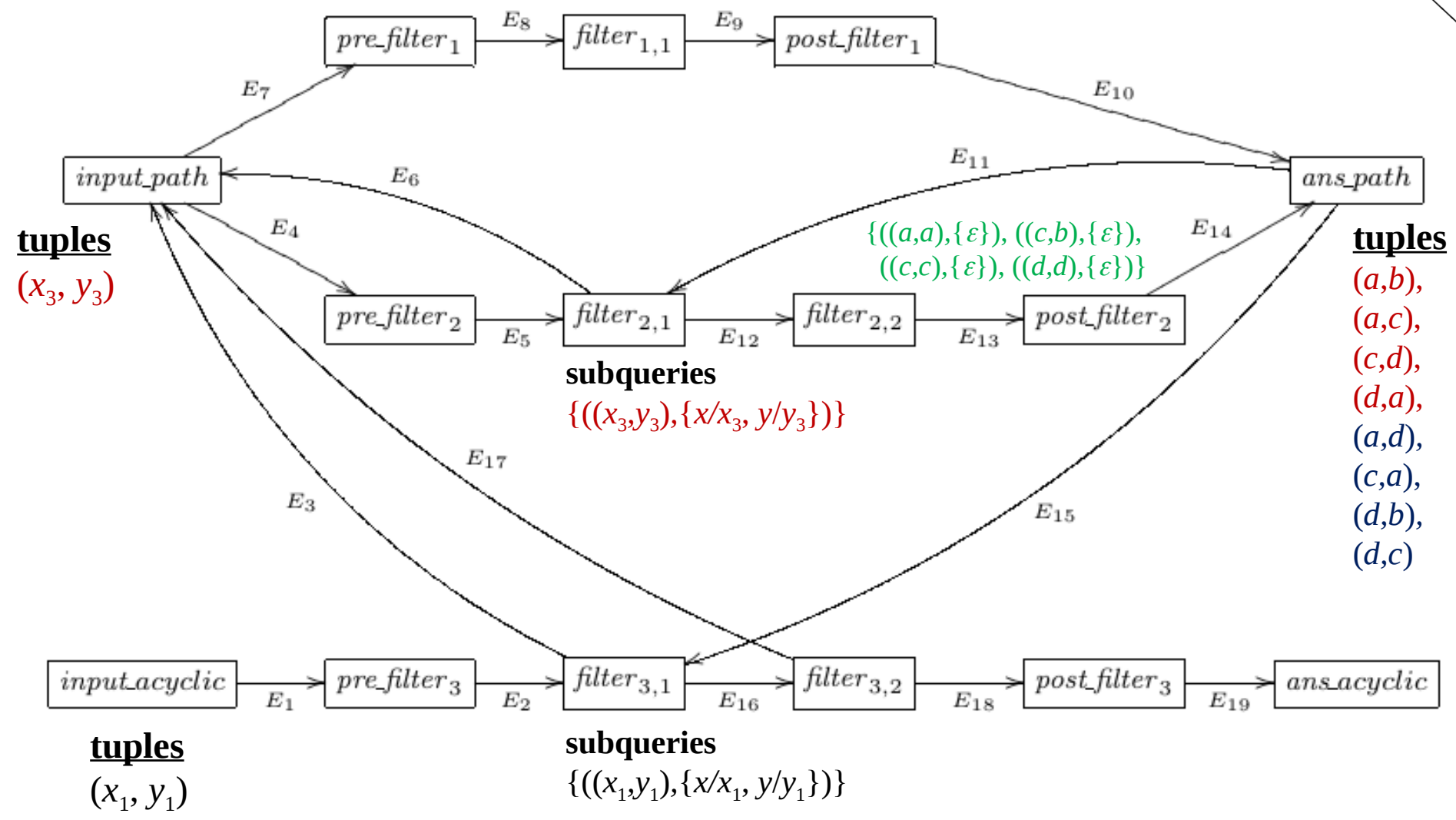
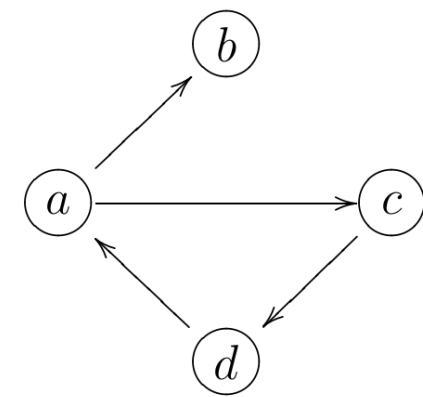
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



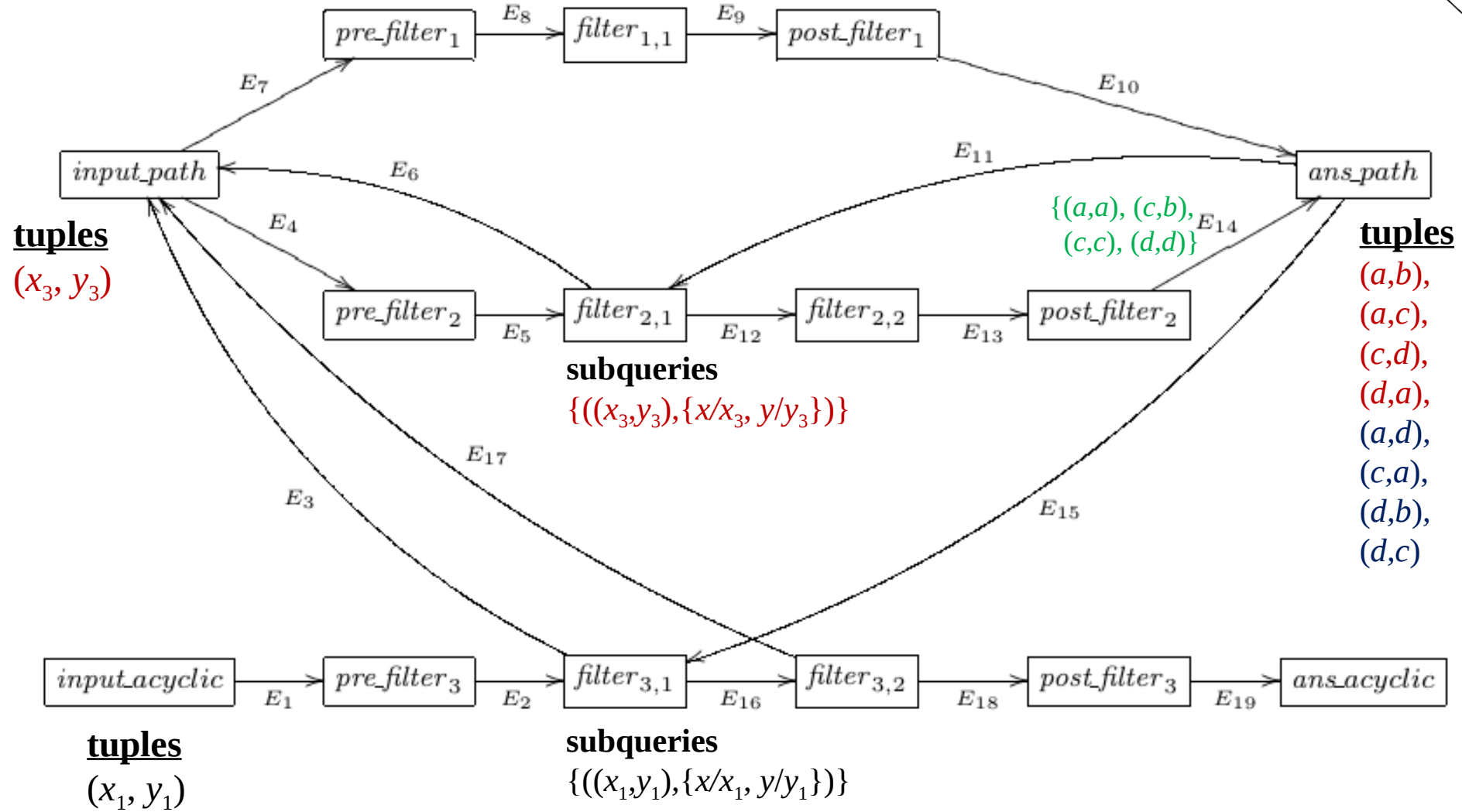
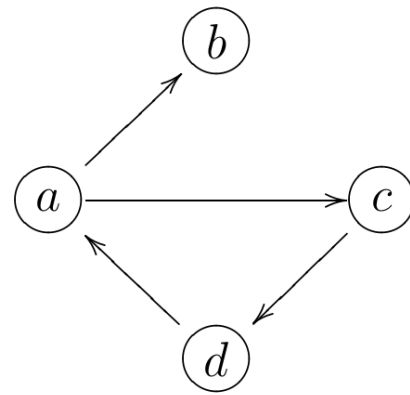
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



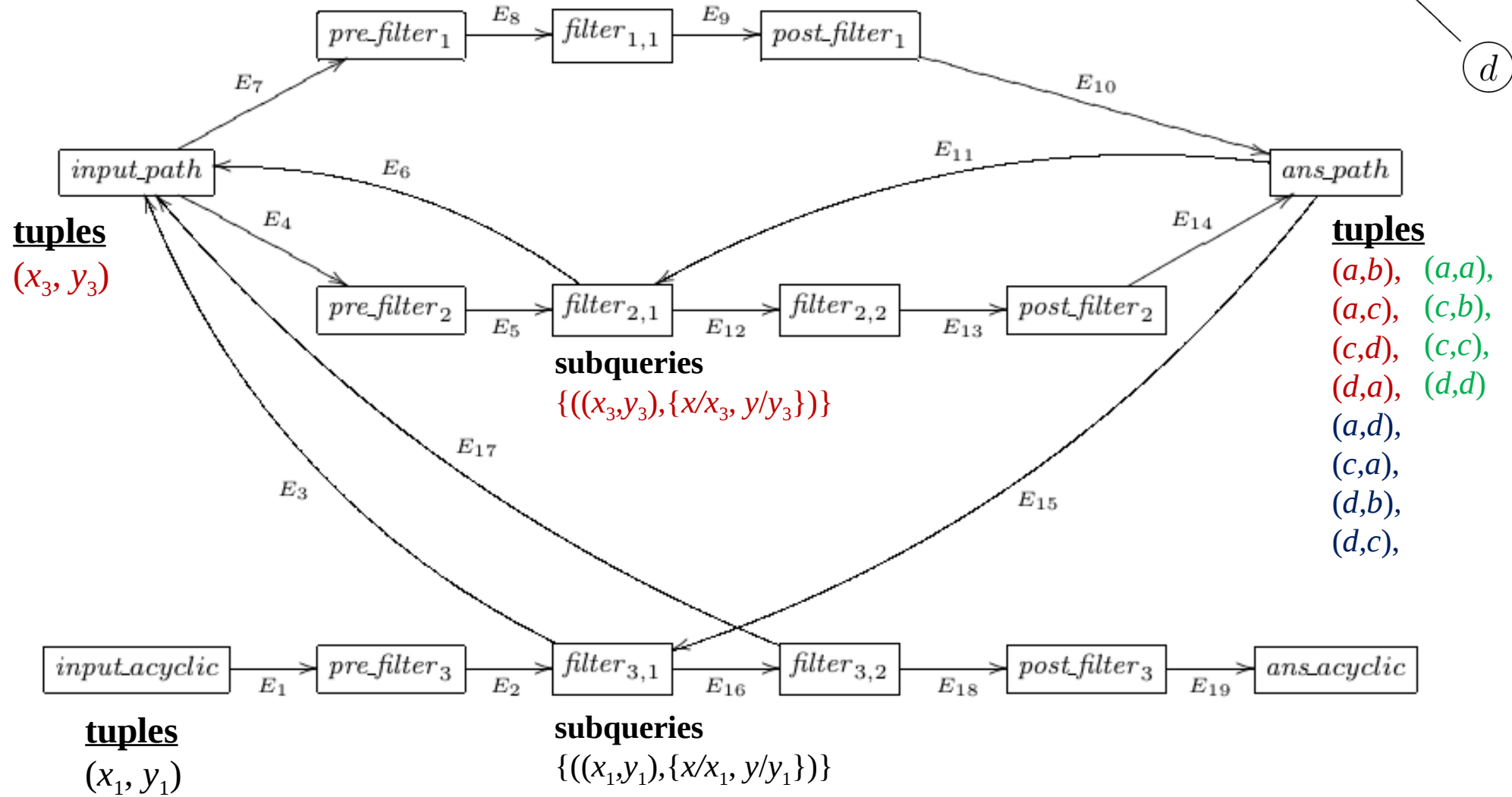
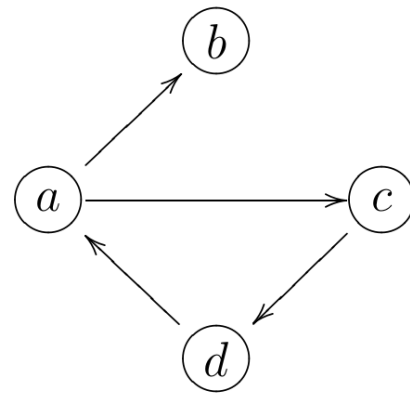
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



