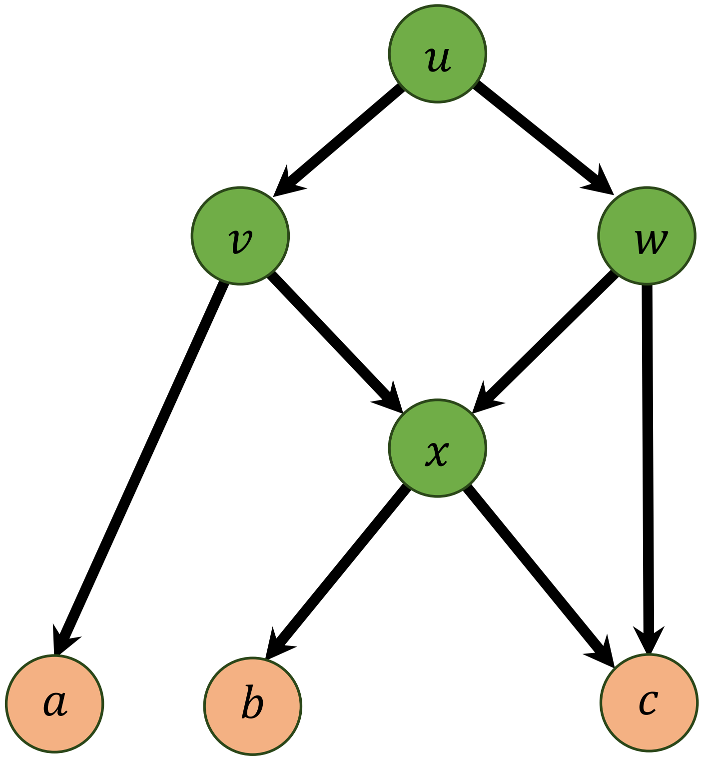
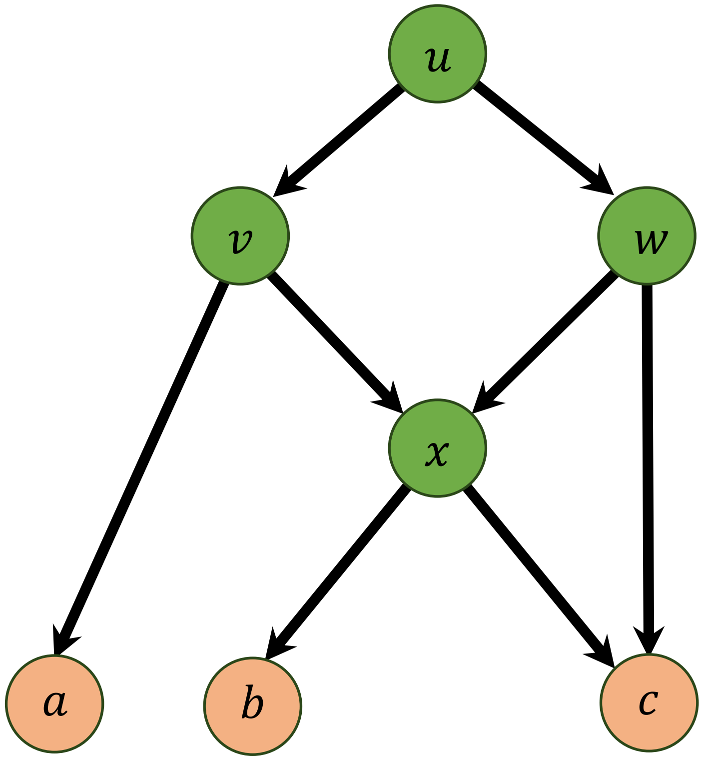
Descendant Clusters

Bruno J. Schmidt, Marc Hellmuth, Peter F. Stadler

Leaf & Descendant Clusters

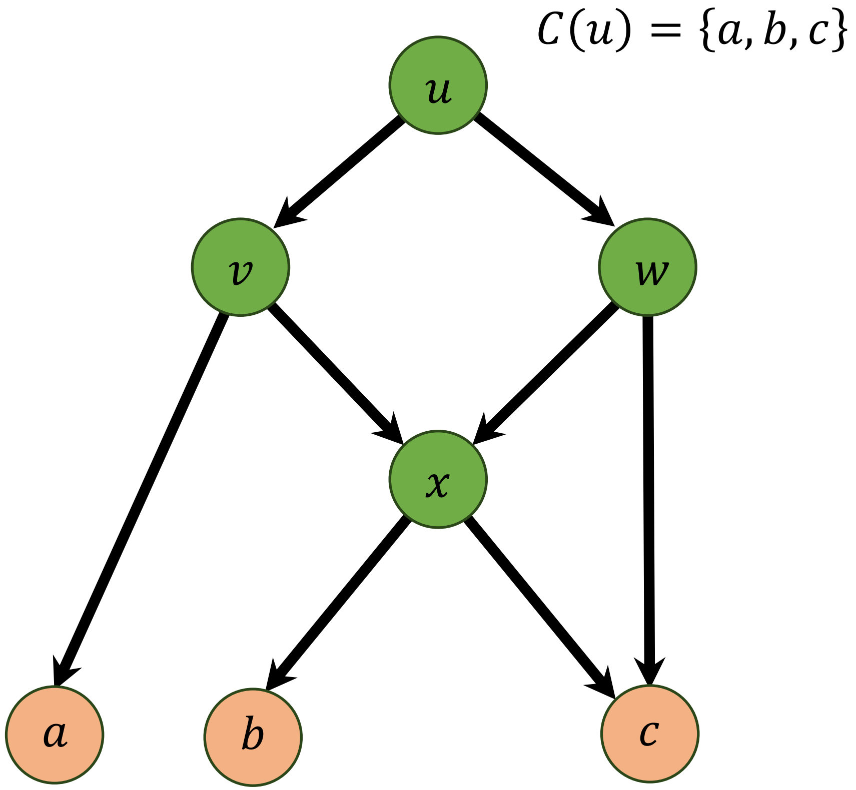


Leaf & Descendant Clusters



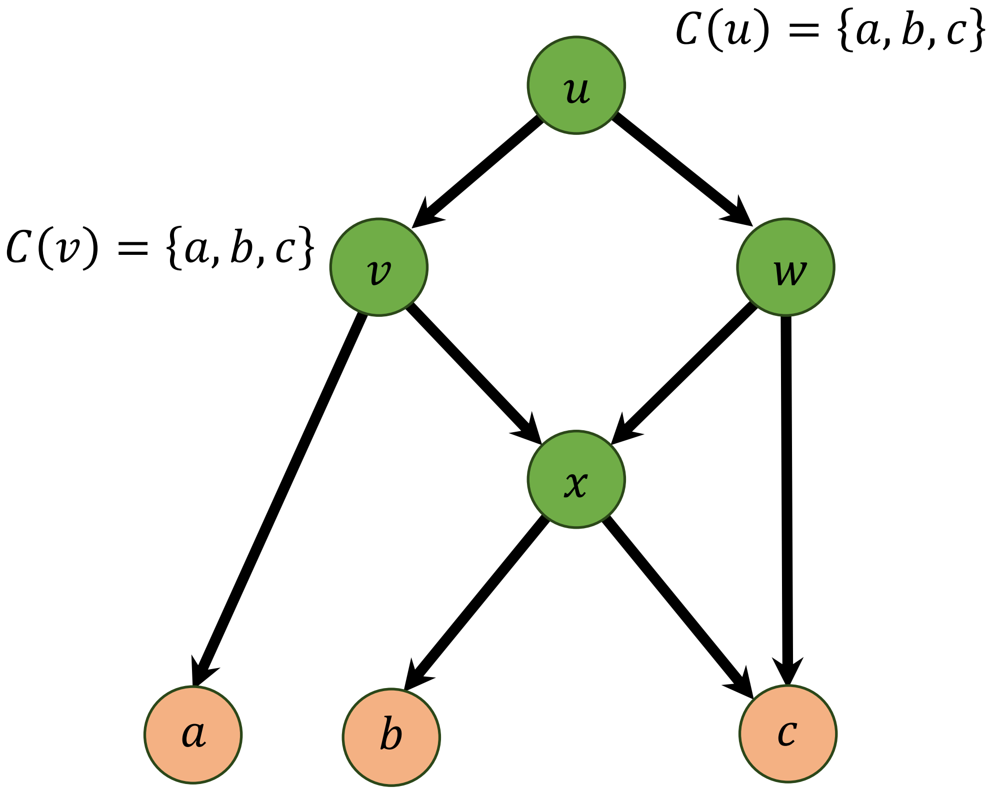
- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$

Leaf & Descendant Clusters



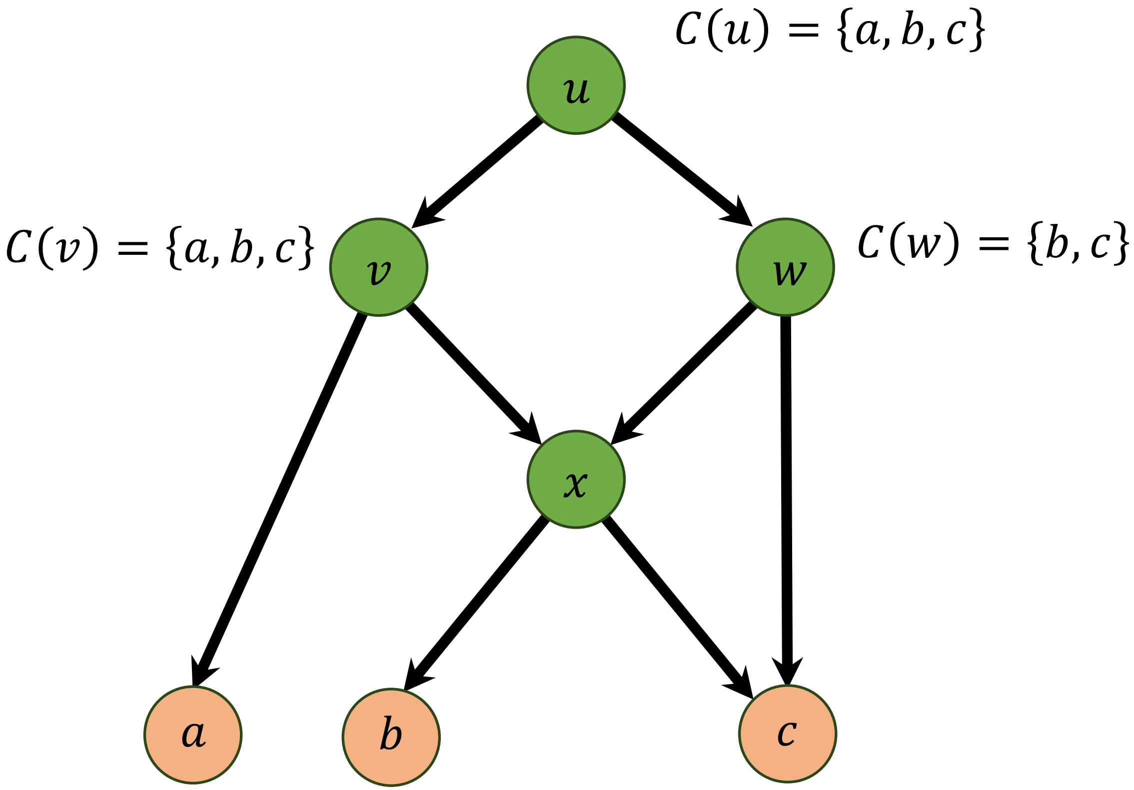
- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$

Leaf & Descendant Clusters



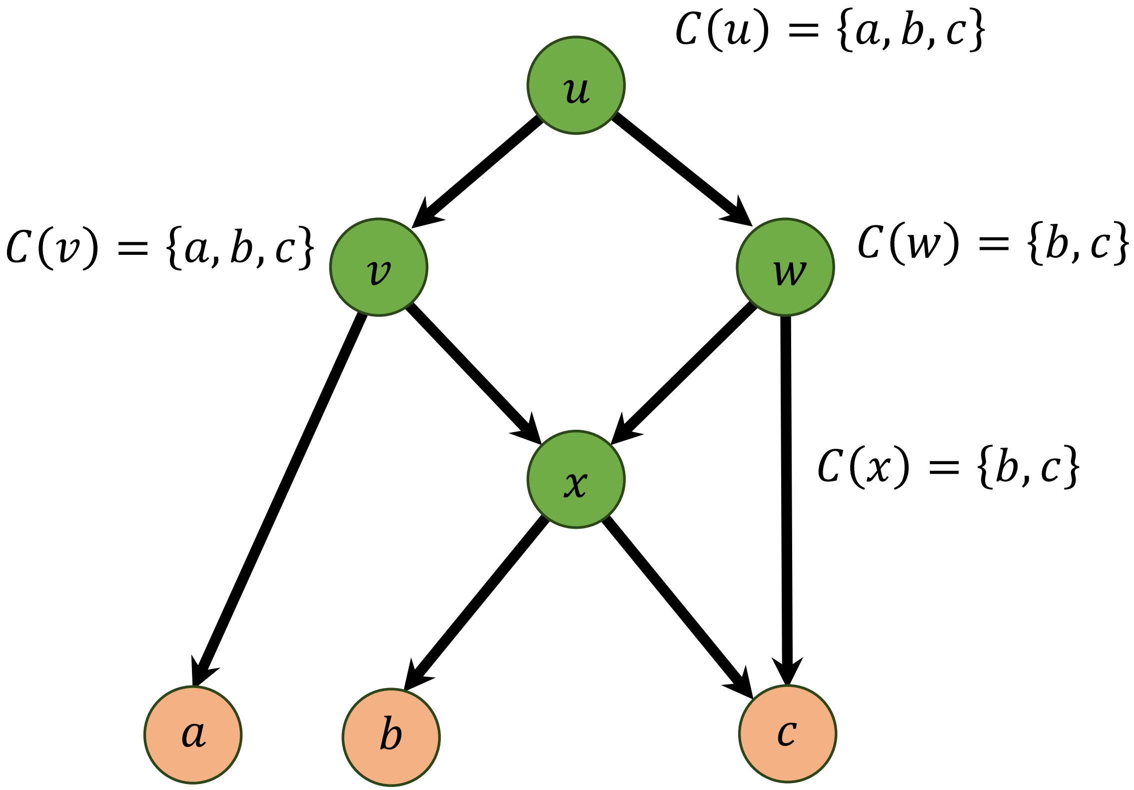
- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$