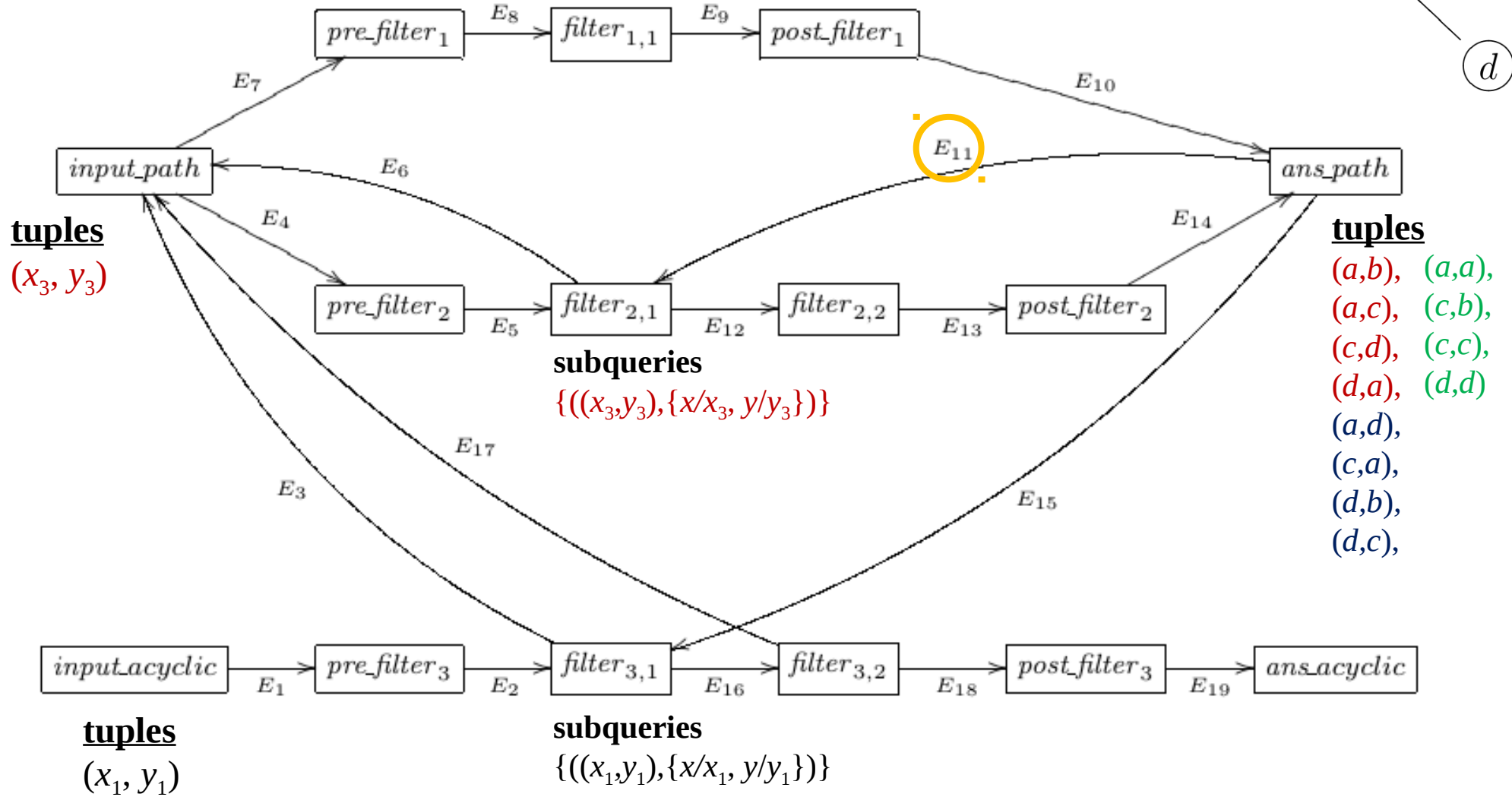
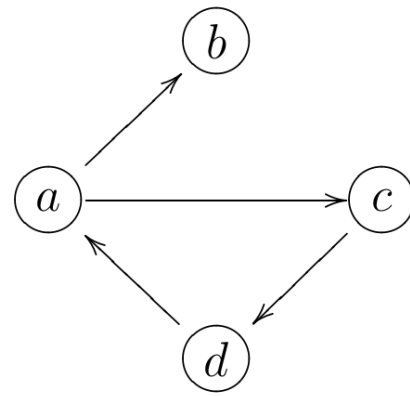
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



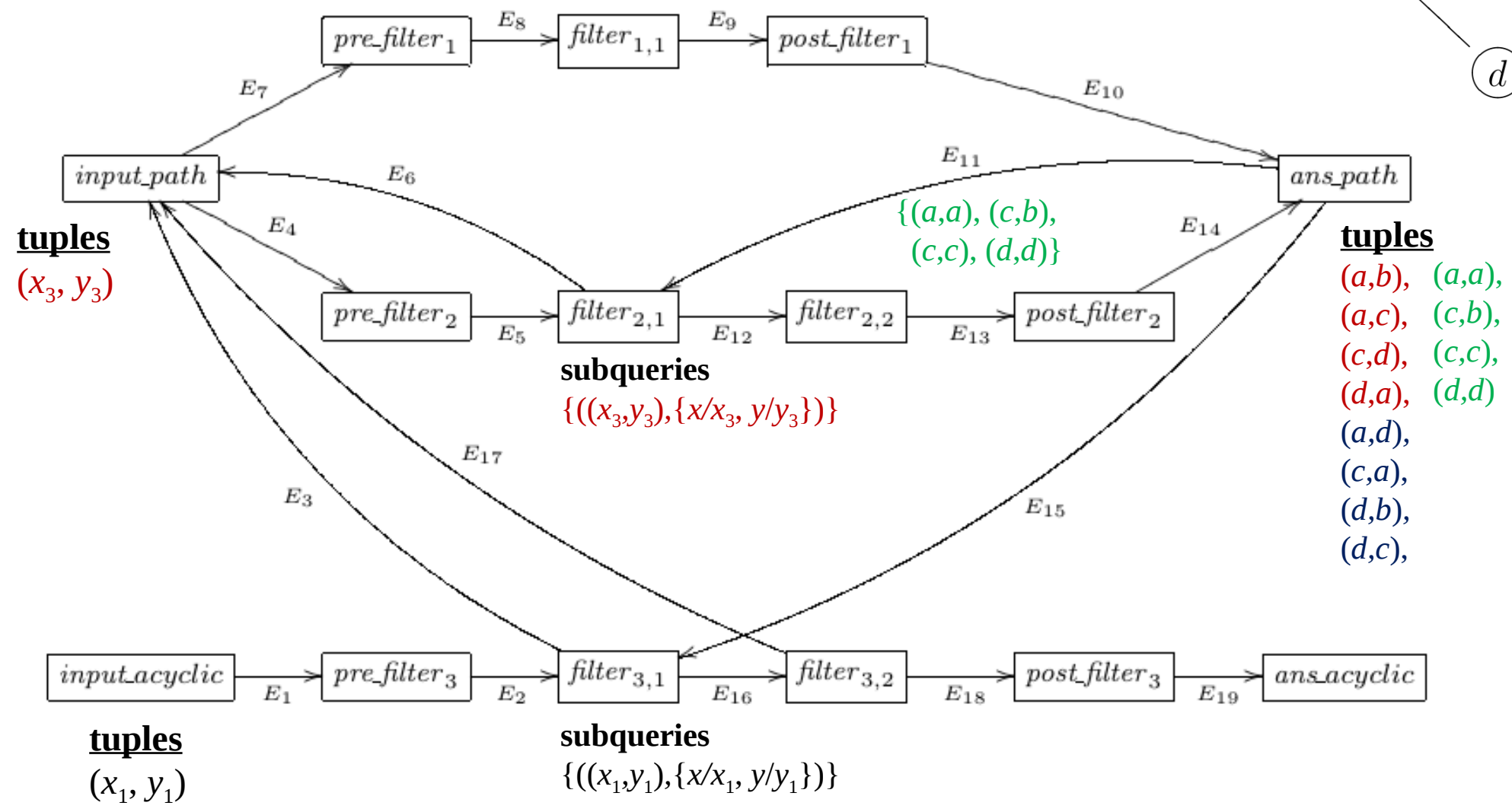
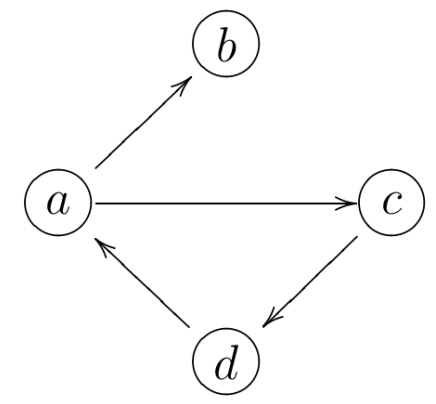
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



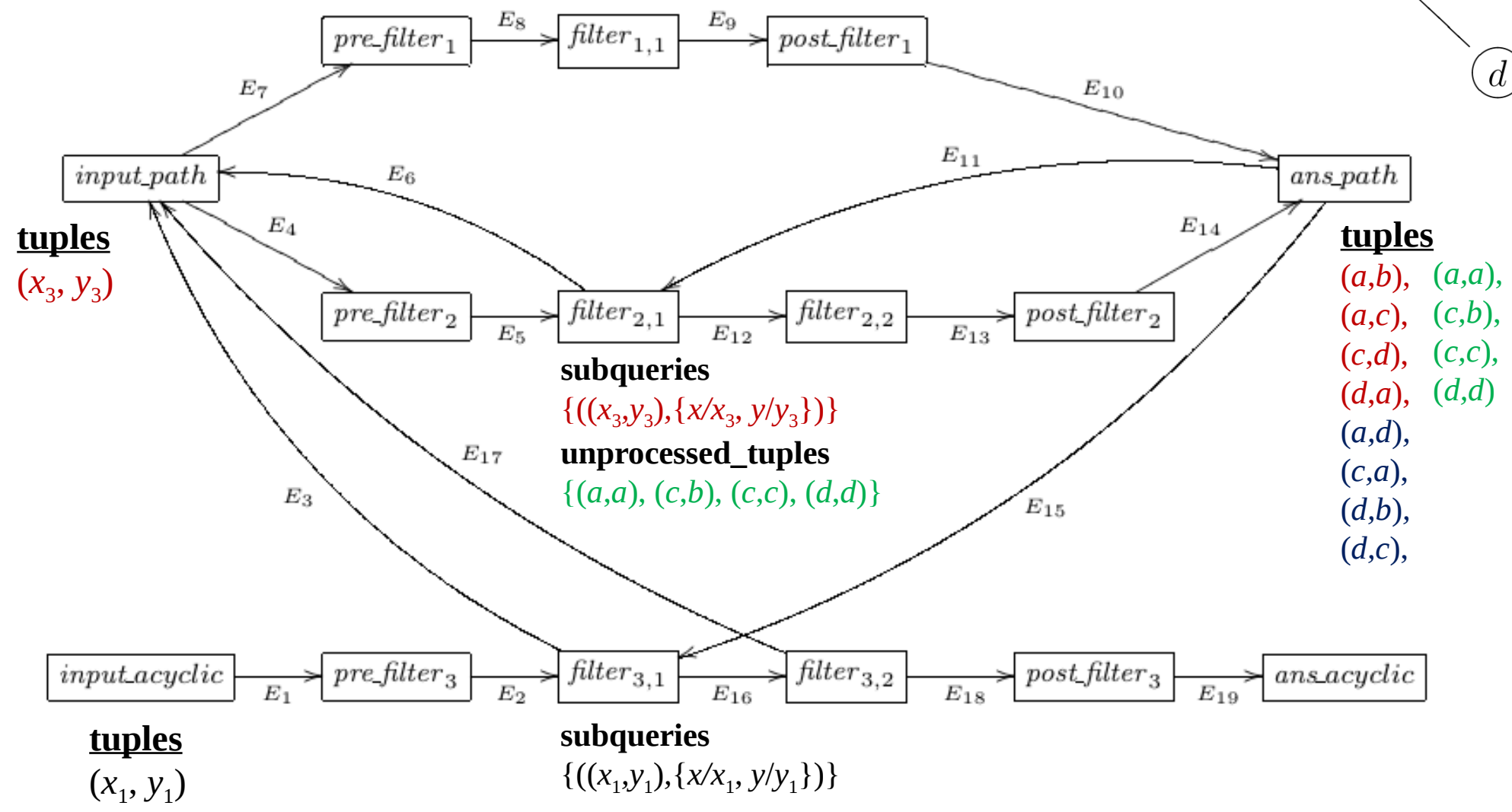
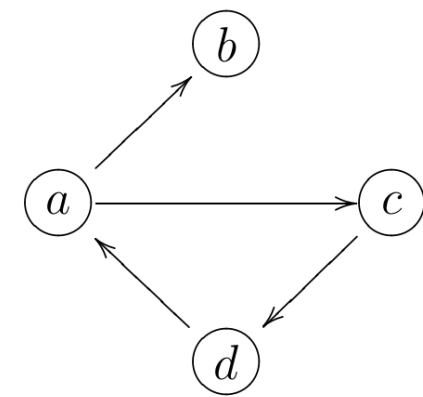
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



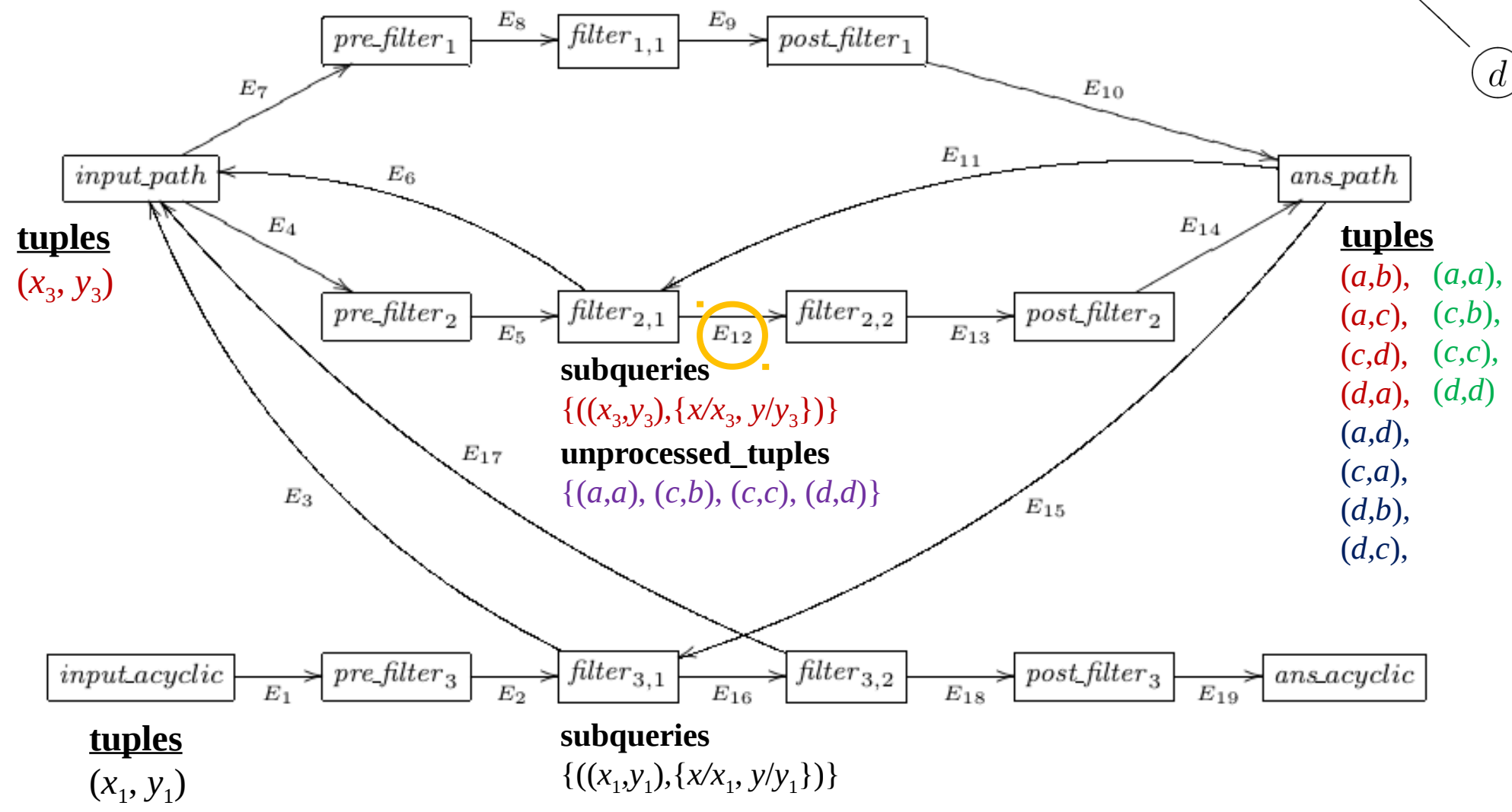
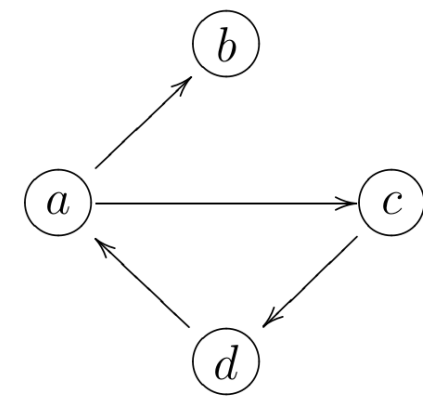
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



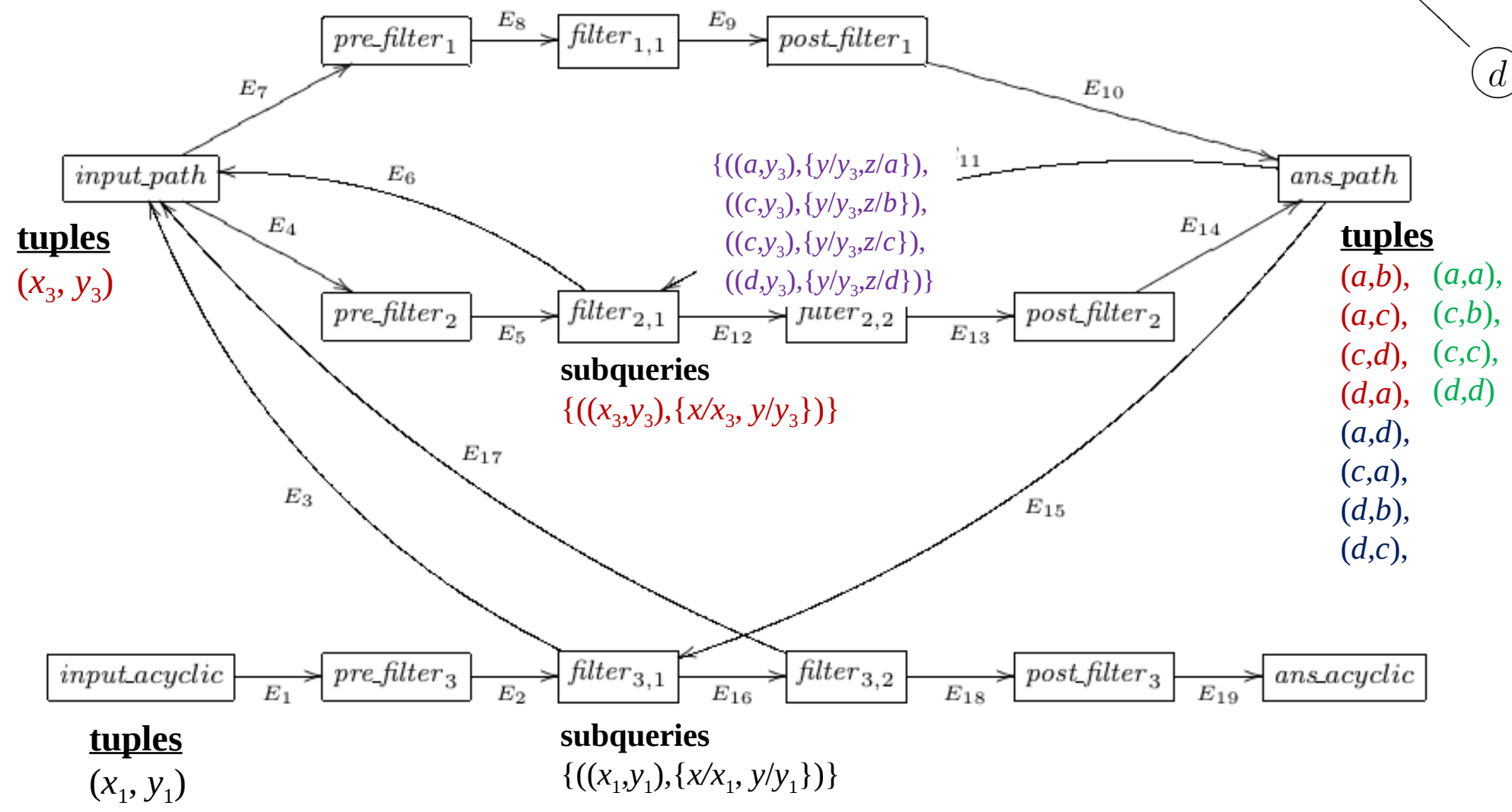
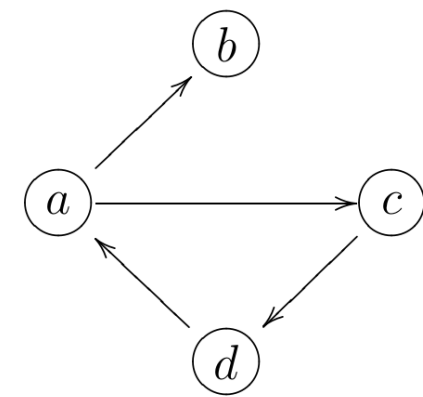
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



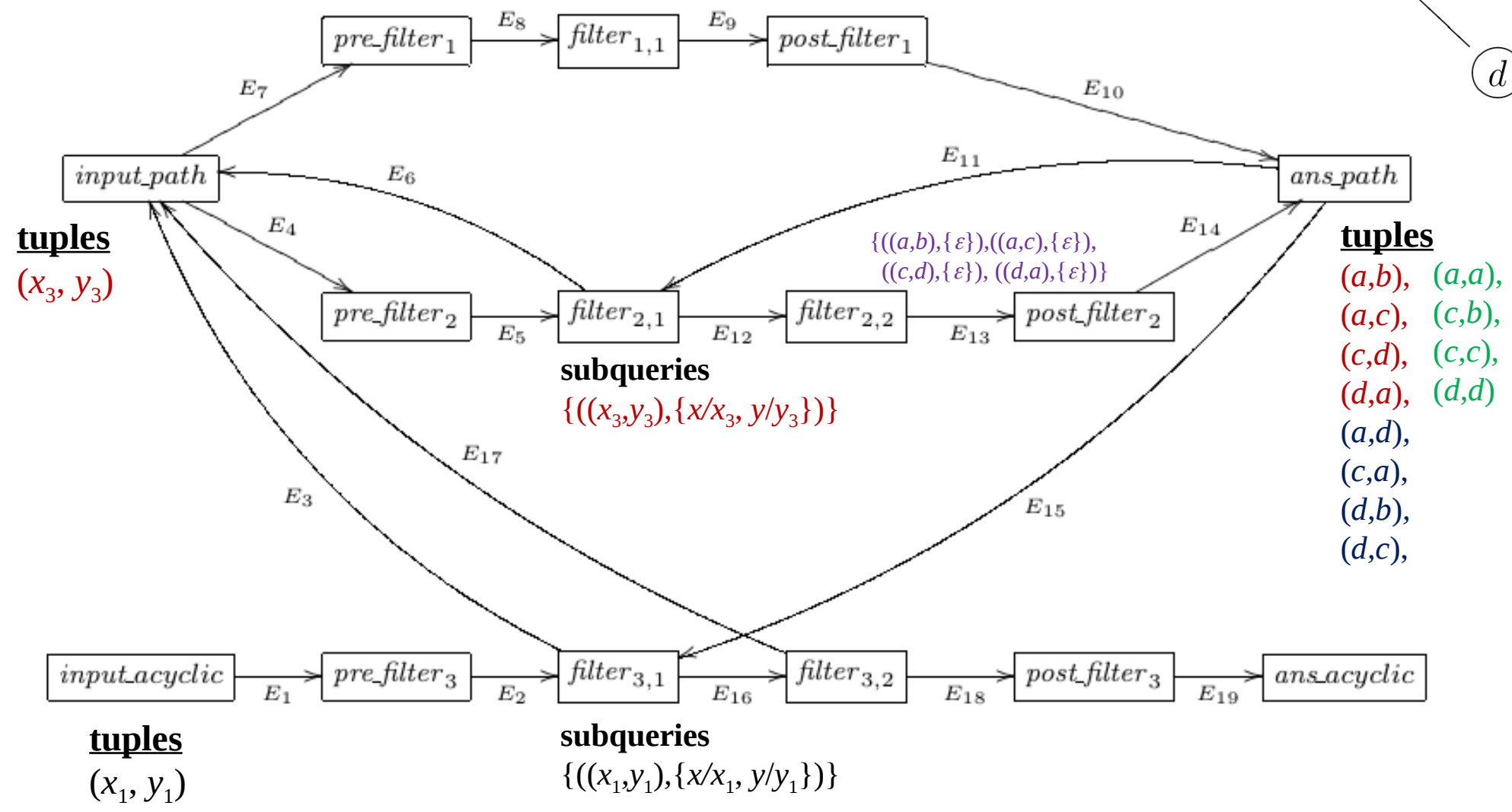
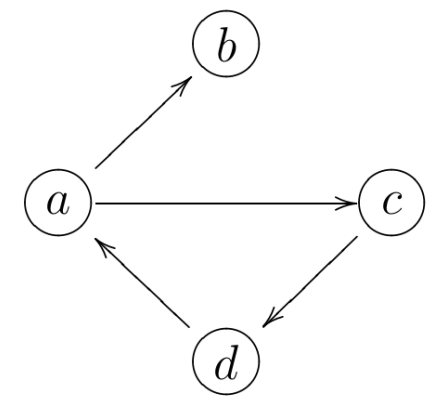
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