Leaf & Descendant Clusters



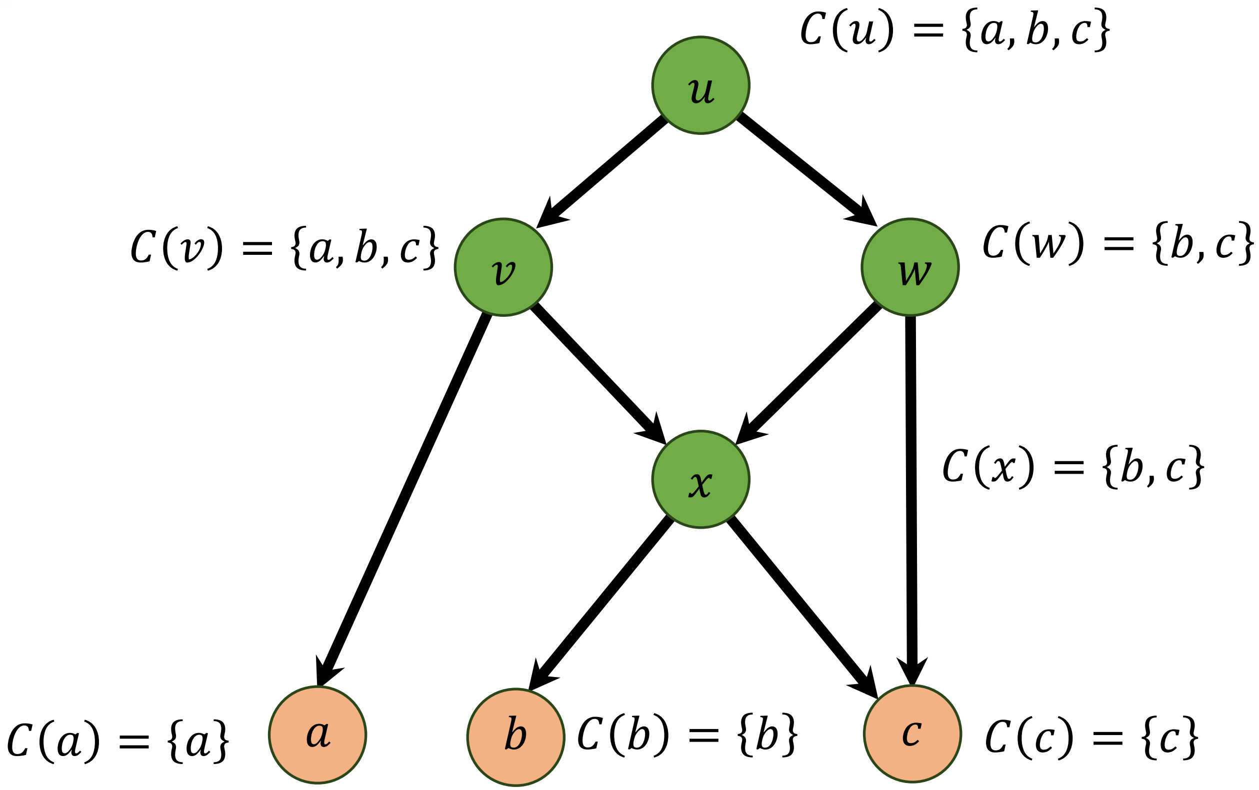
- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$

Leaf & Descendant Clusters



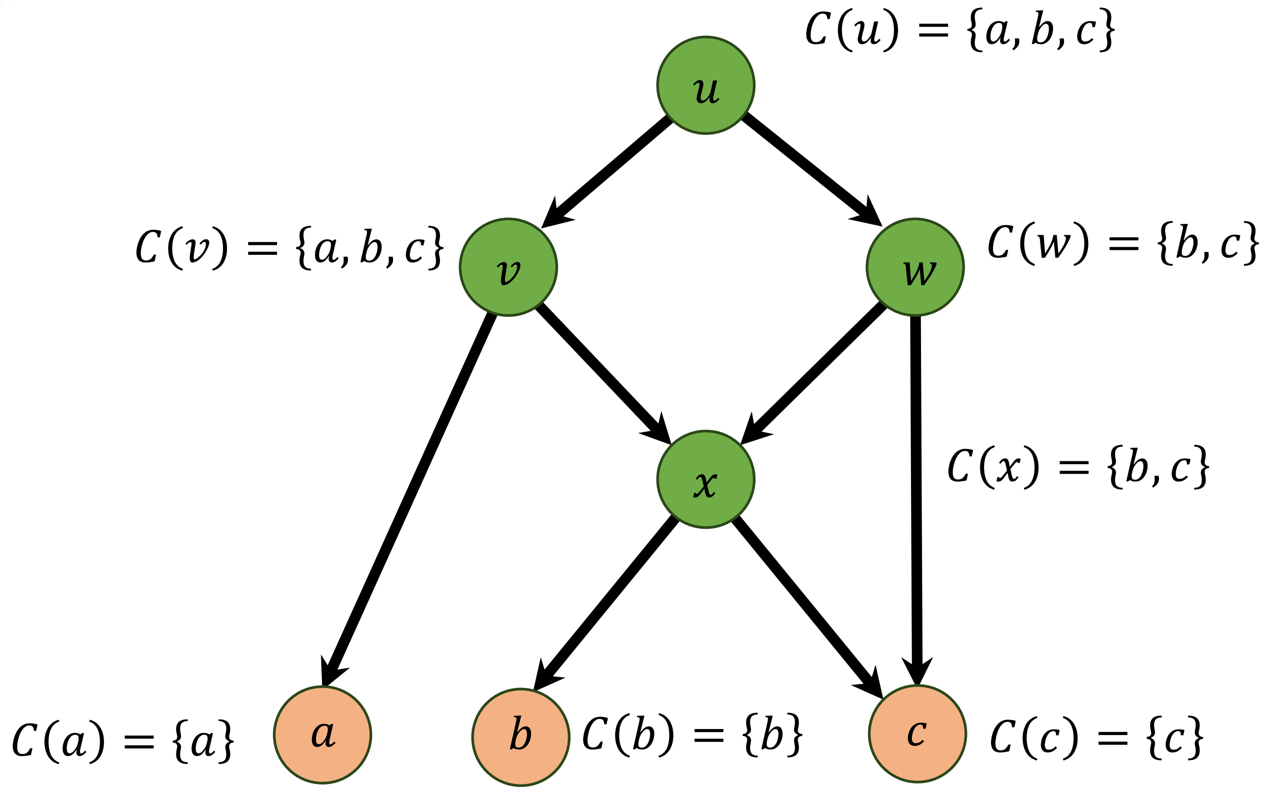
- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$

Leaf & Descendant Clusters



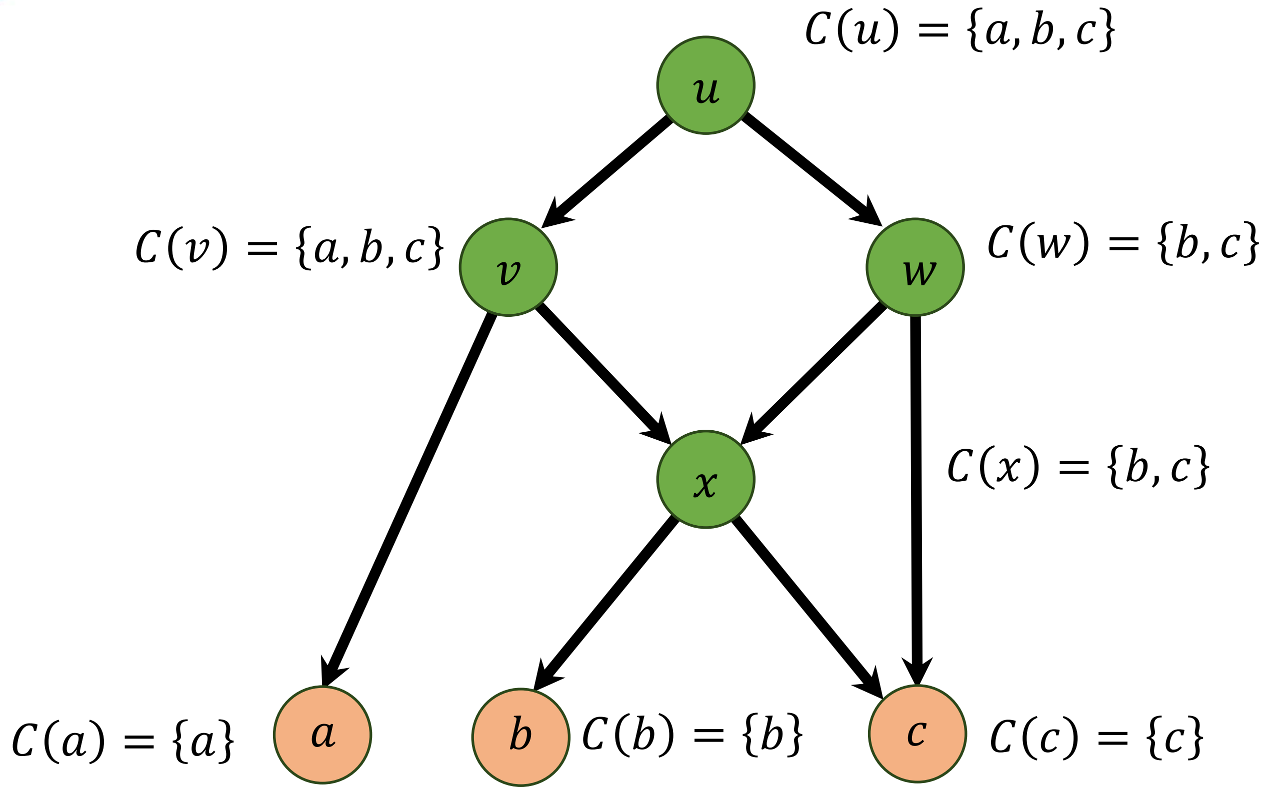
- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$