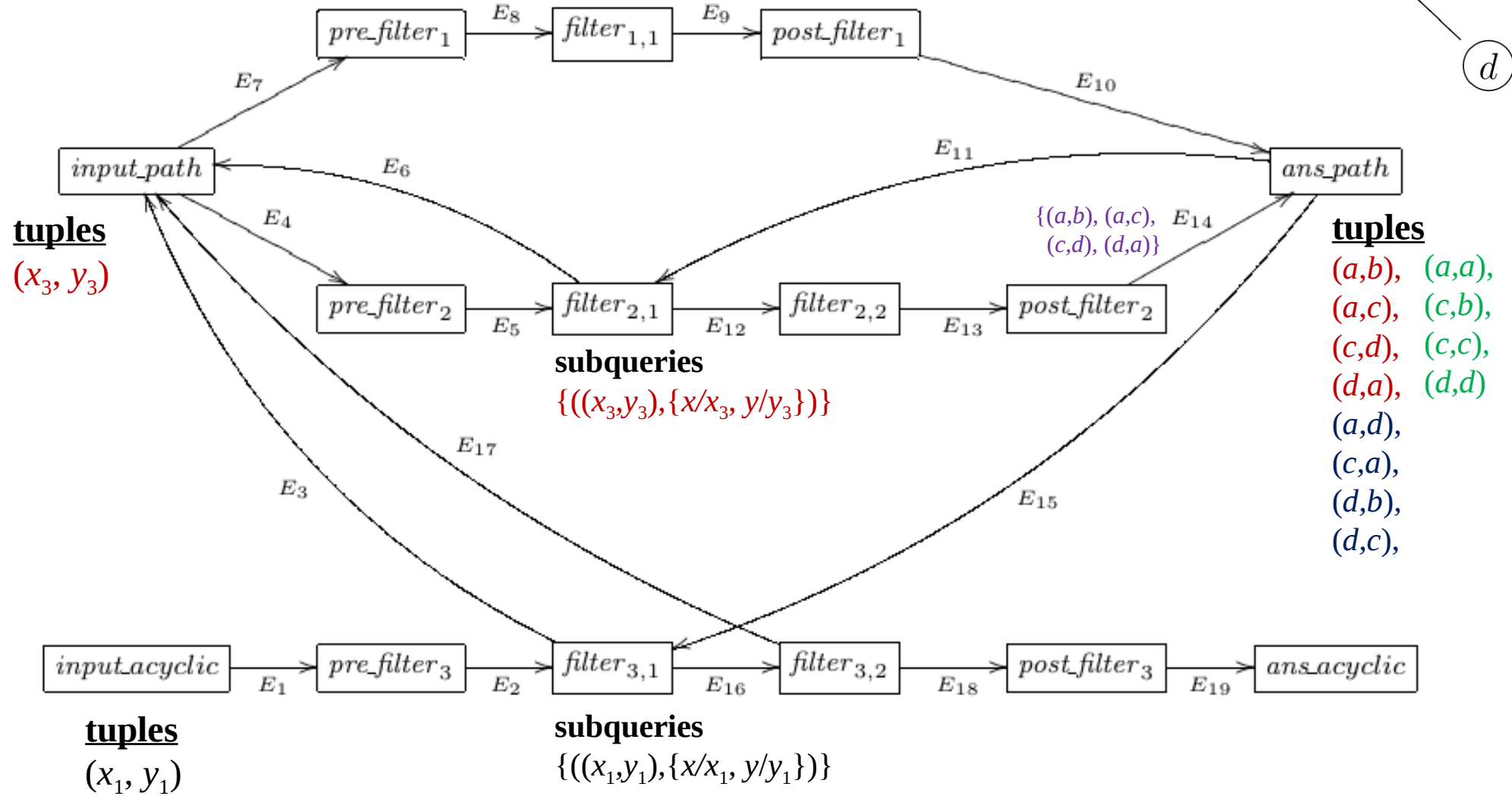
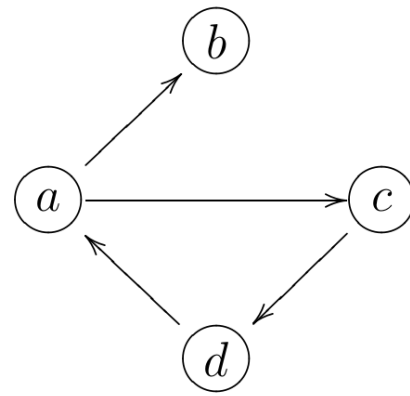
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



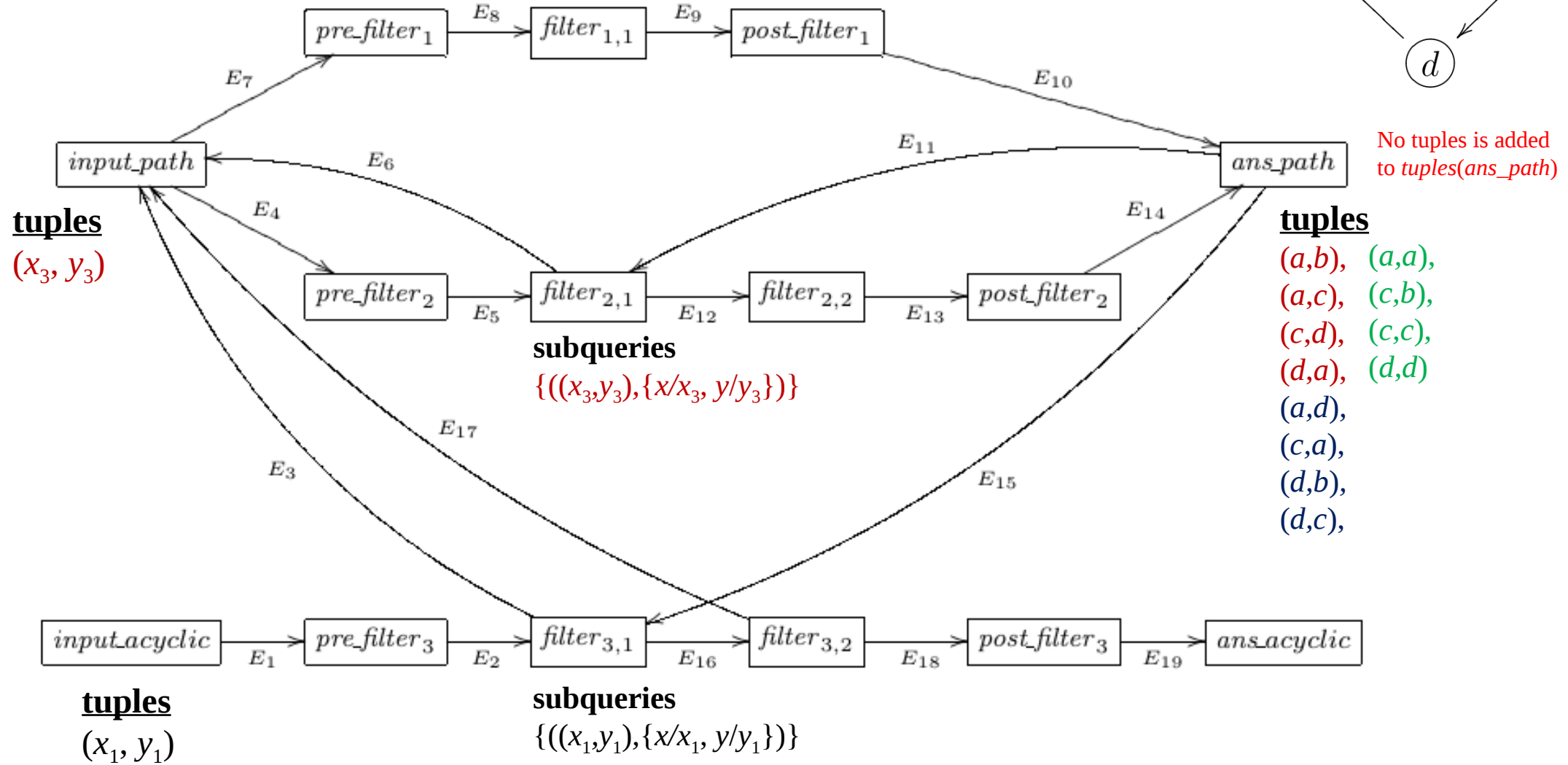
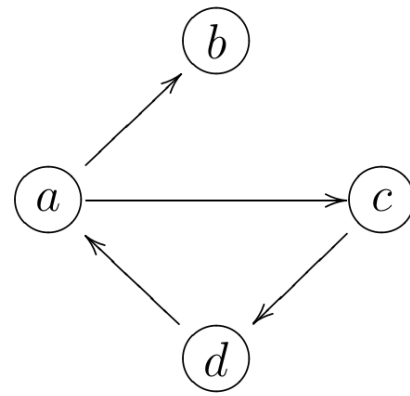
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



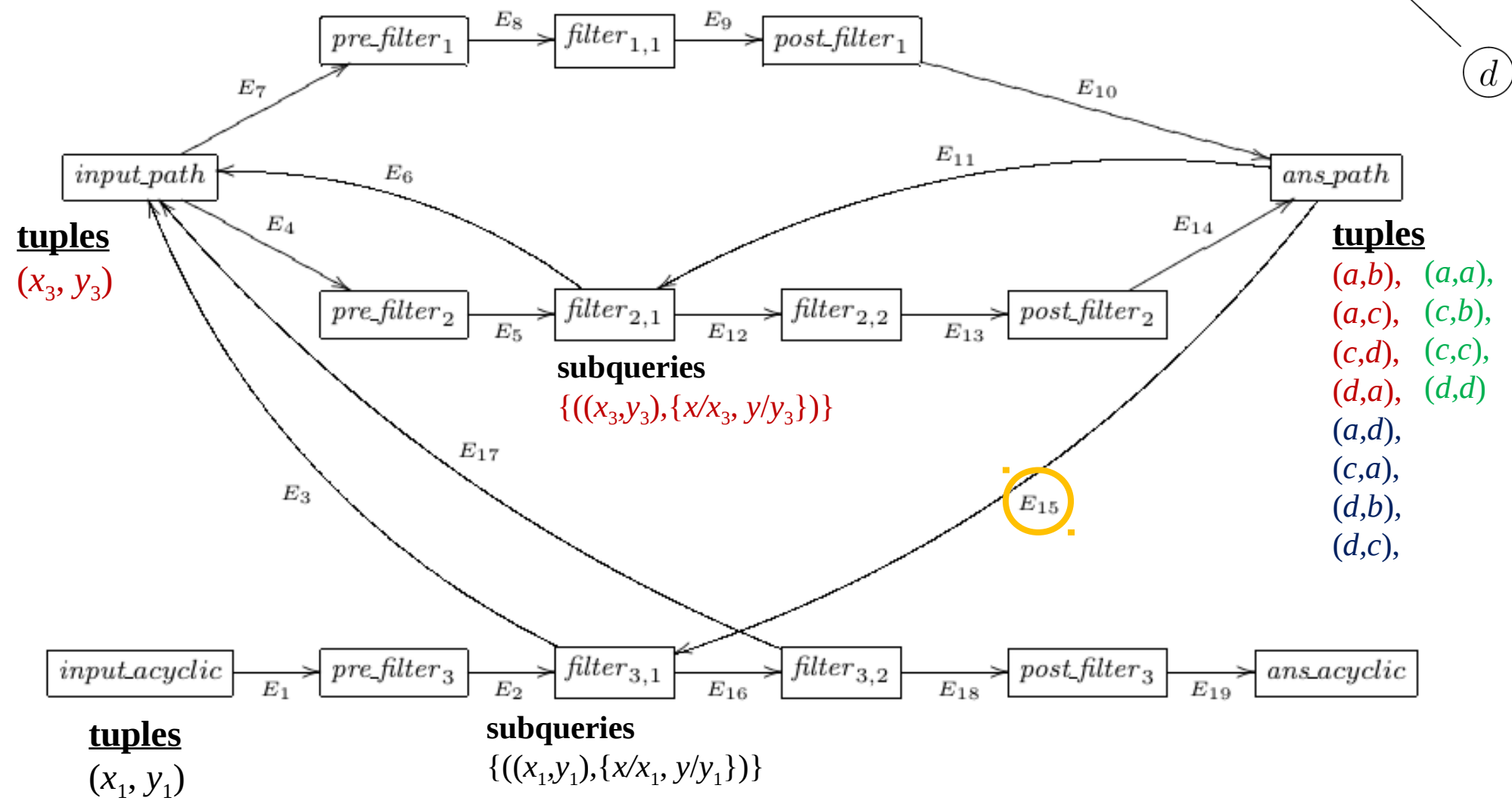
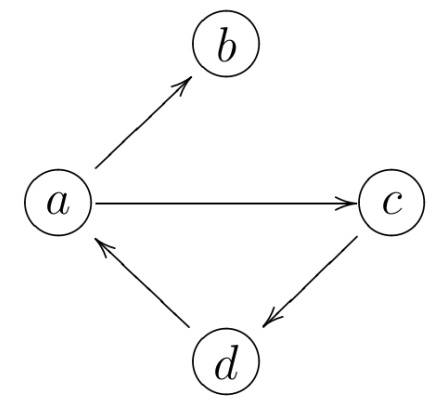
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



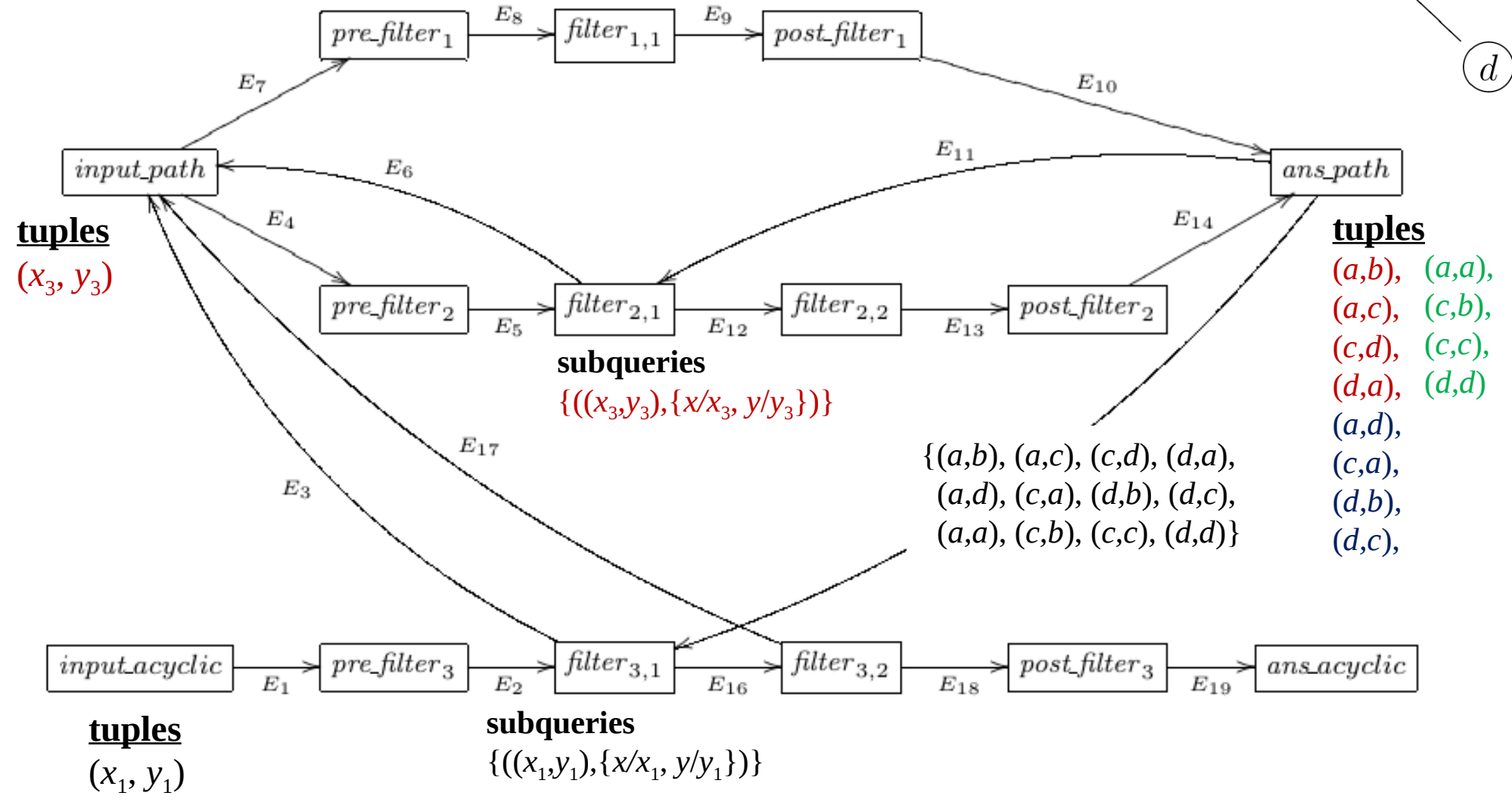
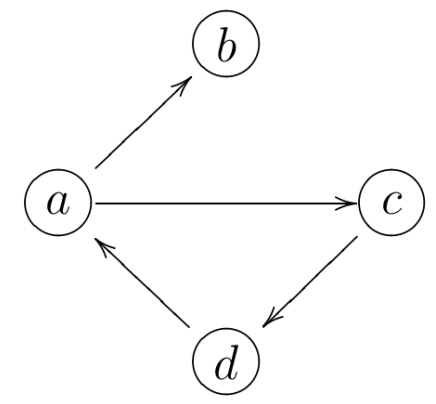
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



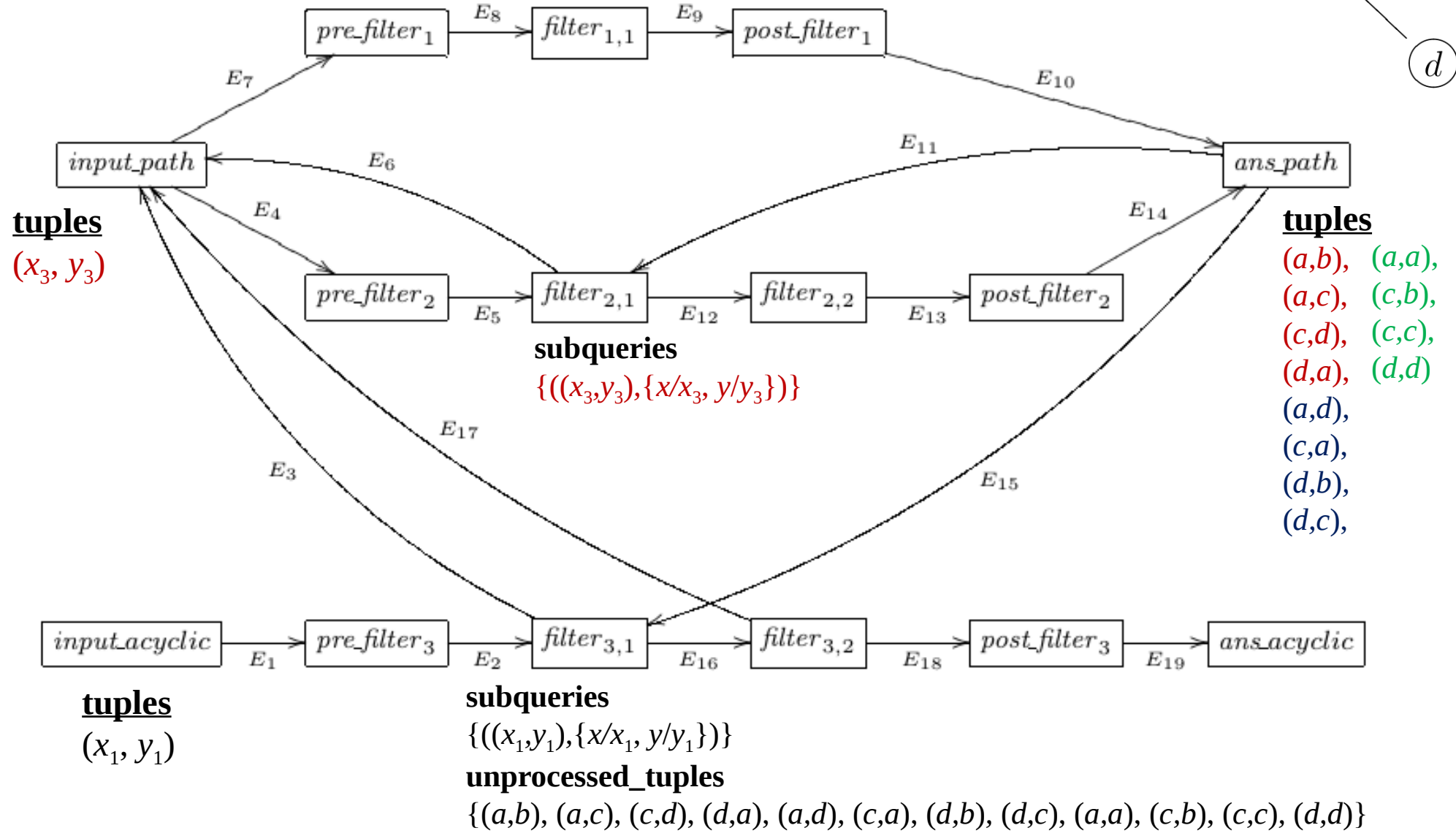
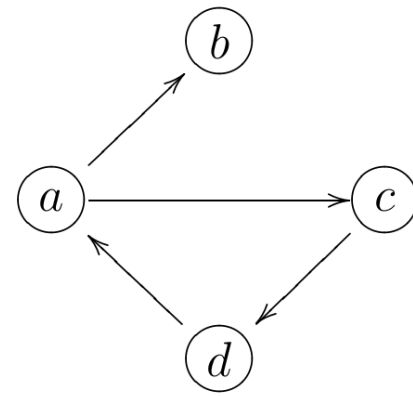
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