Leaf & Descendant Clusters



- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$

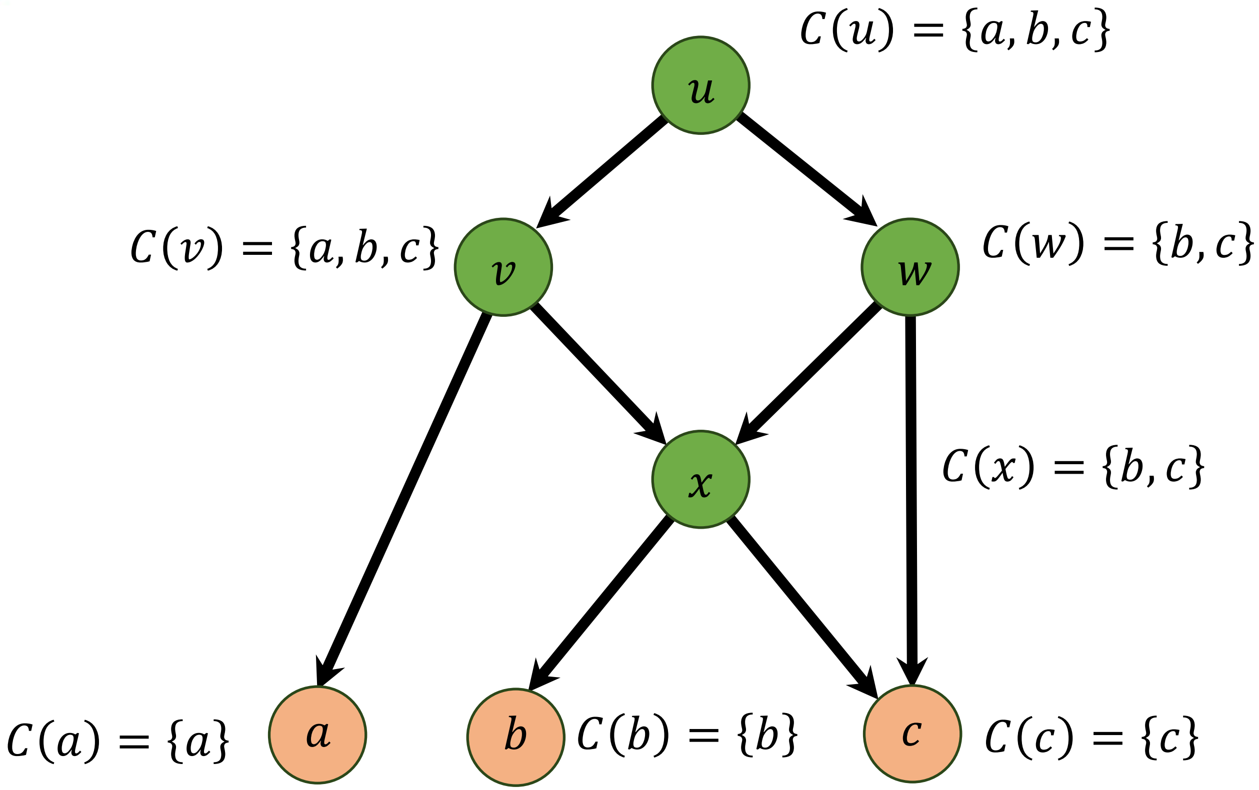
Leaf & Descendant Clusters



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$

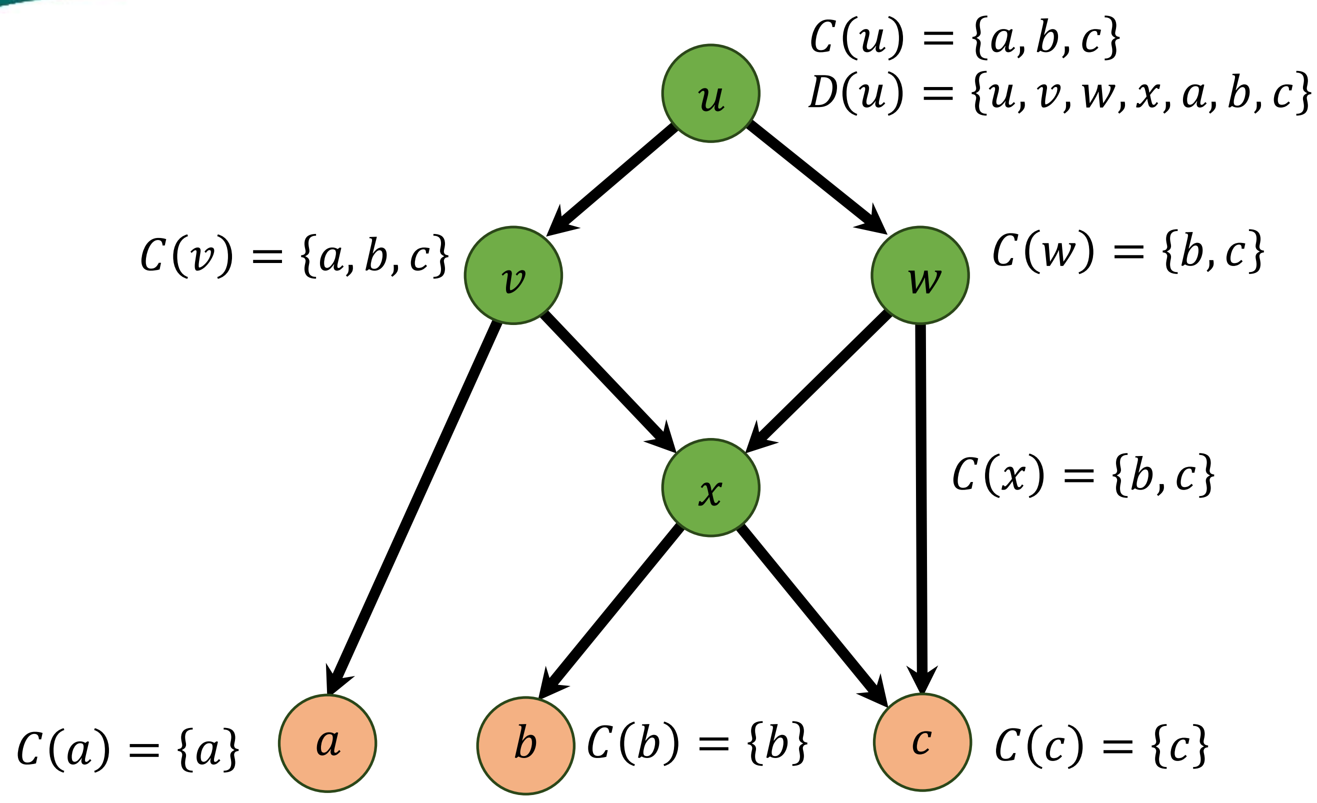
Leaf & Descendant Clusters



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$

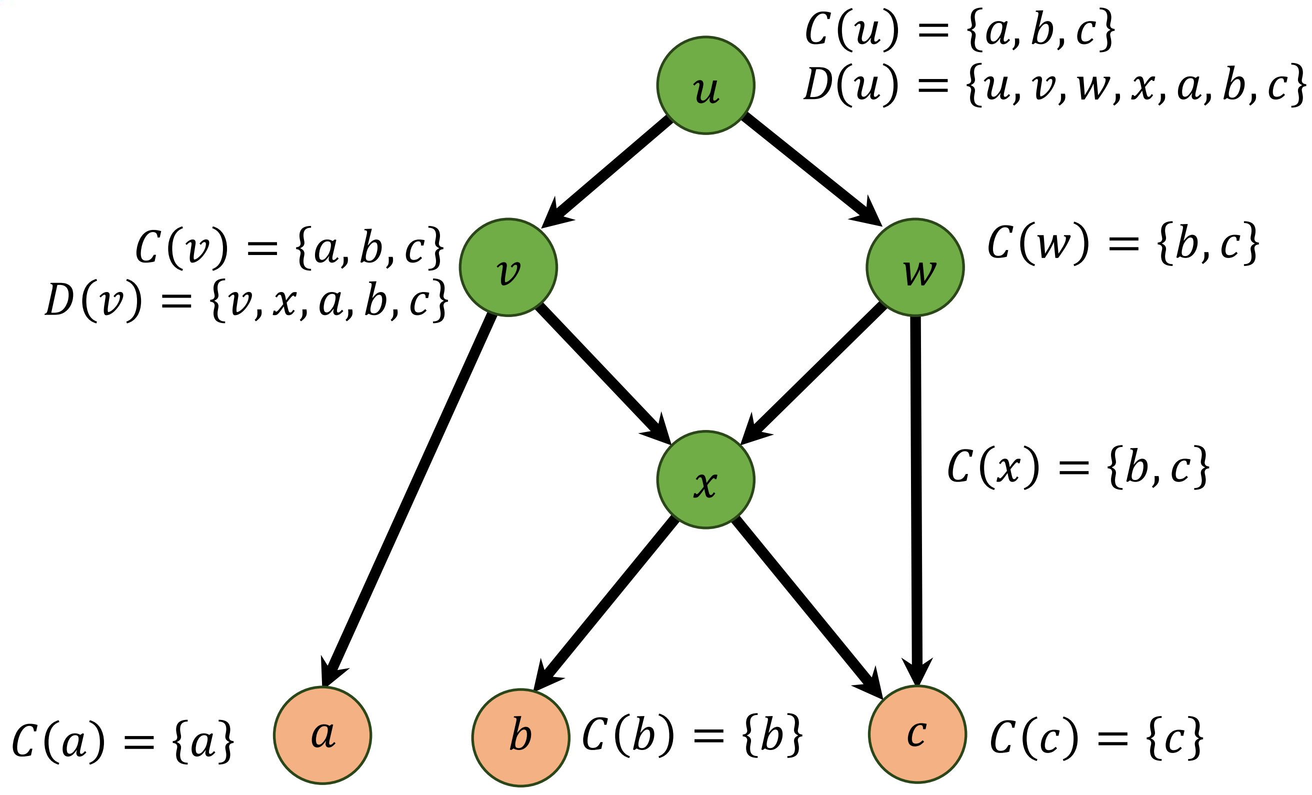
Leaf & Descendant Clusters



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$

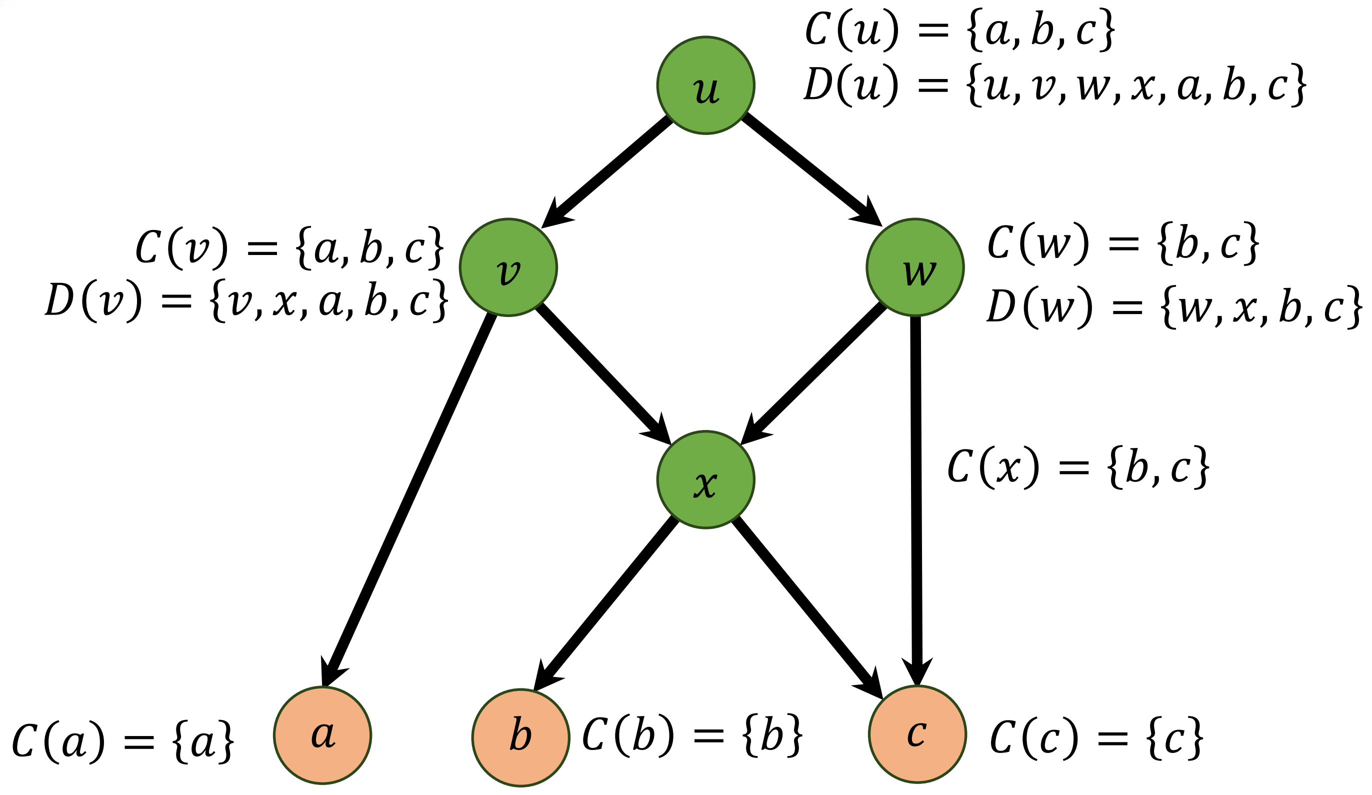
Leaf & Descendant Clusters



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$

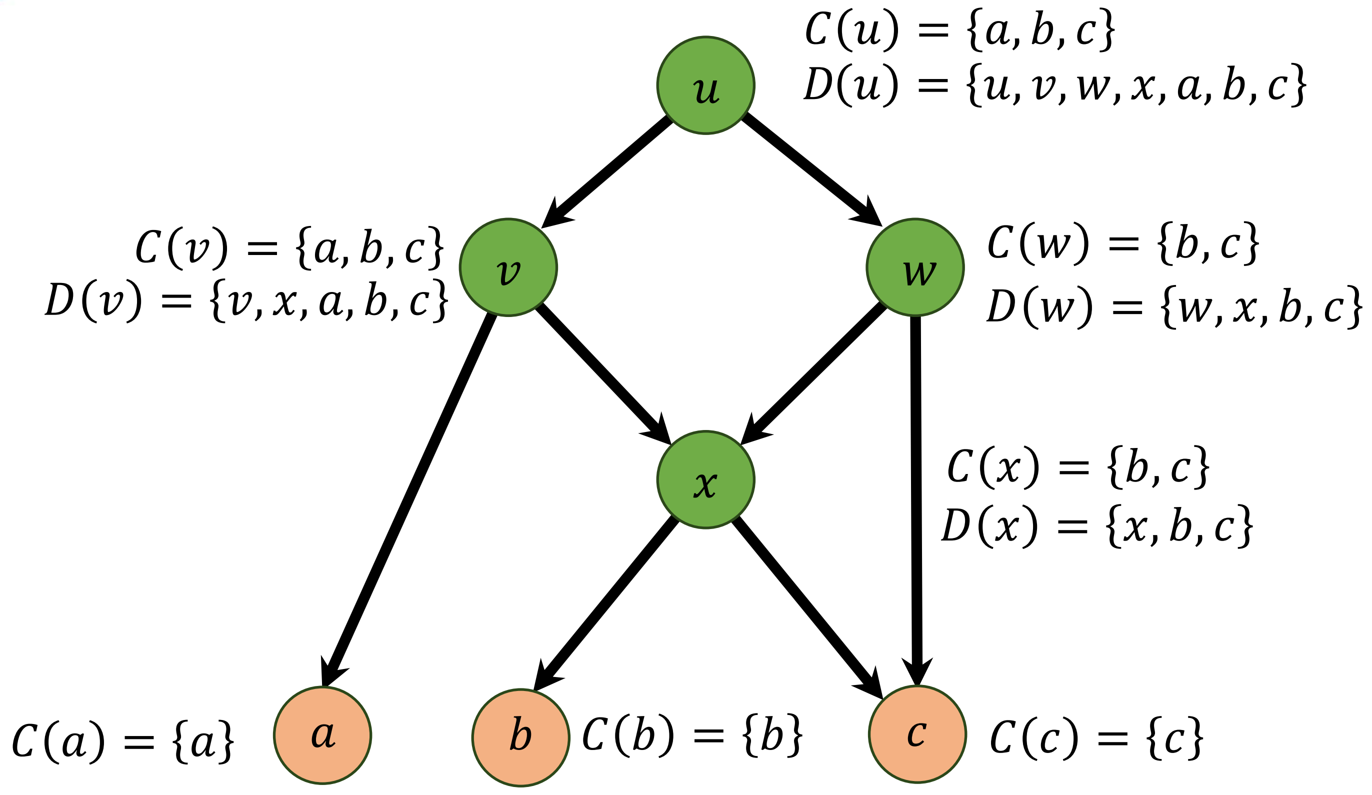
Leaf & Descendant Clusters



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$

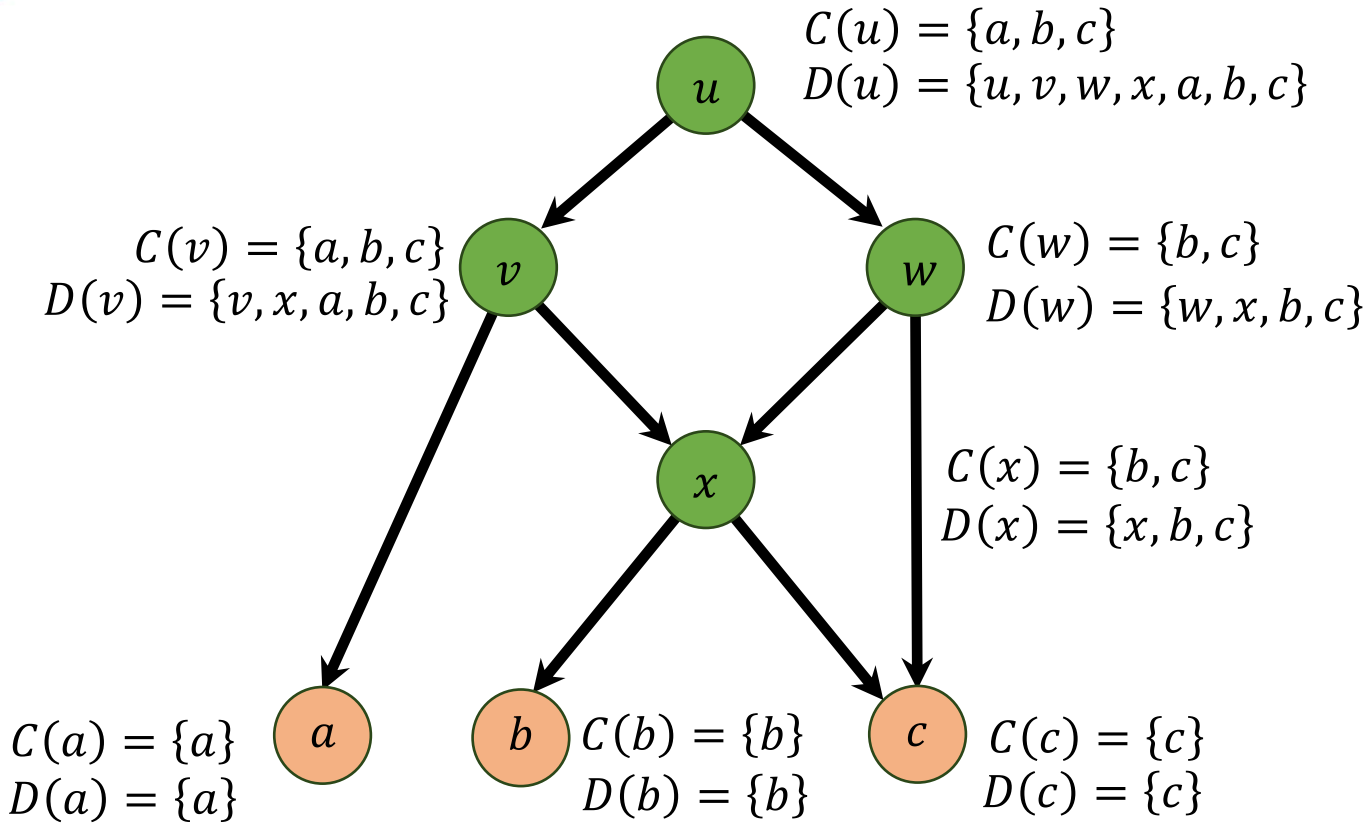
Leaf & Descendant Clusters



$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$

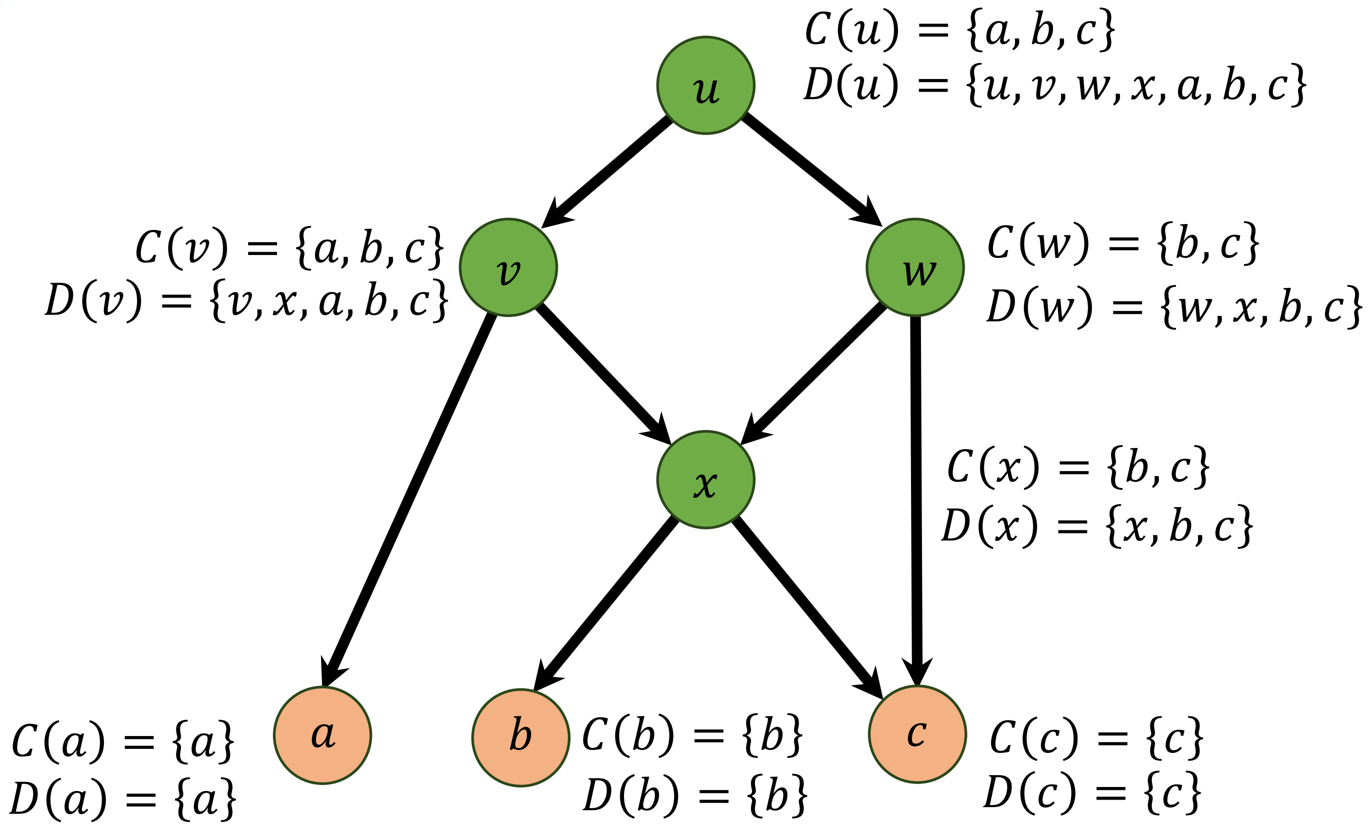
Leaf & Descendant Clusters



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$

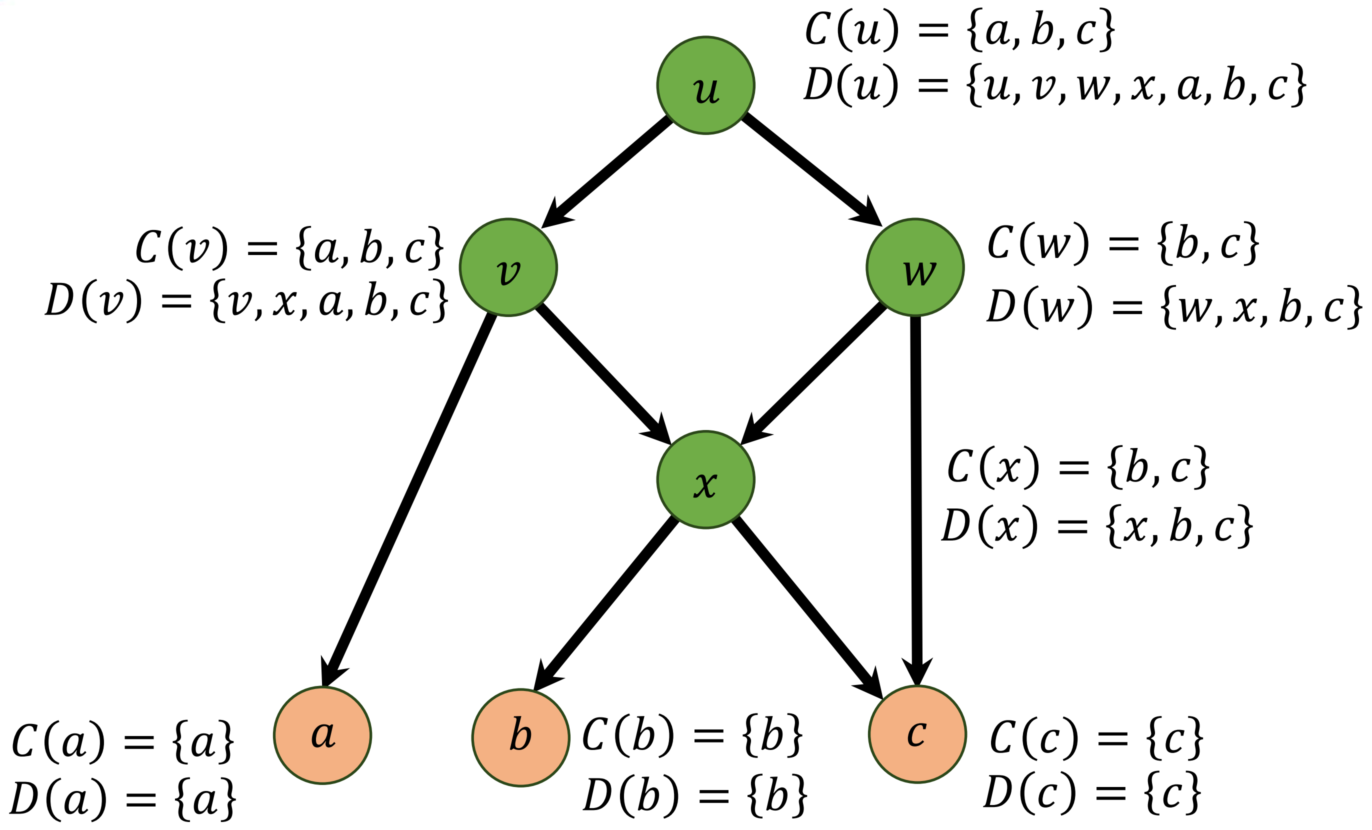
Leaf & Descendant Clusters



$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$
- **Descendant Cluster** (of a DAG)
  → Set of all descendant sets
  → $\mathfrak{D} = \{D(u) \mid u \in V(G)\}$

Leaf & Descendant Clusters



$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

- **Leaf Set (of a vertex u):**
  → All **leaves** reachable from u
  → denoted by $C(u)$
- **Leaf Cluster** (of a DAG)
  → Set of all leaf sets
  → $\mathfrak{C} = \{C(u) \mid u \in V(G)\}$
- **Descendant Set (of a vertex u)**
  → All **vertices** reachable from u
  → (all vertices v with $u \succcurlyeq v$)
  → denoted by $D(u)$