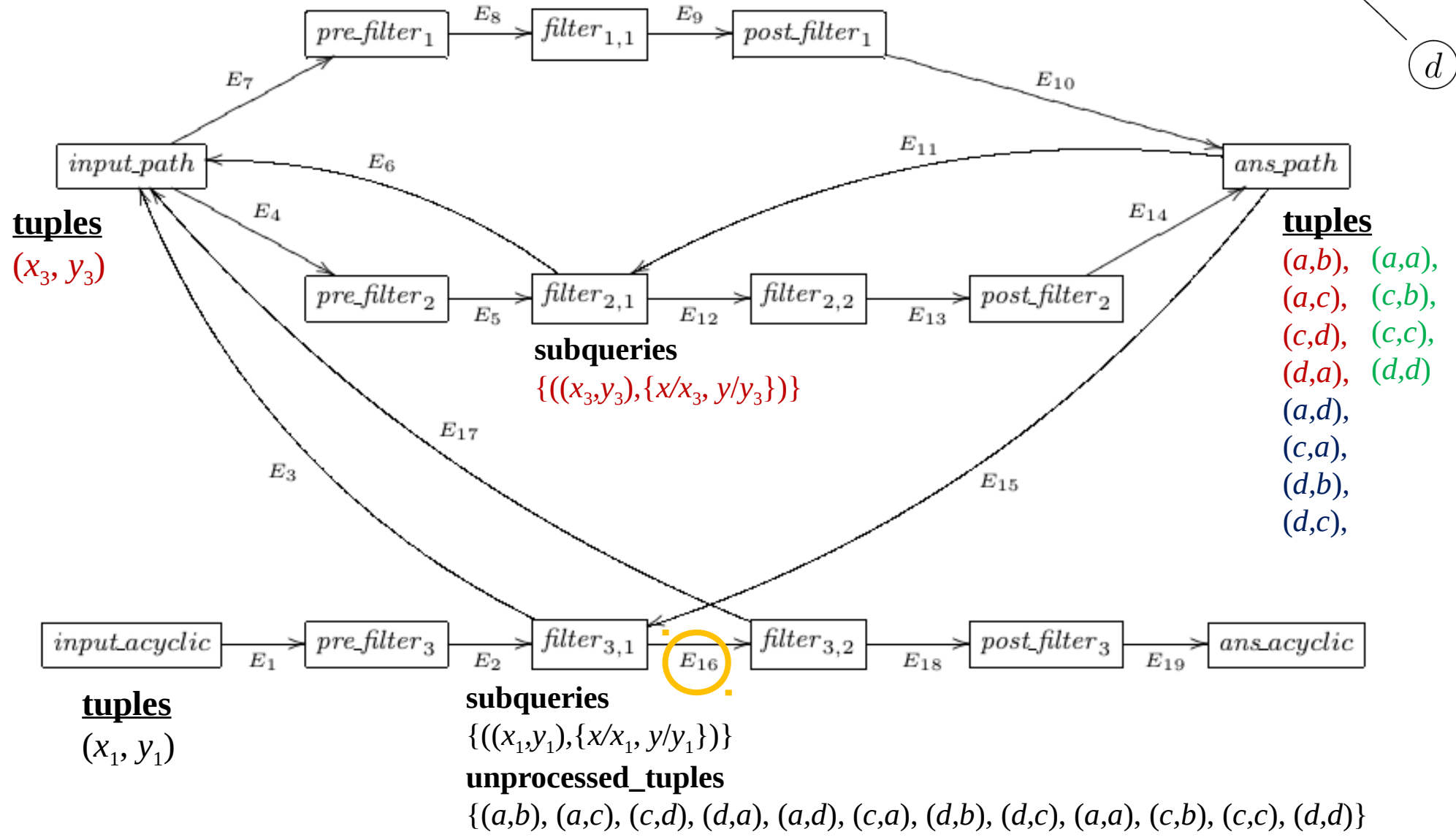
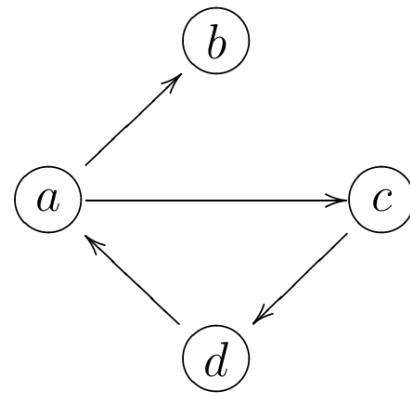
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



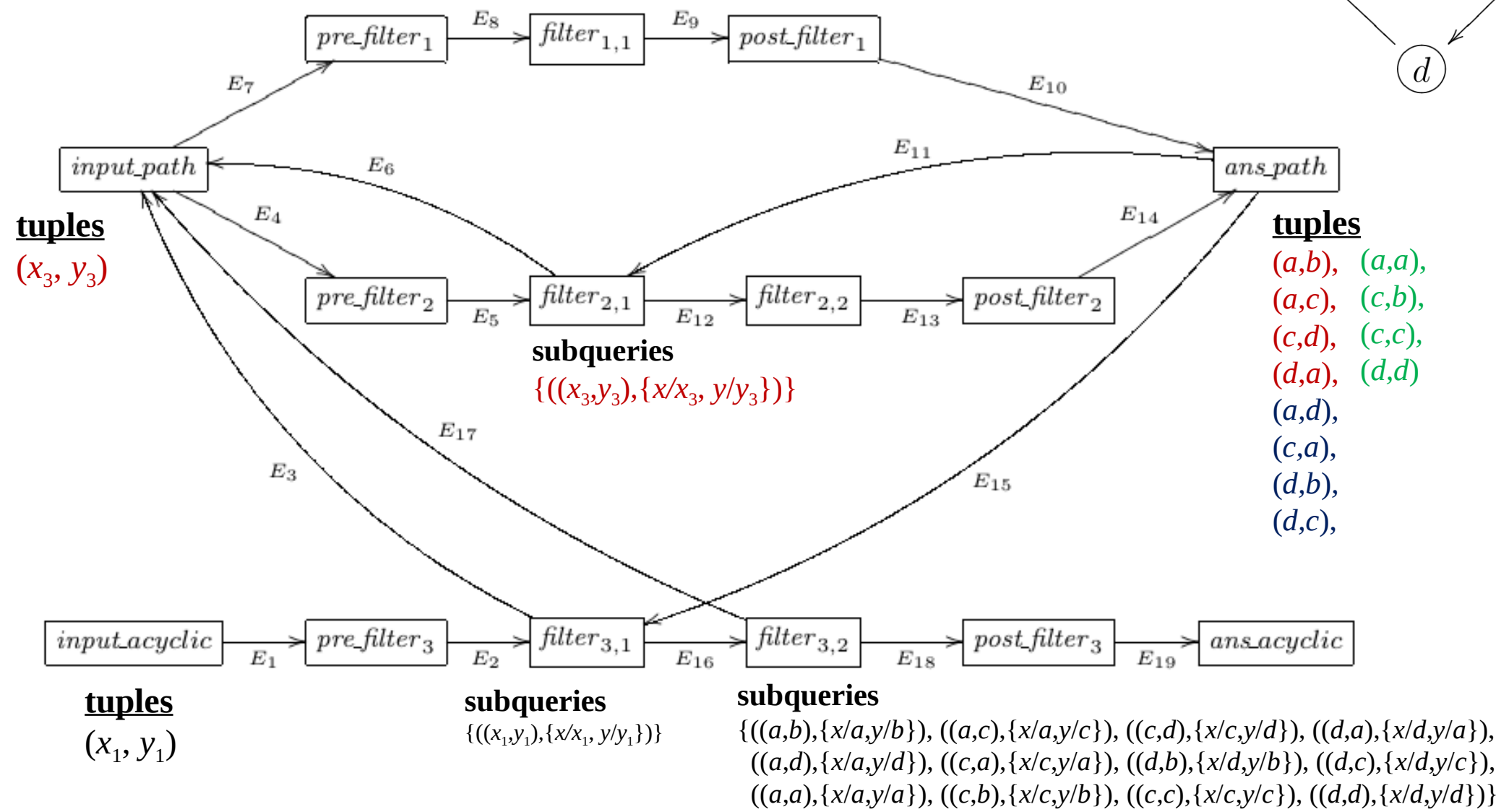
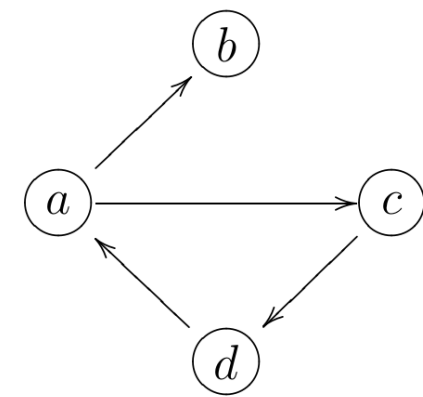
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



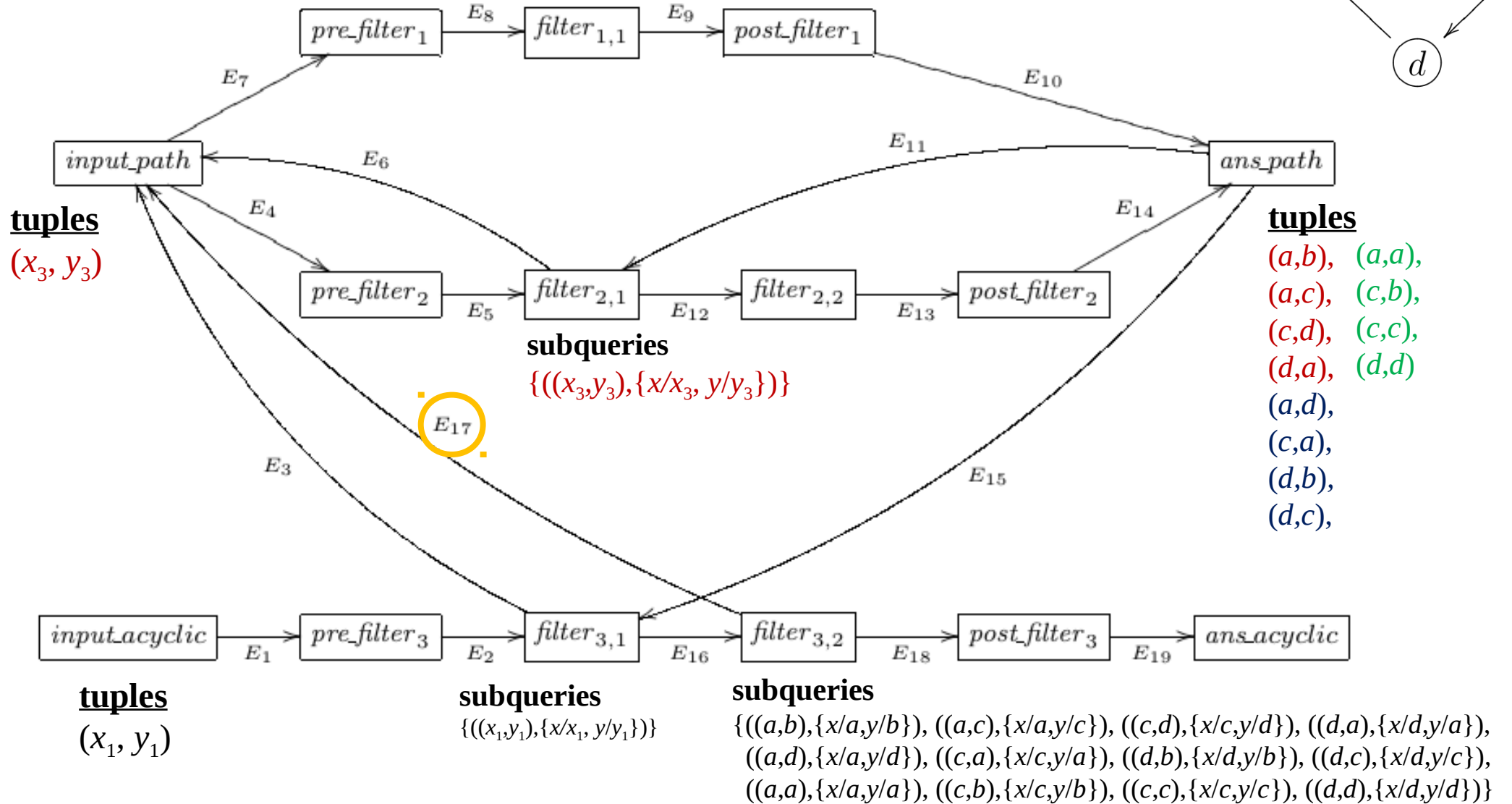
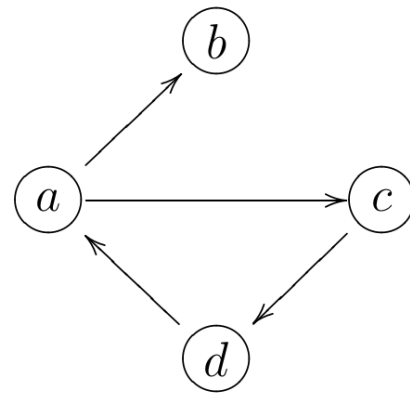
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



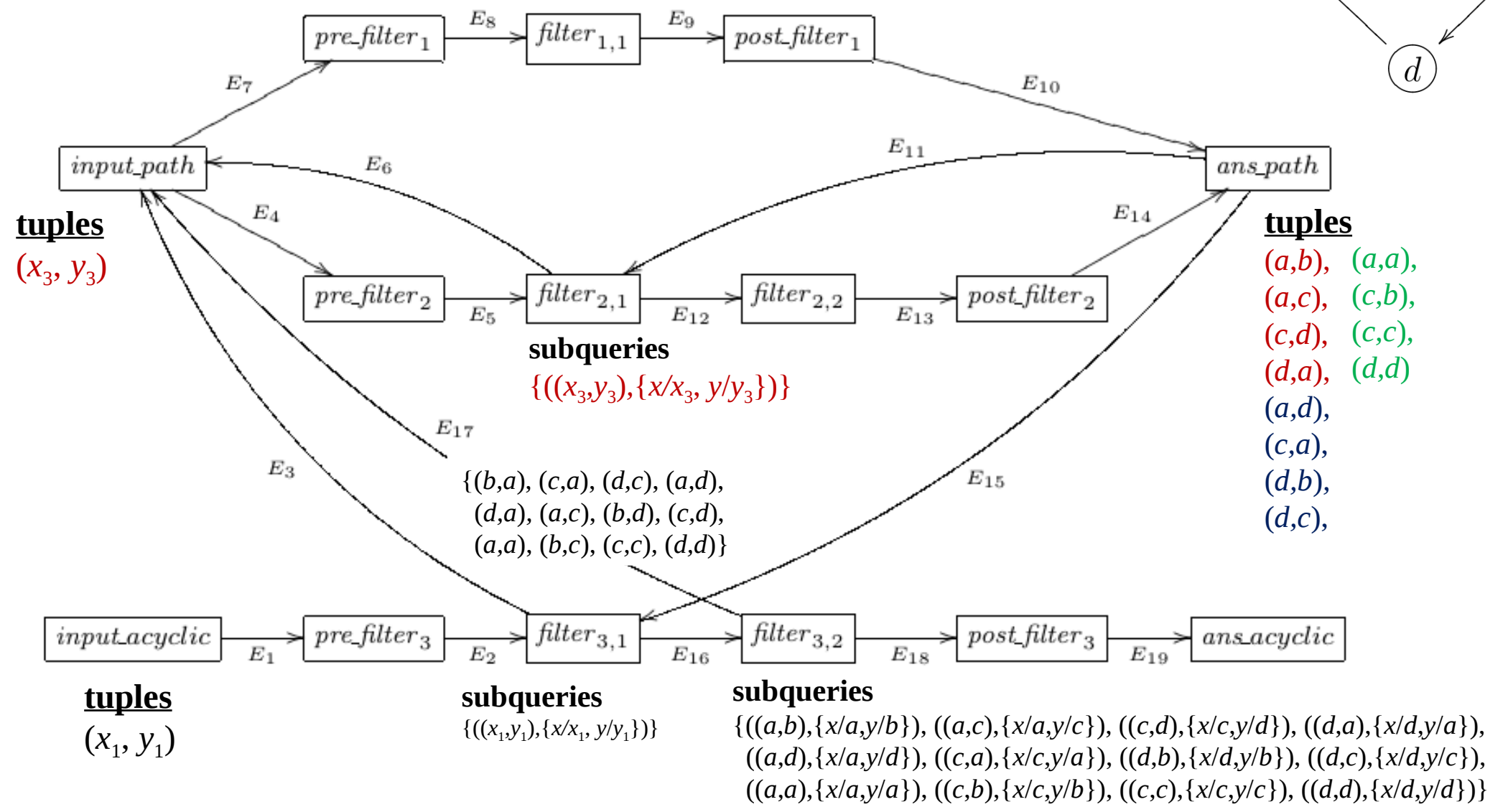
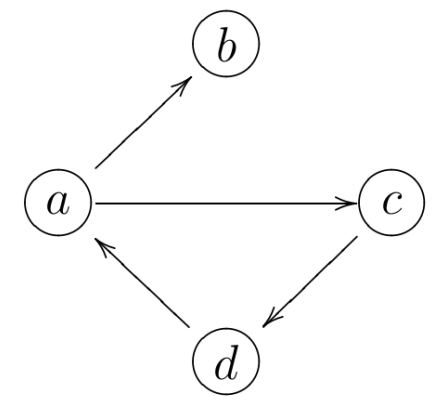
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



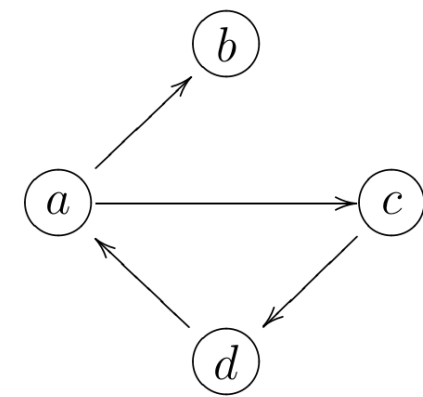
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



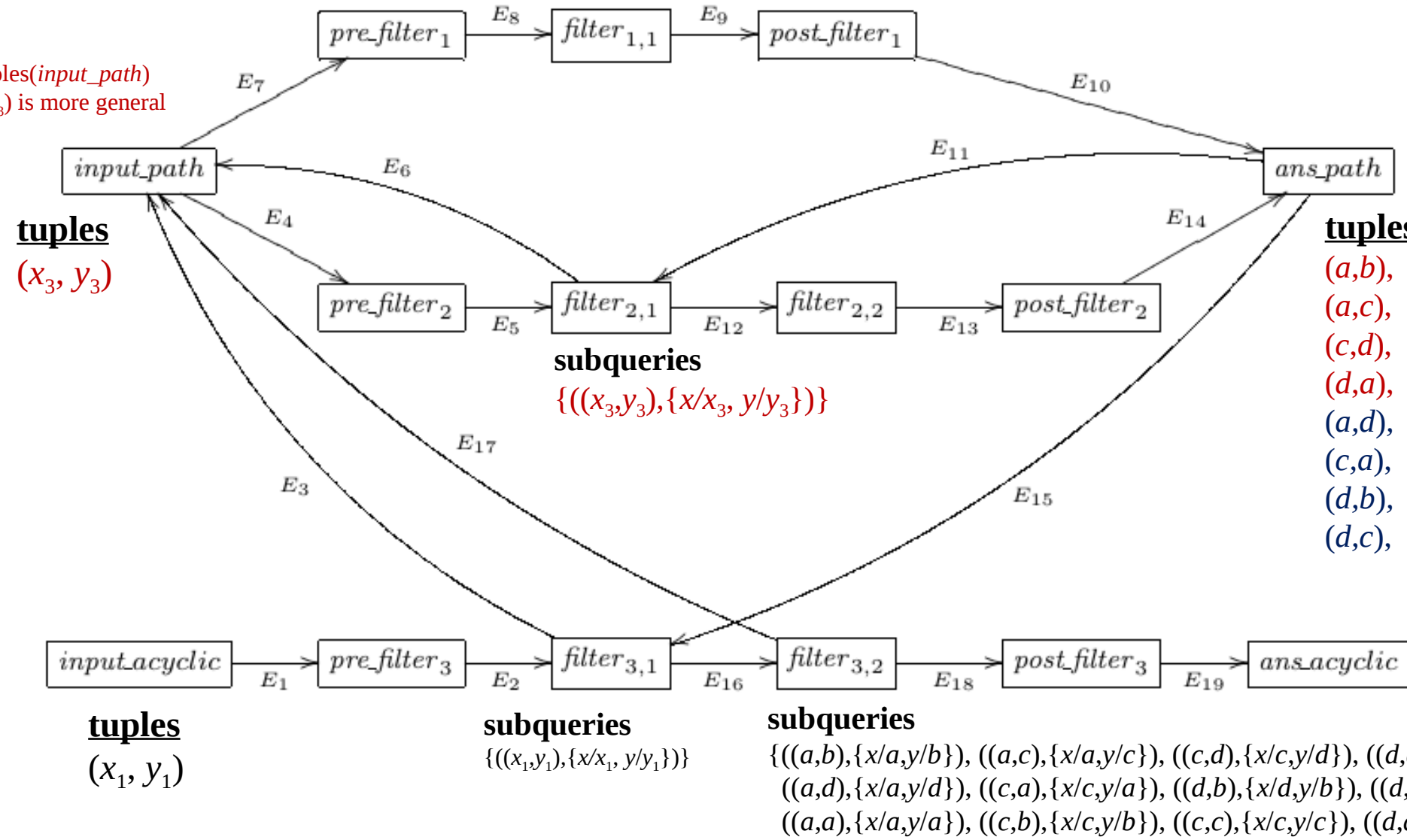
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



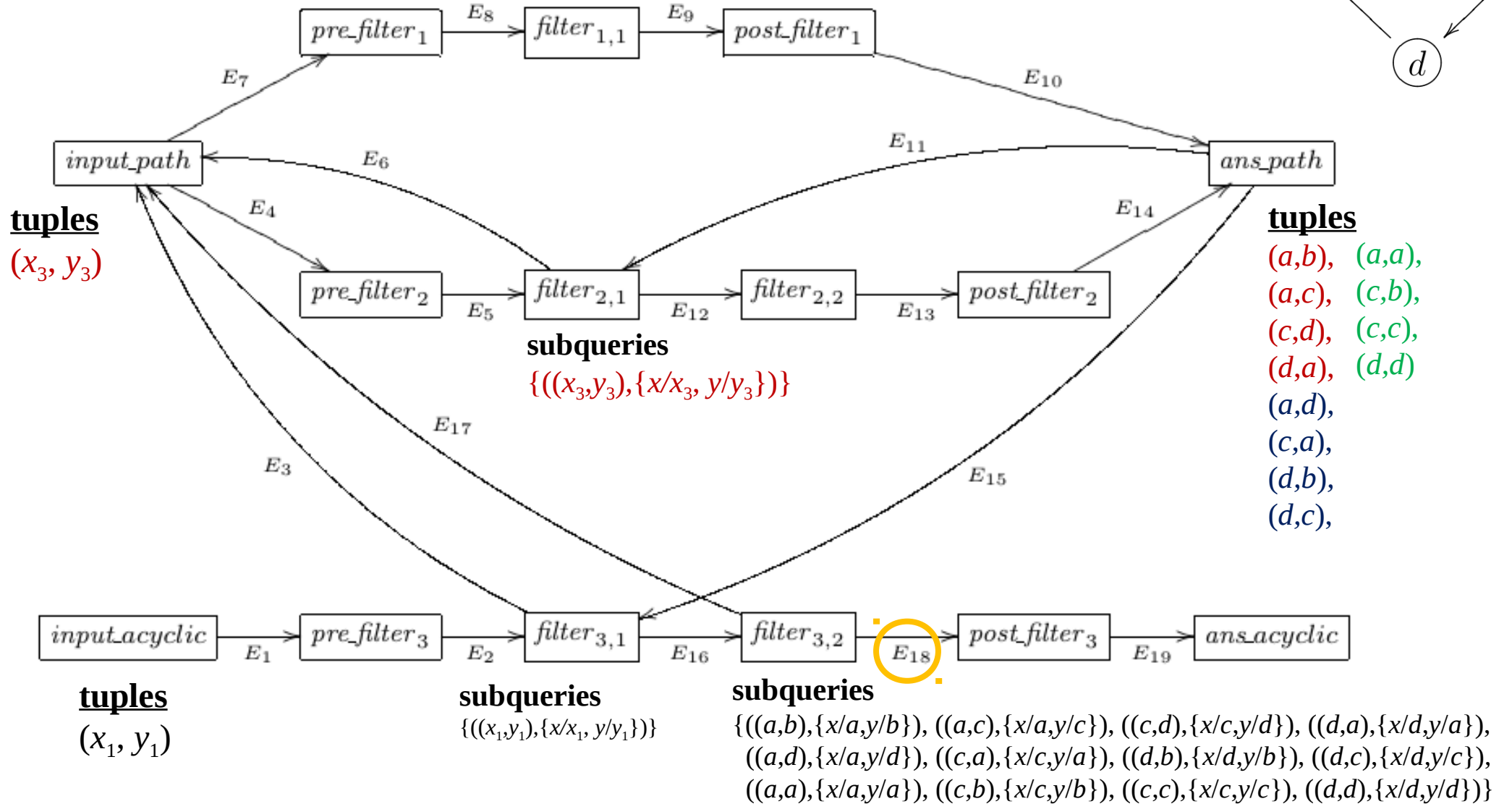
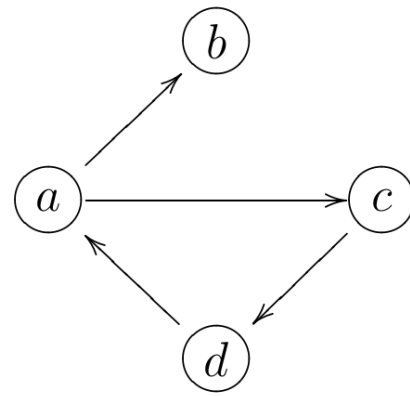
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