- **Descendant Cluster** (of a DAG)
  → Set of all descendant sets
  → $\mathfrak{D} = \{D(u) \mid u \in V(G)\}$

Hasse Diagram

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

Hasse Diagram

- “Diagram” or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $\mathrm{y} \subseteq x, \nexists\, w$ with $\mathrm{y} \subseteq w \subseteq x$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

- “Diagram” or Graph to **visualize partial orders**
→ edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
→ partial order is $(\mathfrak{S}, \subseteq)$
→ edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$\mathcal{H}(\mathfrak{S})$

$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$\mathcal{H}(\mathfrak{S})$

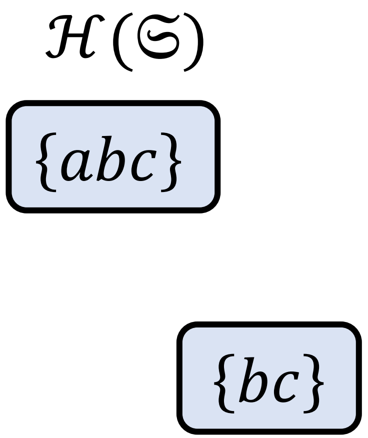
$\{abc\}$

$$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$



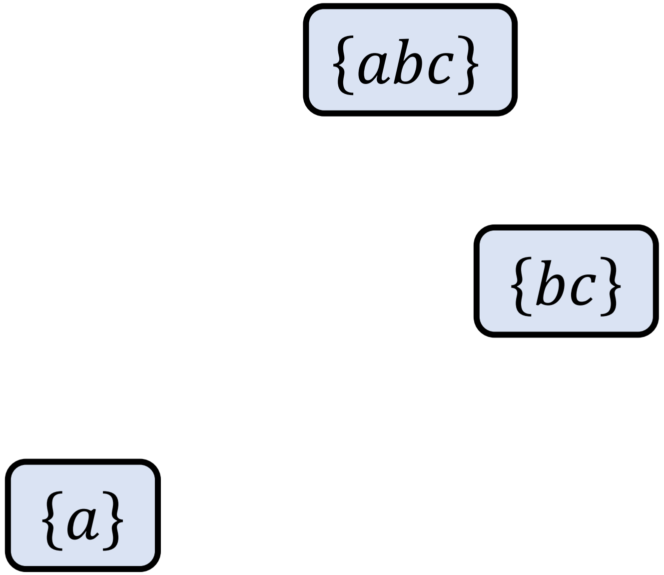
$$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$\mathcal{H}(\mathfrak{S})$

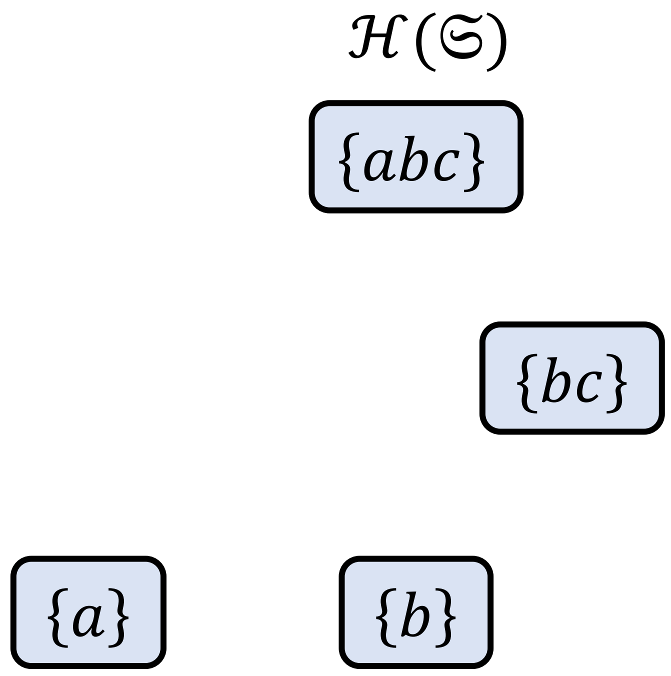


$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$



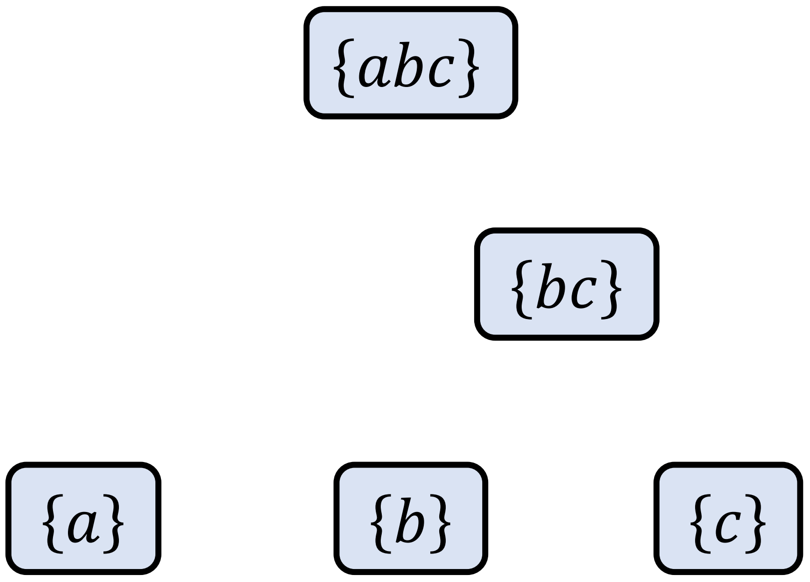
$$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$\mathcal{H}(\mathfrak{S})$



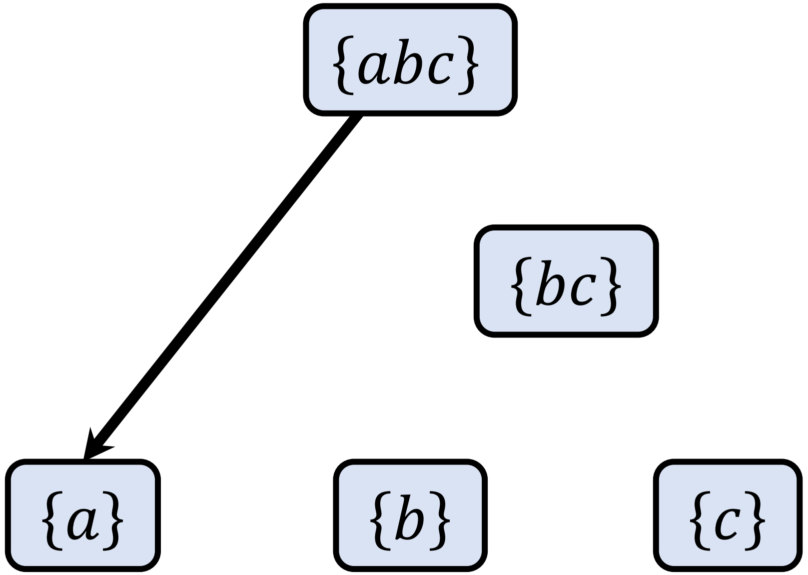
$$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \gneq y, \nexists\, w$ with $x \gneq w \gneq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$\mathcal{H}(\mathfrak{S})$

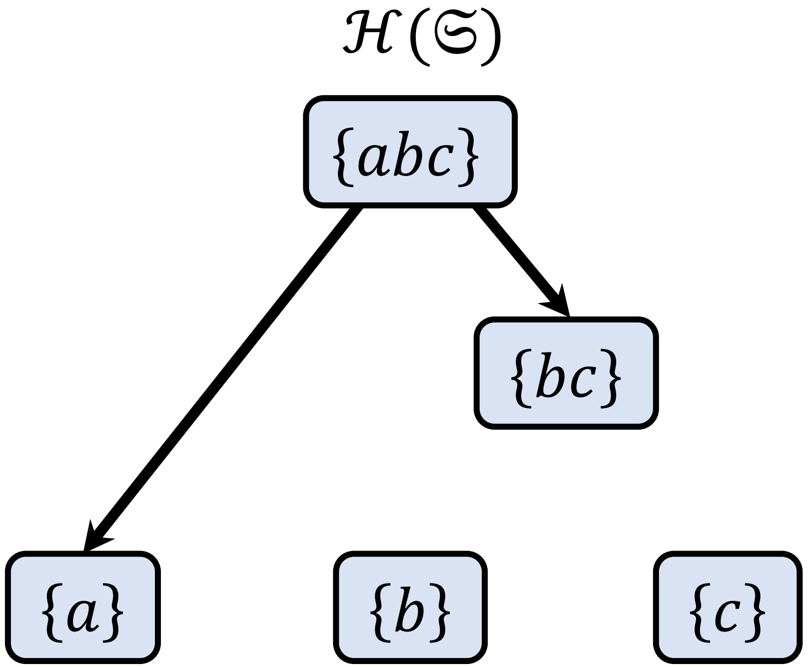


$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$



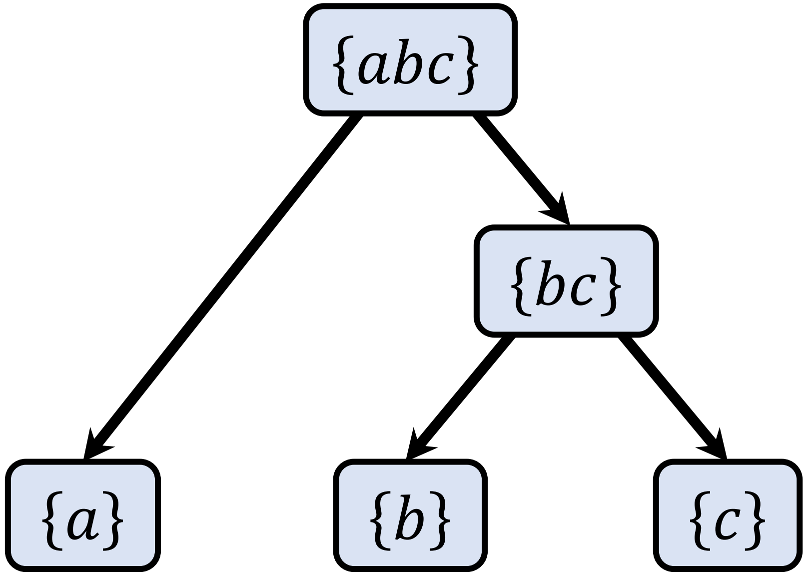
$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$

Hasse Diagram

- "Diagram" or Graph to **visualize partial orders**
  → edge (x, y) iff $x \succcurlyeq y, \nexists\, w$ with $x \succcurlyeq w \succcurlyeq y$

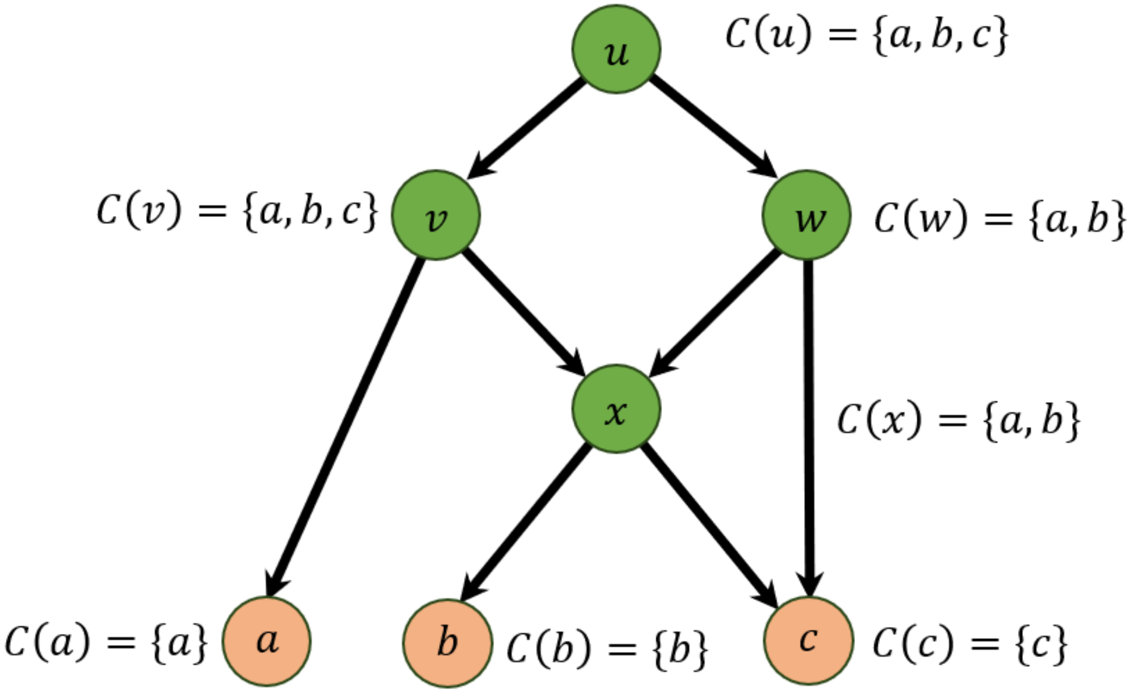
- Often utilized to **visualize set systems**
  → partial order is $(\mathfrak{S}, \subseteq)$
  → edge (x, y) iff $y \subseteq x, \nexists\, w$ with $y \subseteq w \subseteq x$

$\mathcal{H}(\mathfrak{S})$

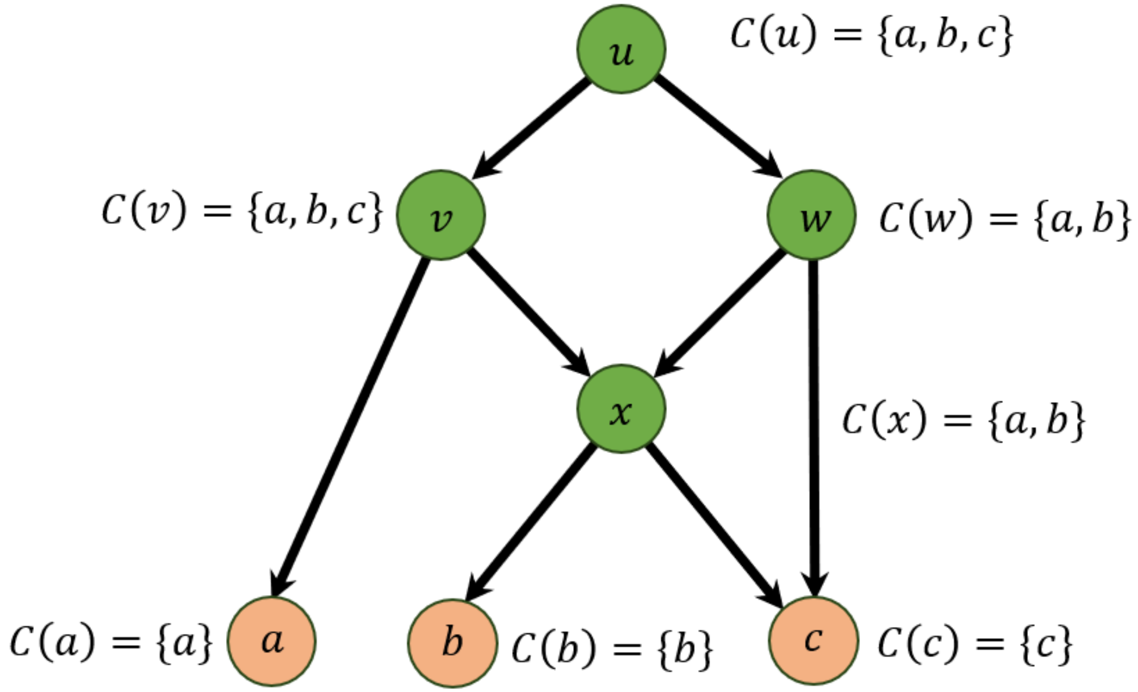


$$\mathfrak{S} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

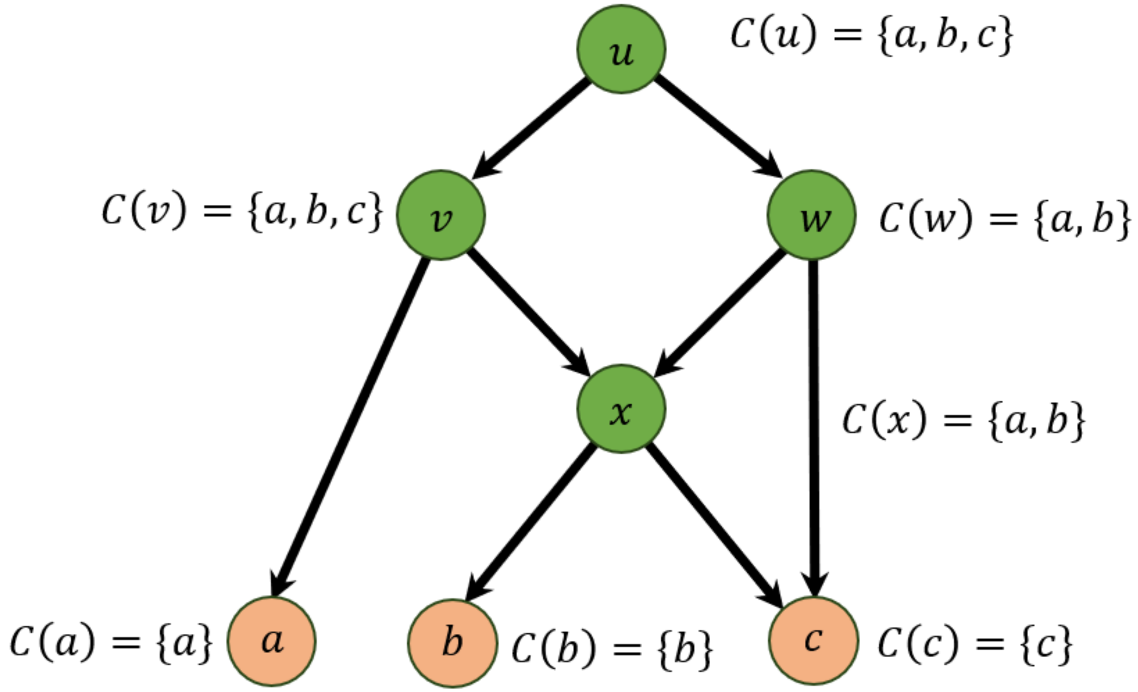


Hasse Diagram



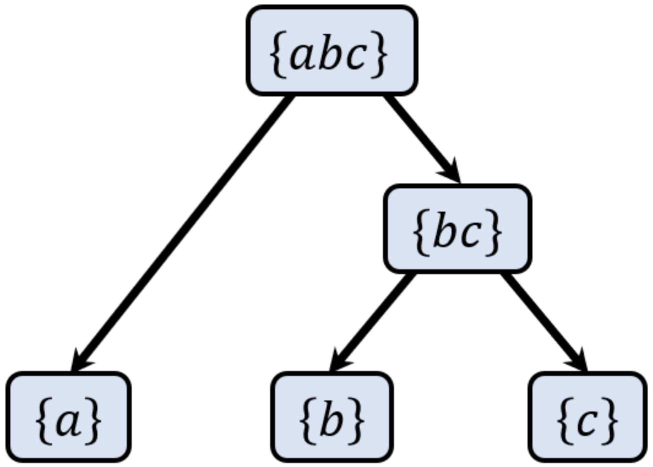
$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram

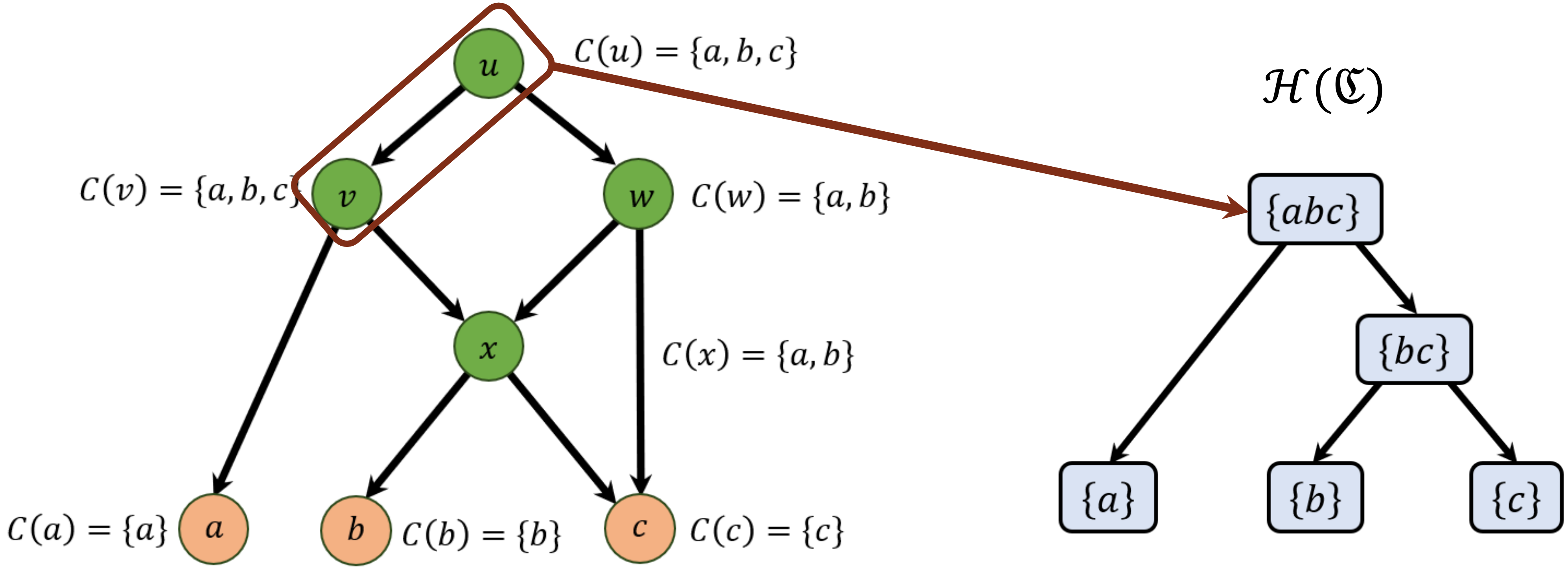


$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

$\mathcal{H}(\mathfrak{C})$

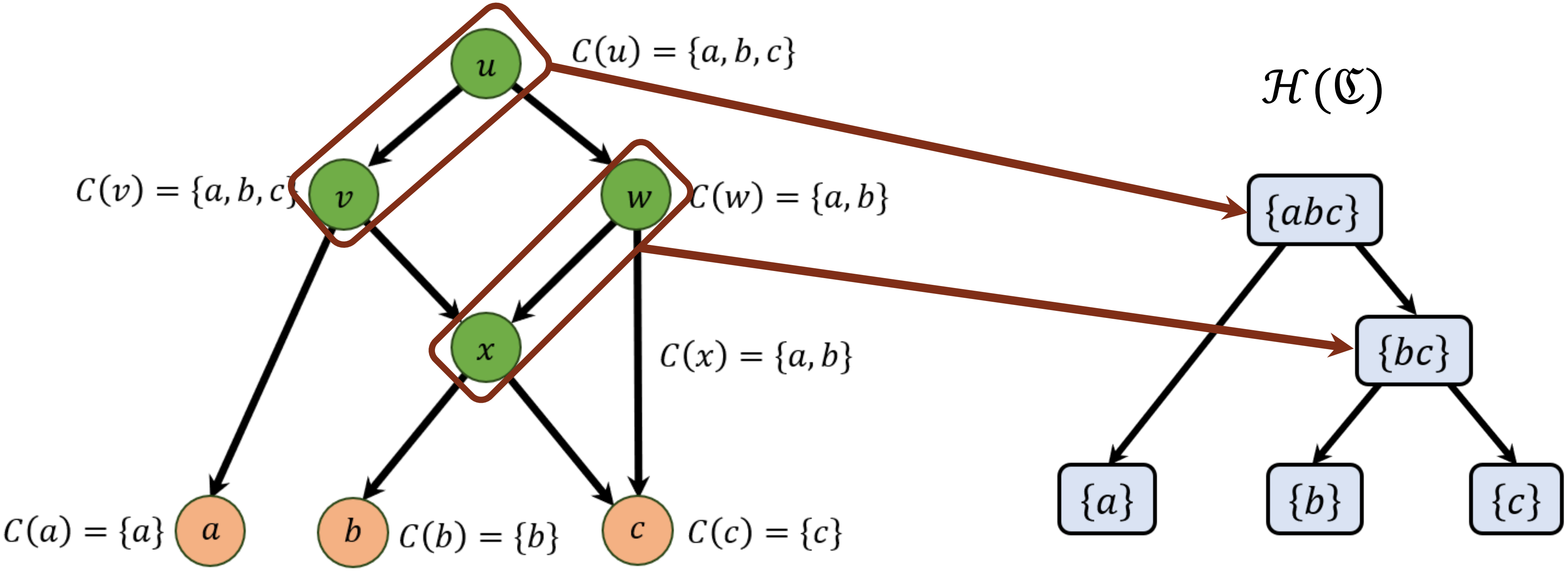


Hasse Diagram



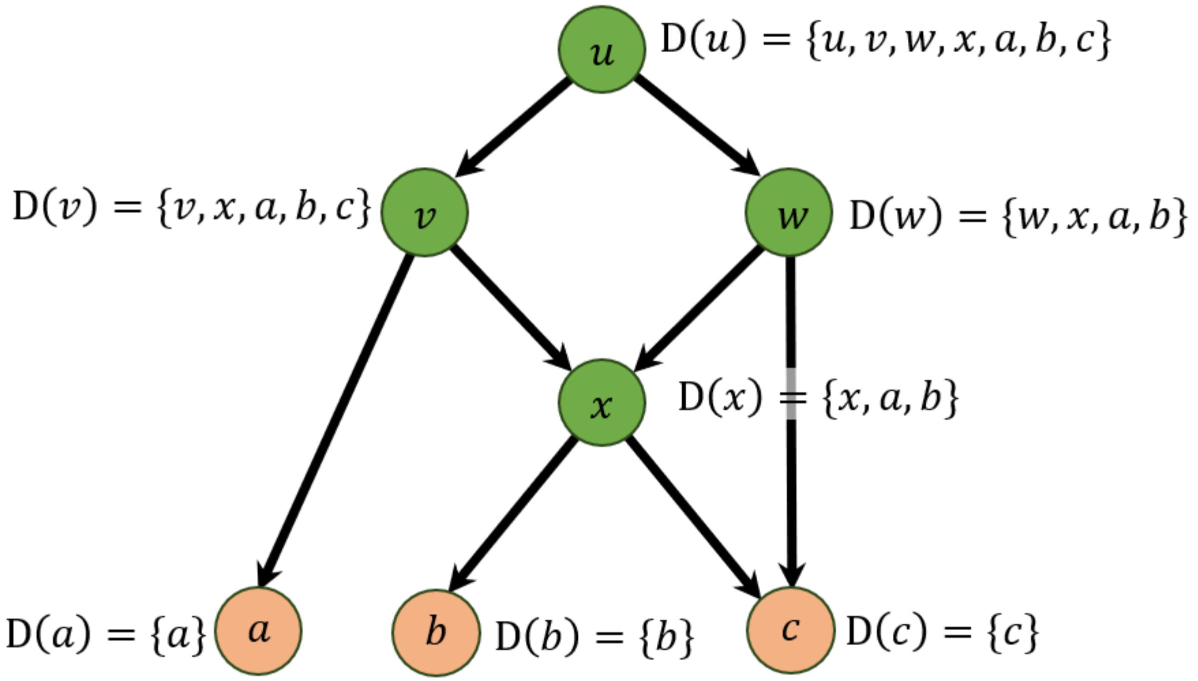
$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