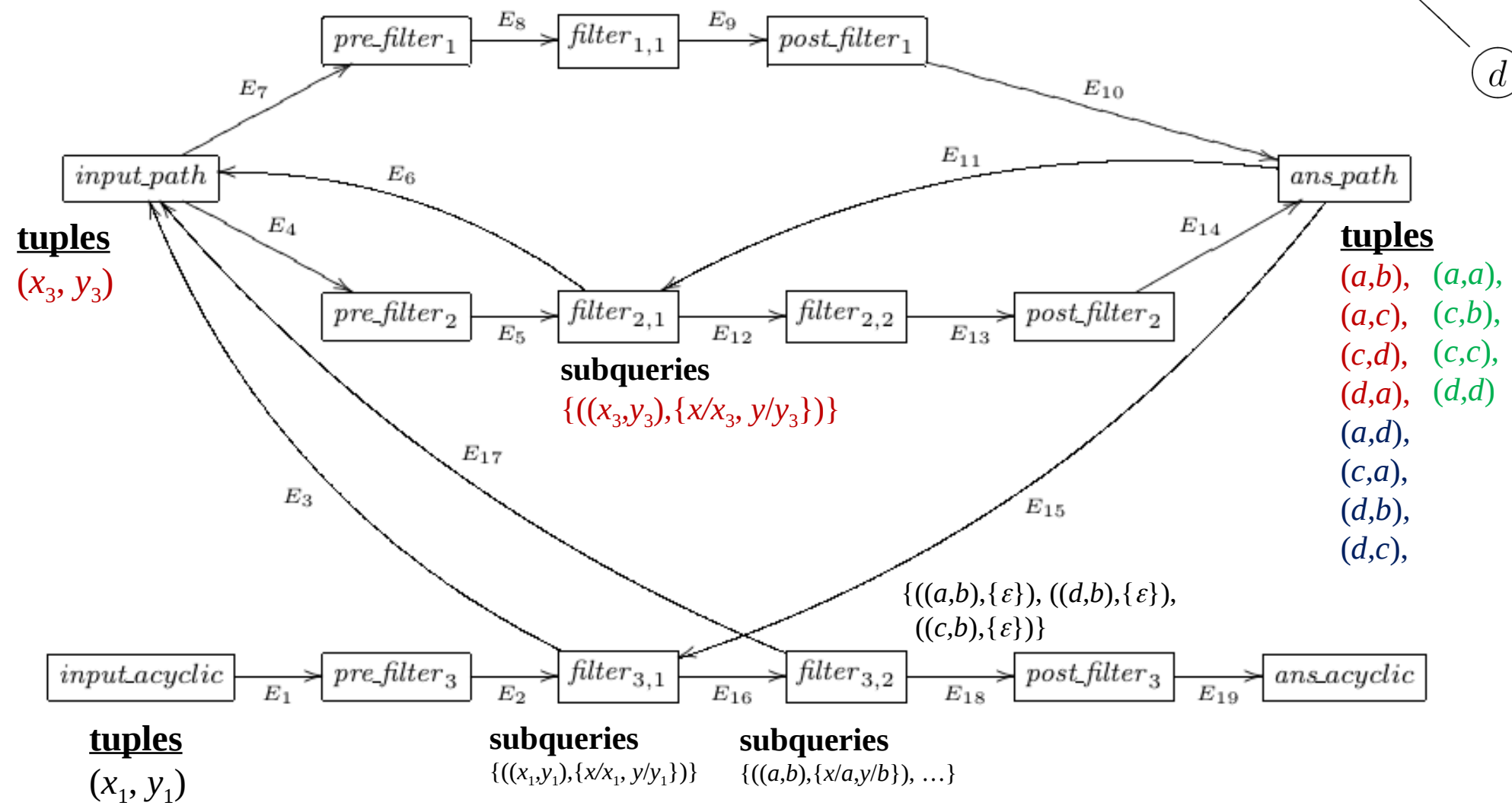
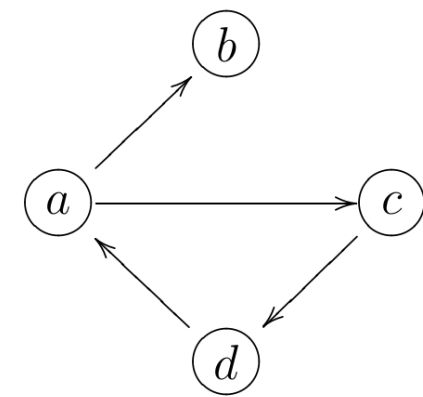
no tuple is added to $tuples(input_path)$ because the tuple (x_3, y_3) is more general than any other tuples



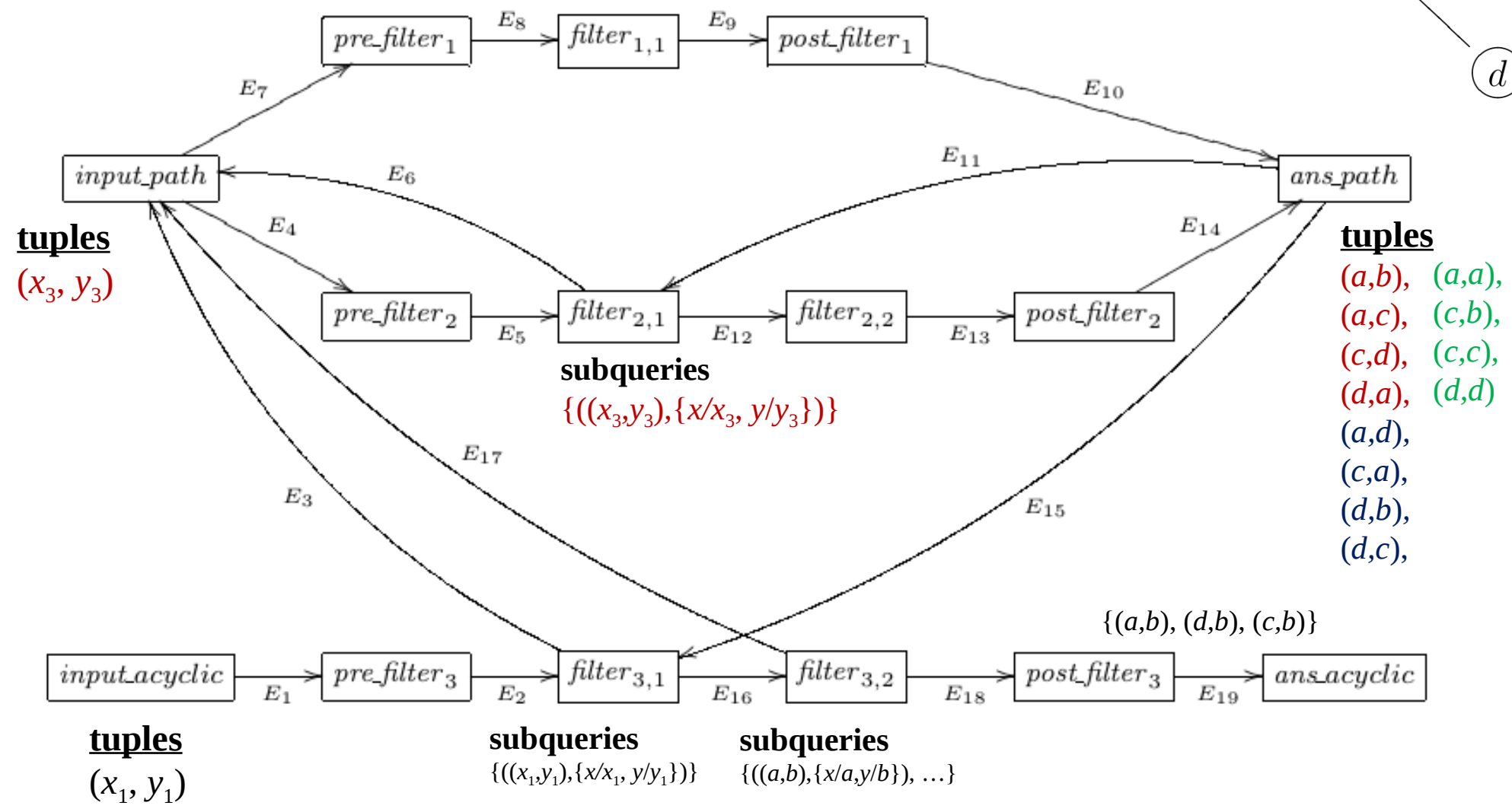
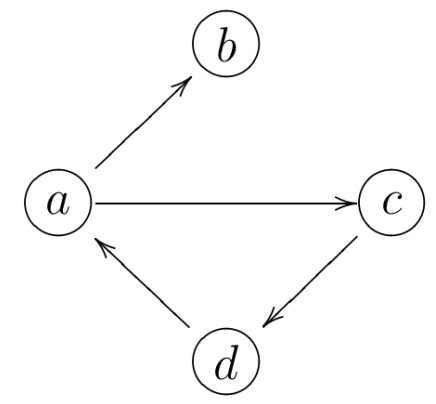
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



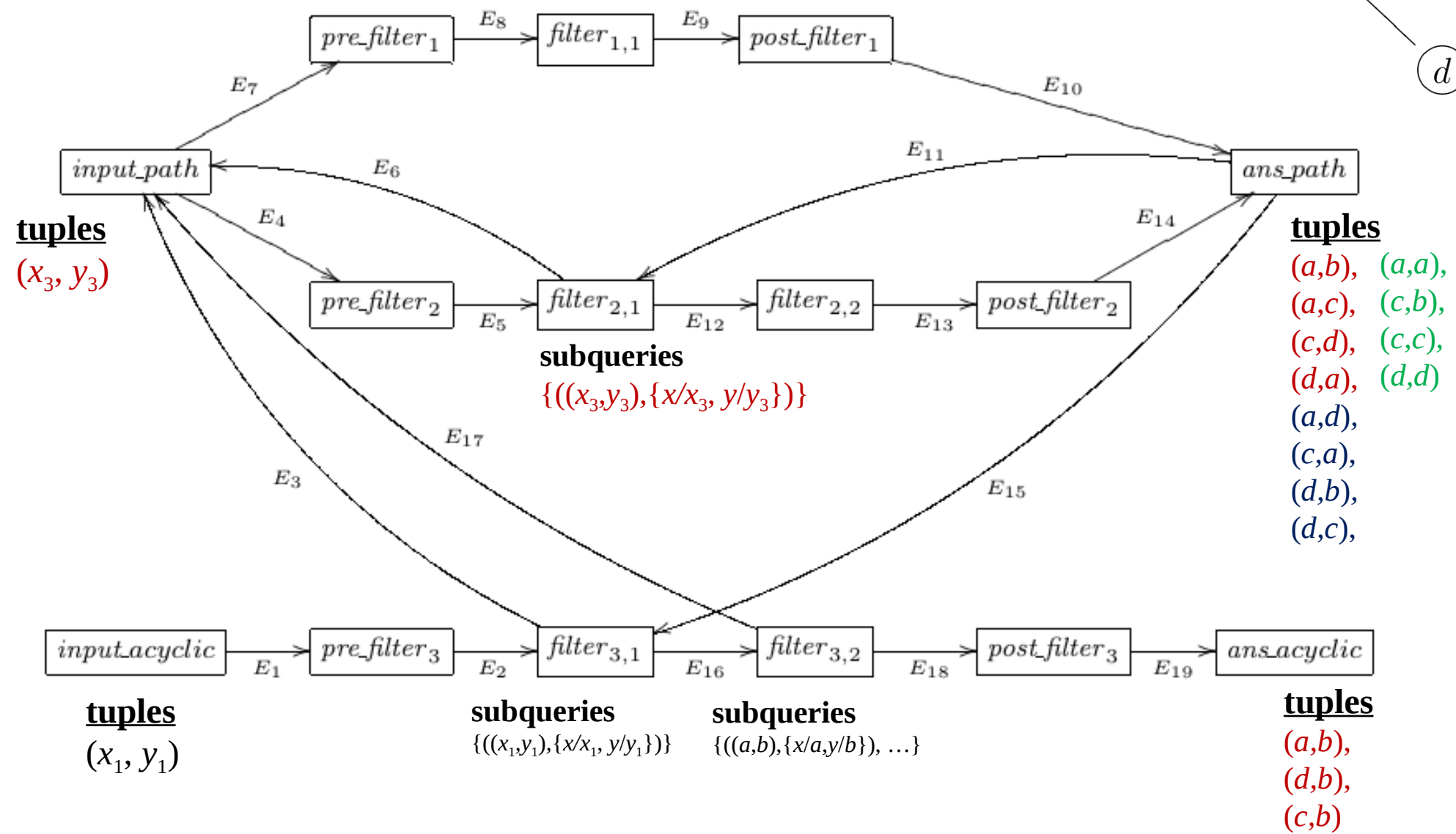
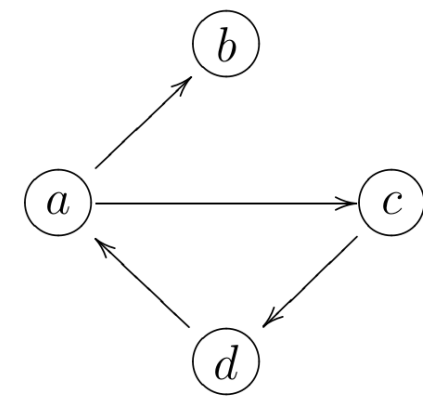
$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



$path(x, y) \leftarrow edge(x, y)$
 $path(x, y) \leftarrow path(x, z), edge(z, y)$
 $acyclic(x, y) \leftarrow path(x, y), \sim path(y, x).$



At this point, no edge in the net is active.
The algorithm terminates and returns the set
 $tuples(ans_acyclic) = \{(a,b), (d,b), (c,b)\}$.