Hasse Diagram



$$\mathfrak{C} = \{\{abc\}, \{bc\}, \{a\}, \{b\}, \{c\}\}$$

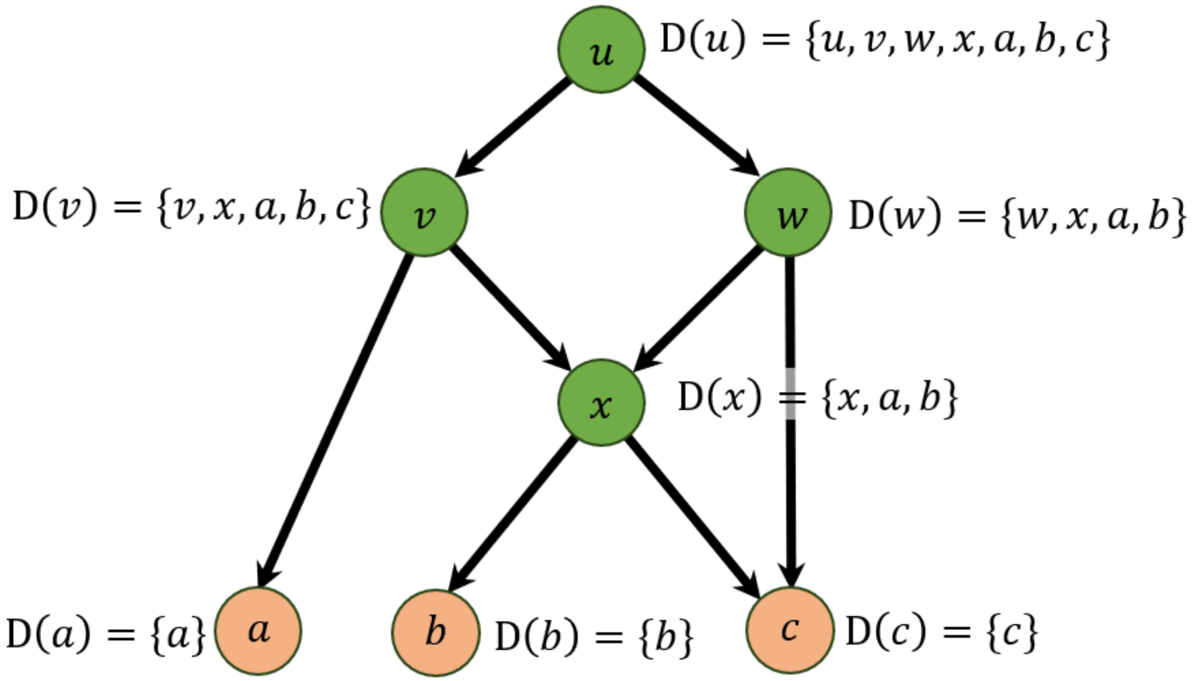
Hasse Diagram



$$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$$

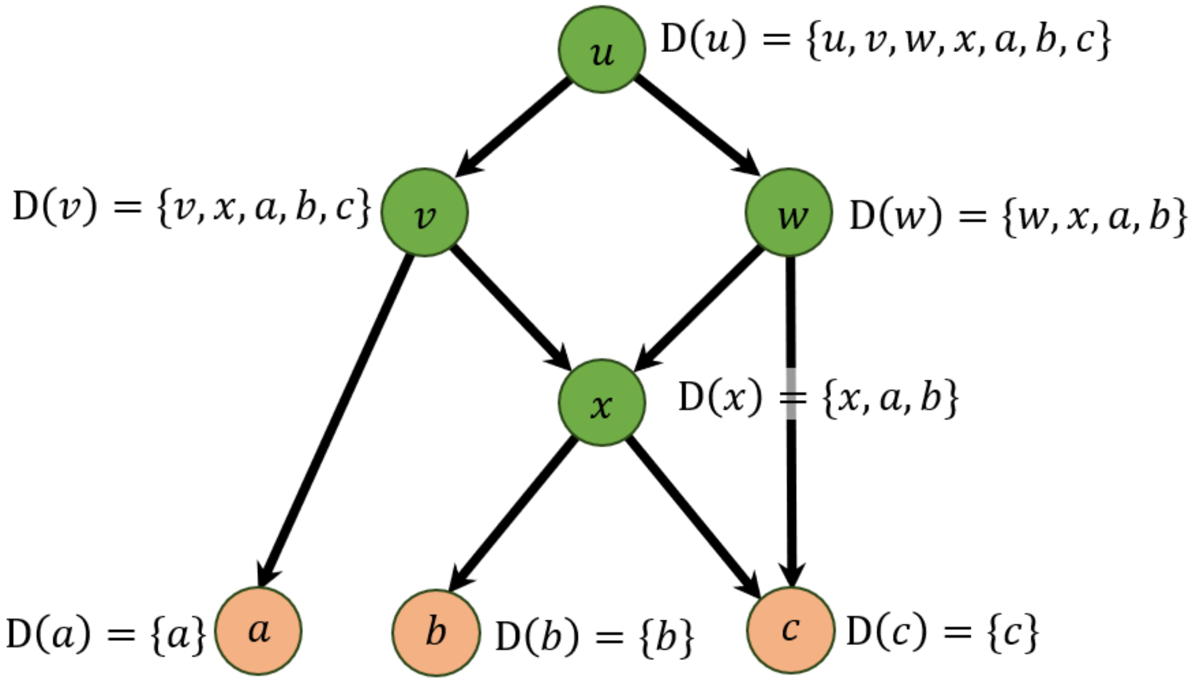
Hasse Diagram

$\mathcal{H}(\mathfrak{D})$



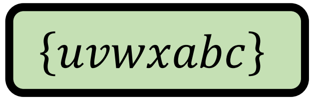
$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

Hasse Diagram

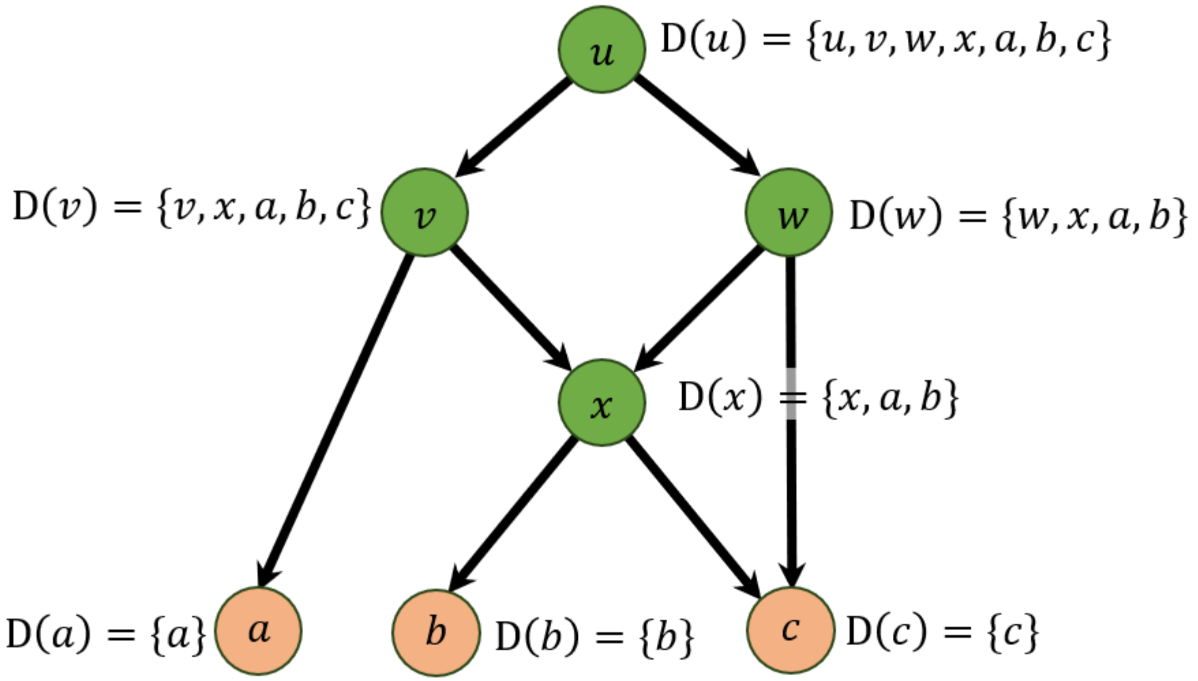


$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

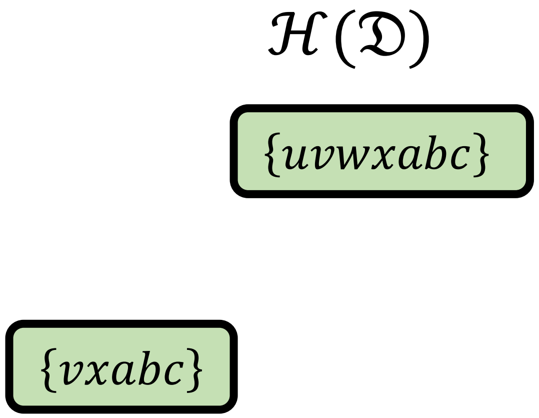
$\mathcal{H}(\mathfrak{D})$



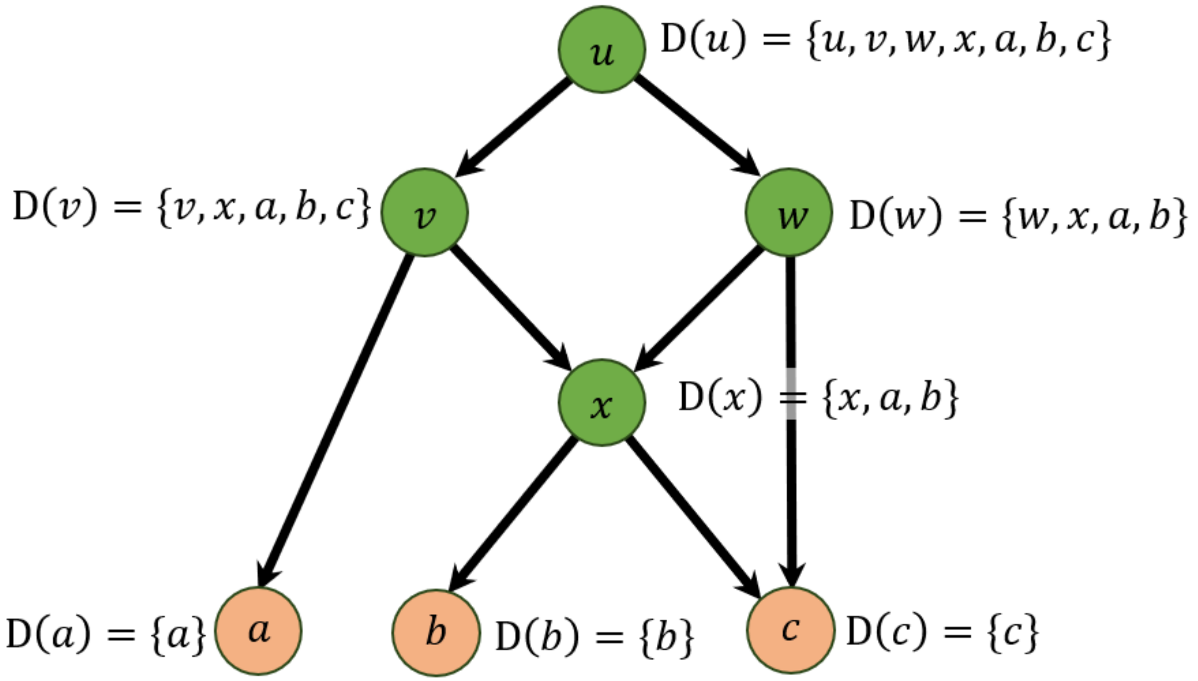
Hasse Diagram



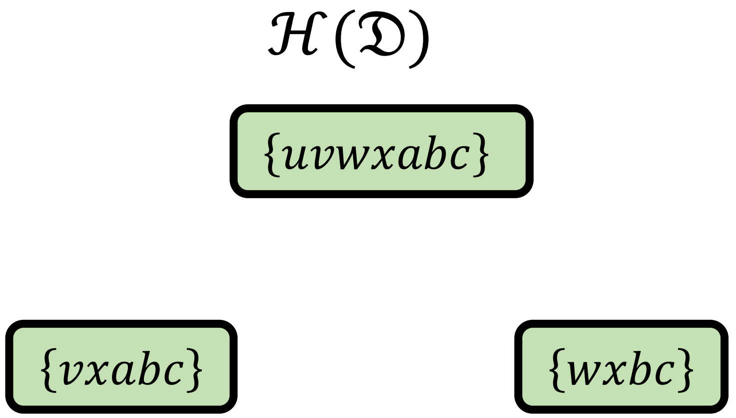
$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$



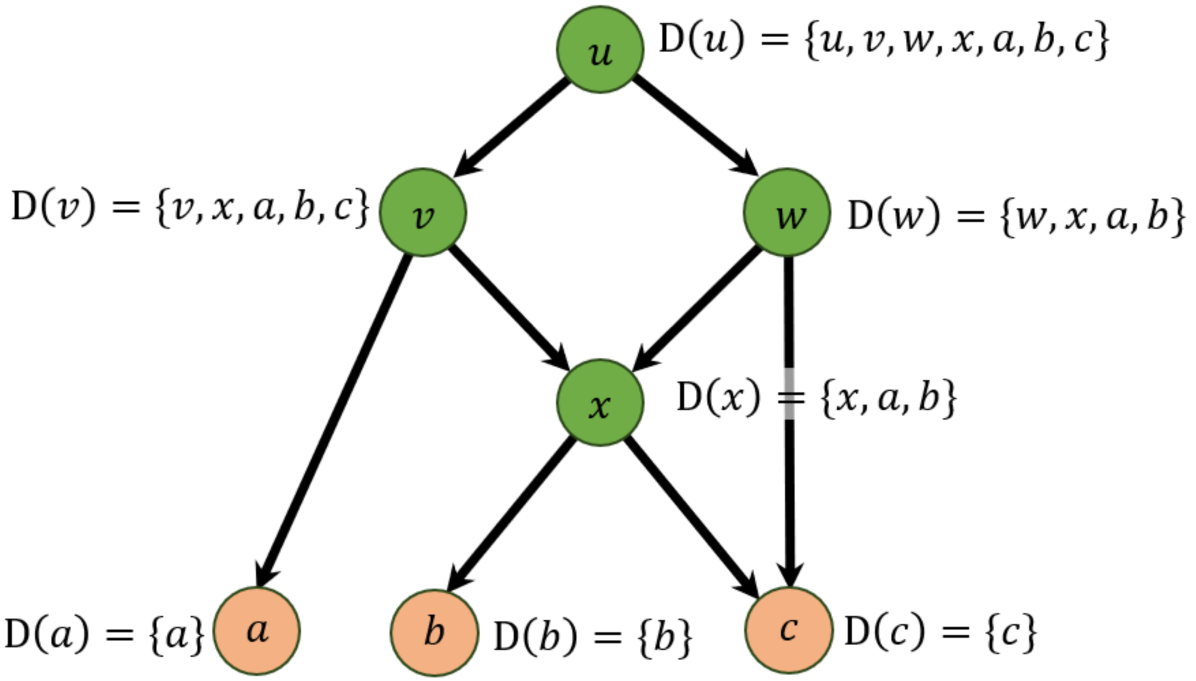
Hasse Diagram



$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

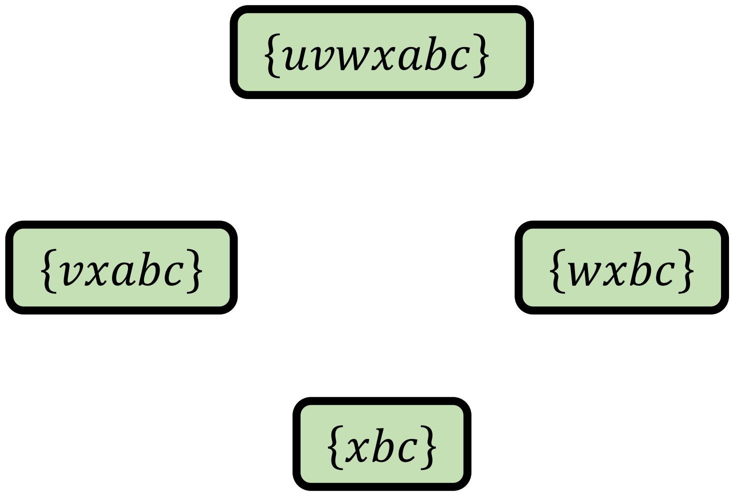


Hasse Diagram

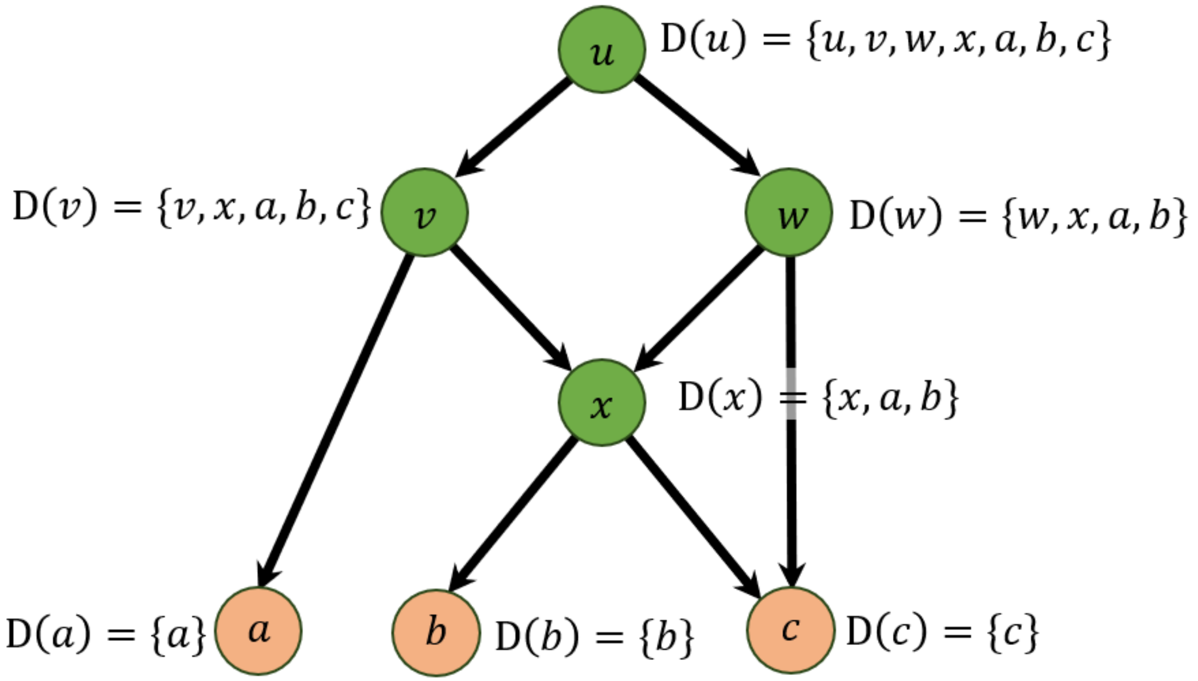


$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

$\mathcal{H}(\mathfrak{D})$

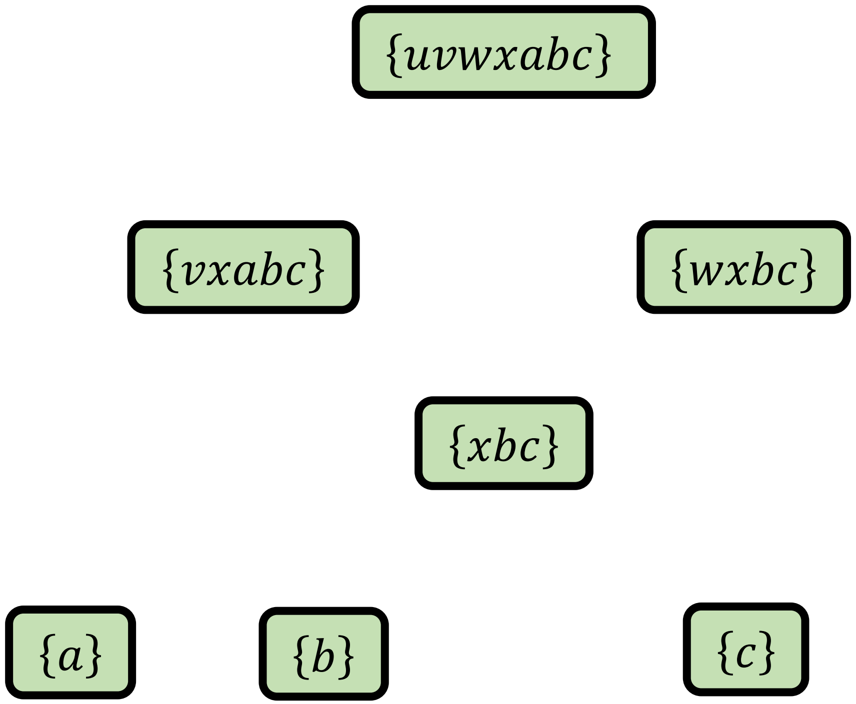


Hasse Diagram

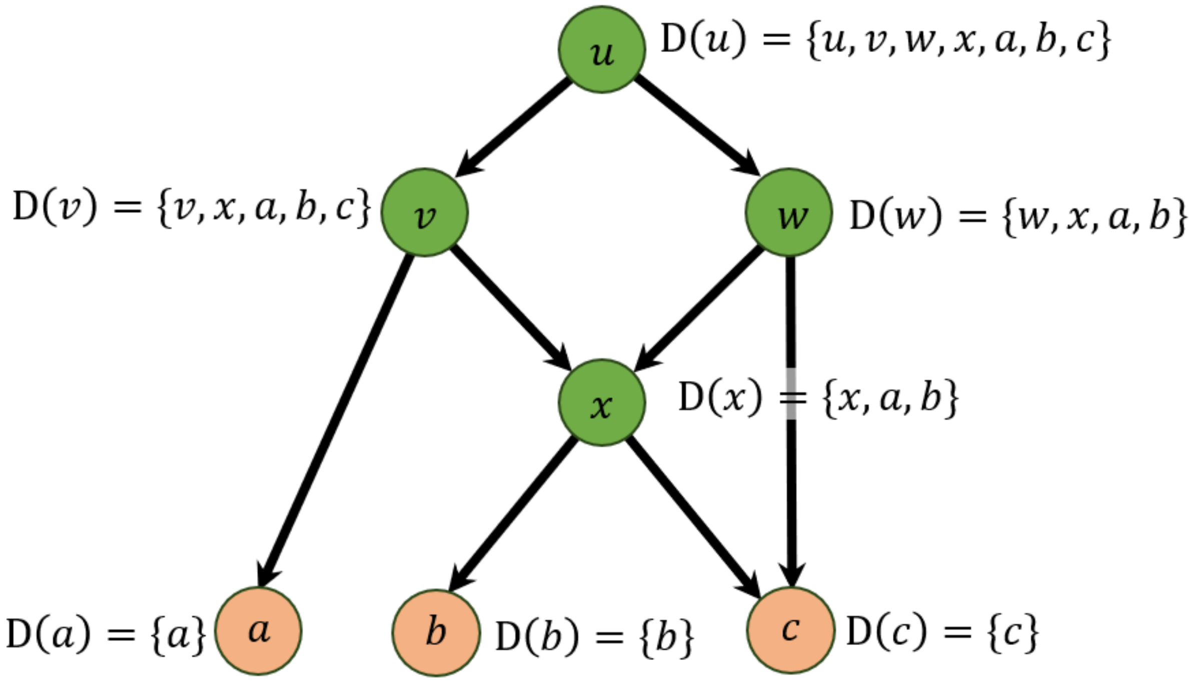


$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

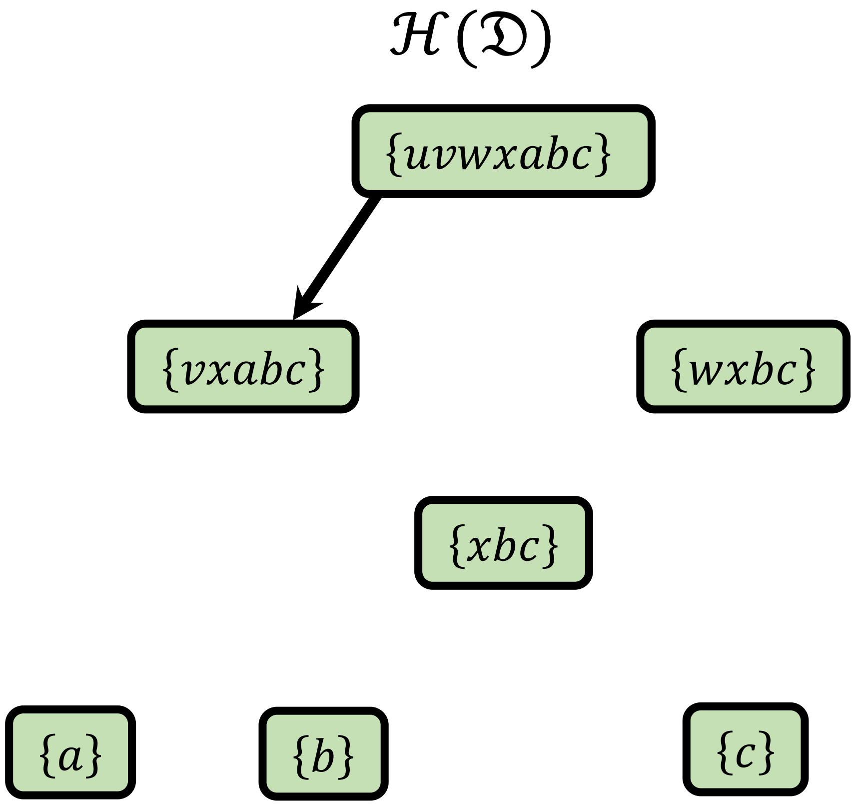
$\mathcal{H}(\mathfrak{D})$



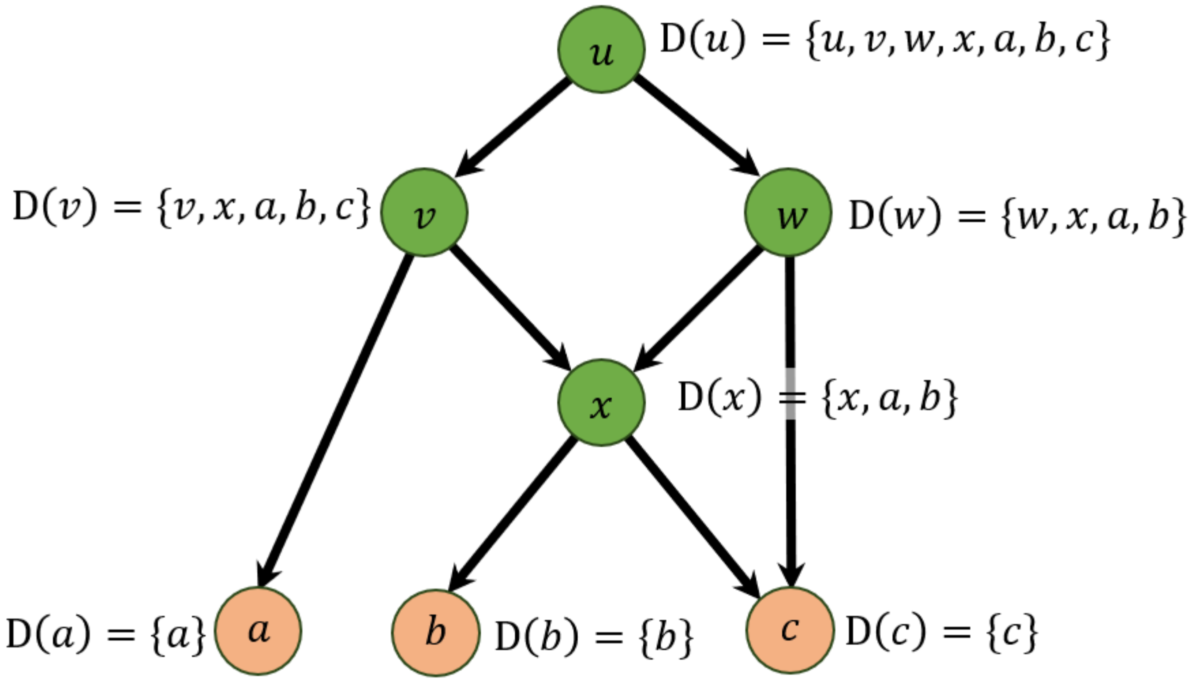
Hasse Diagram



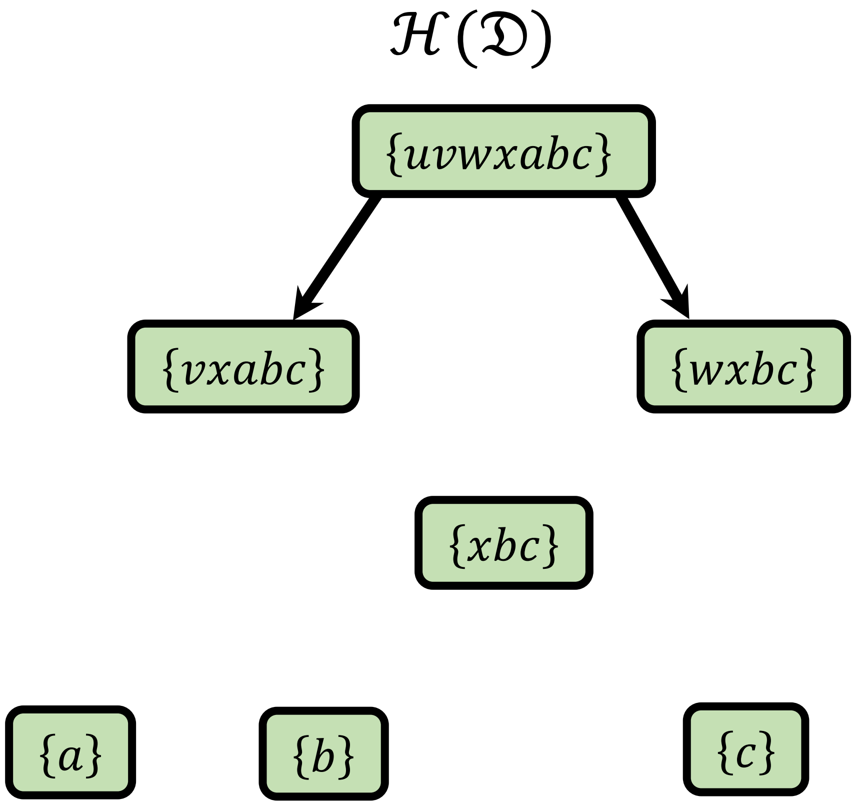
$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$



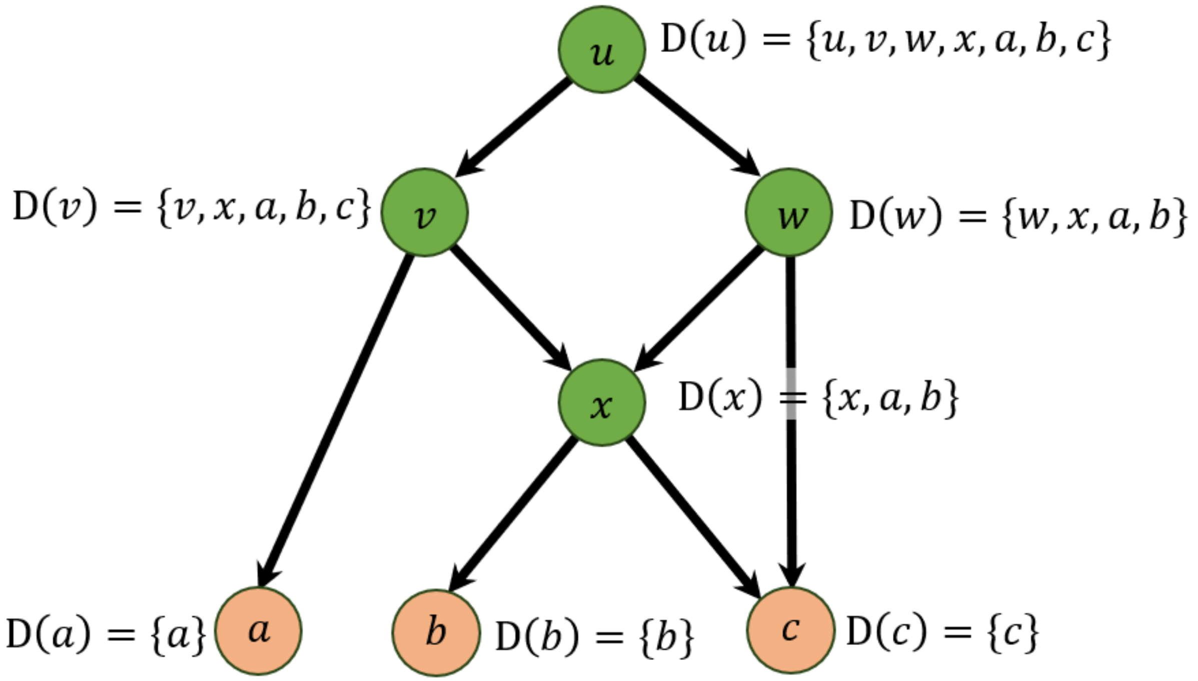
Hasse Diagram



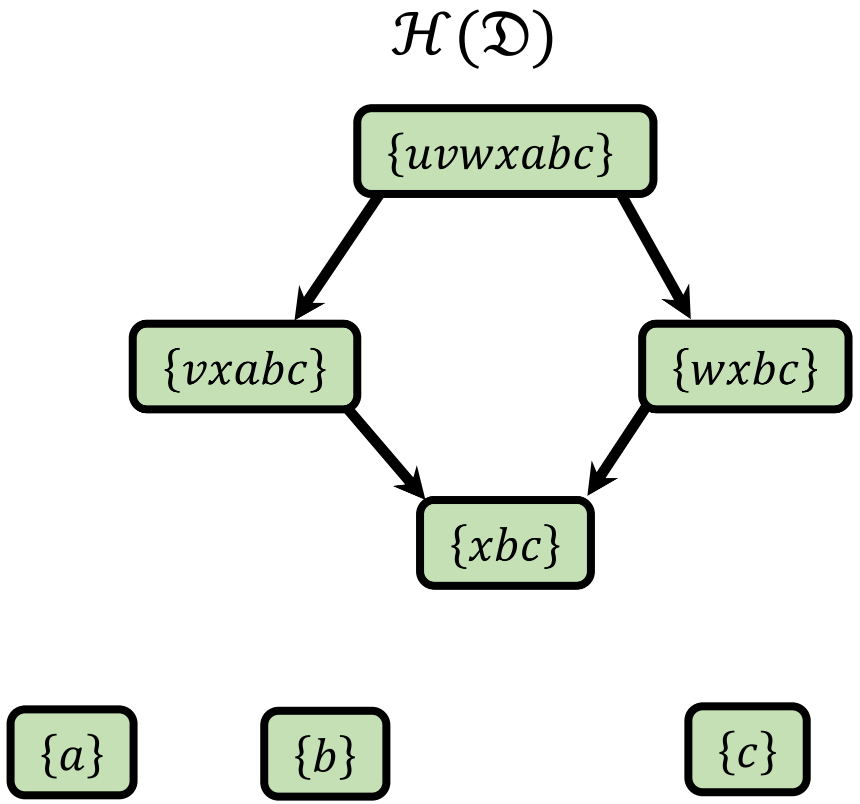
$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$



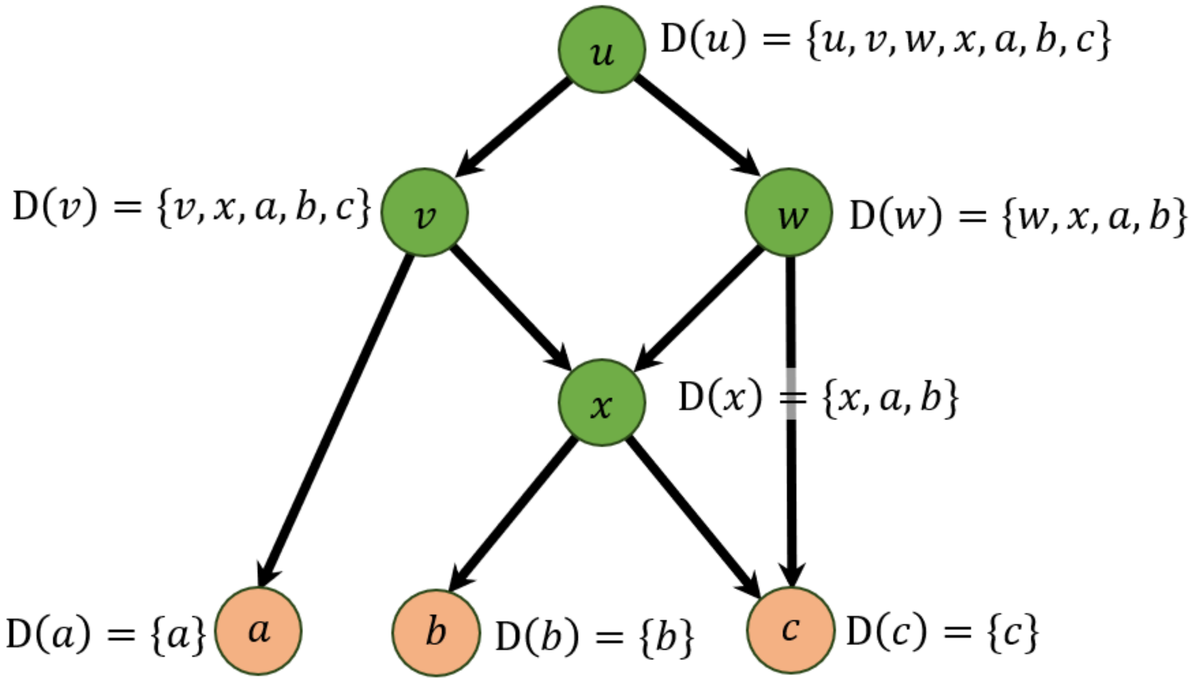
Hasse Diagram



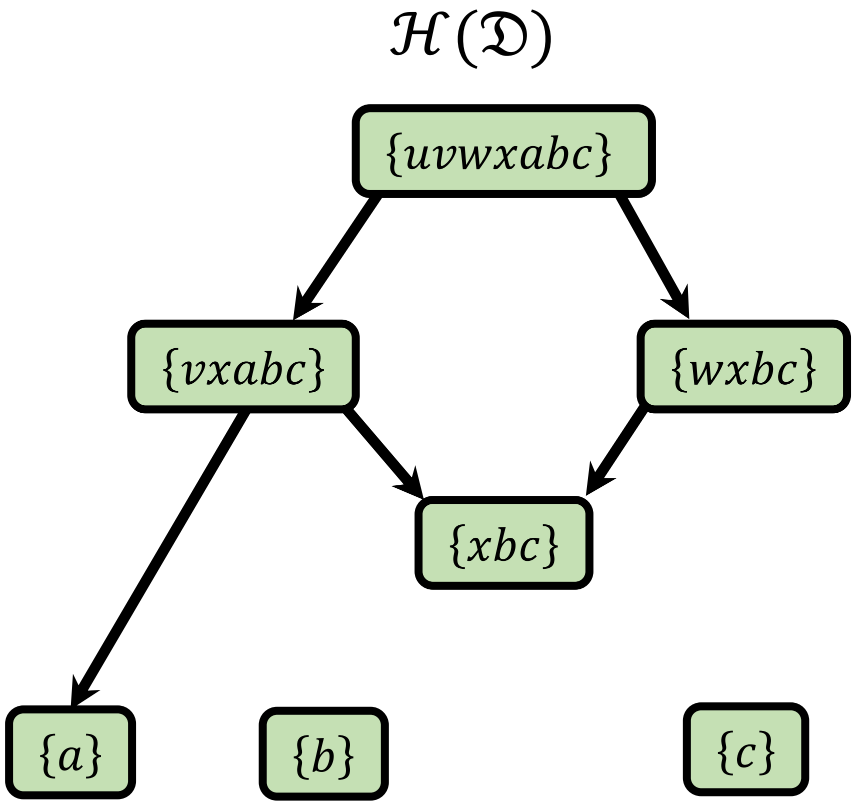
$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$



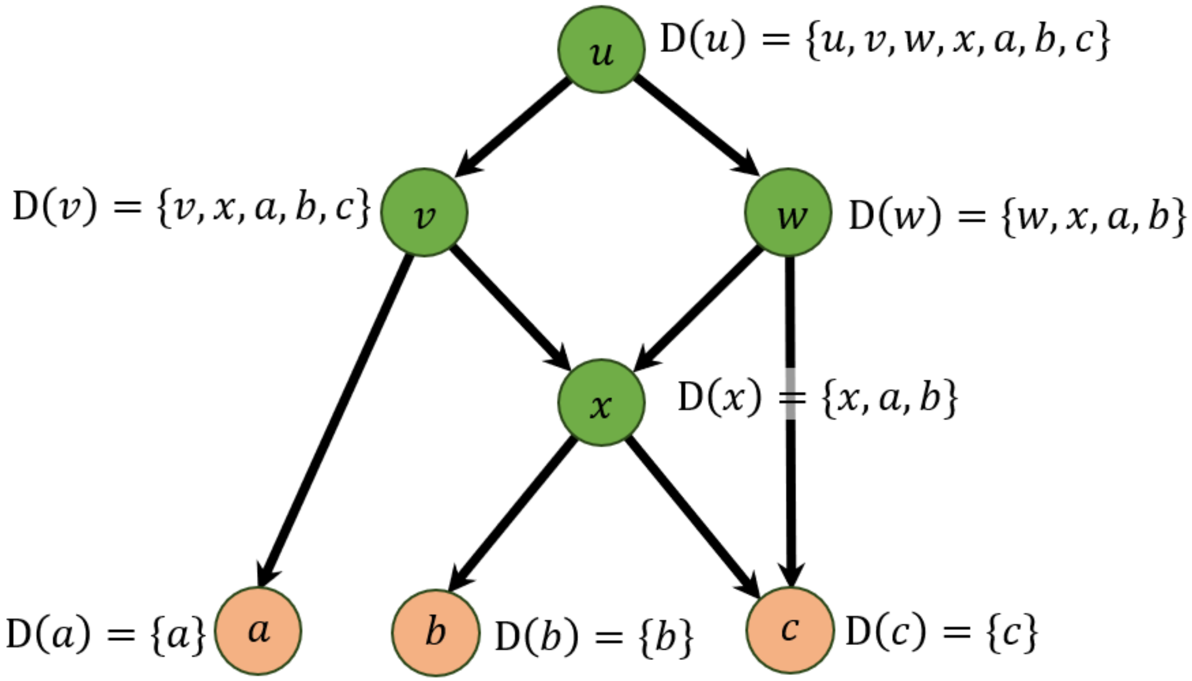
Hasse Diagram



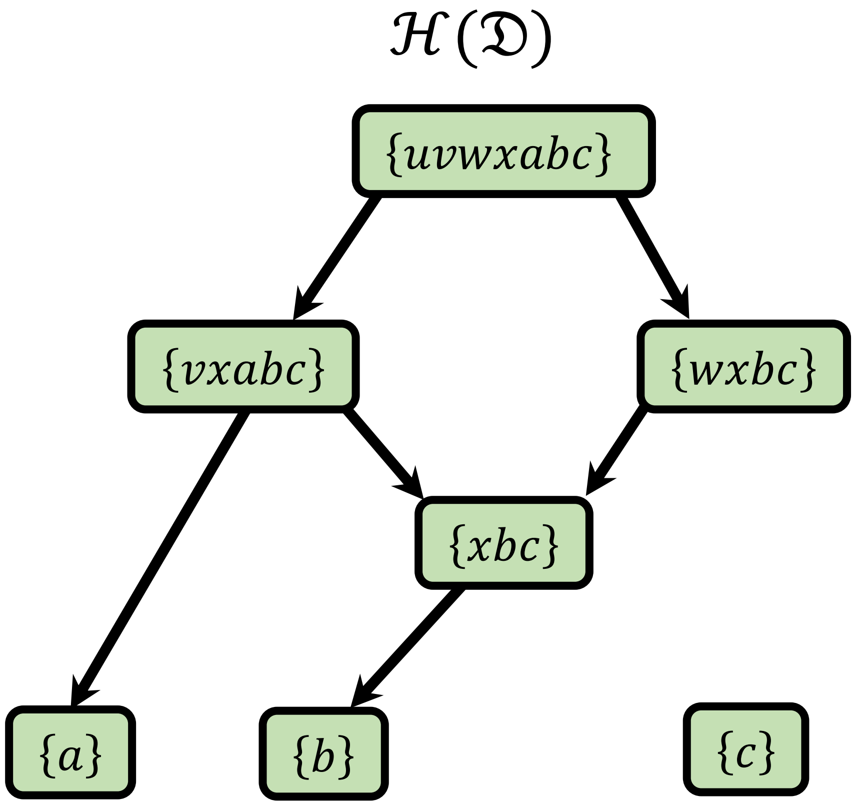
$$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$$



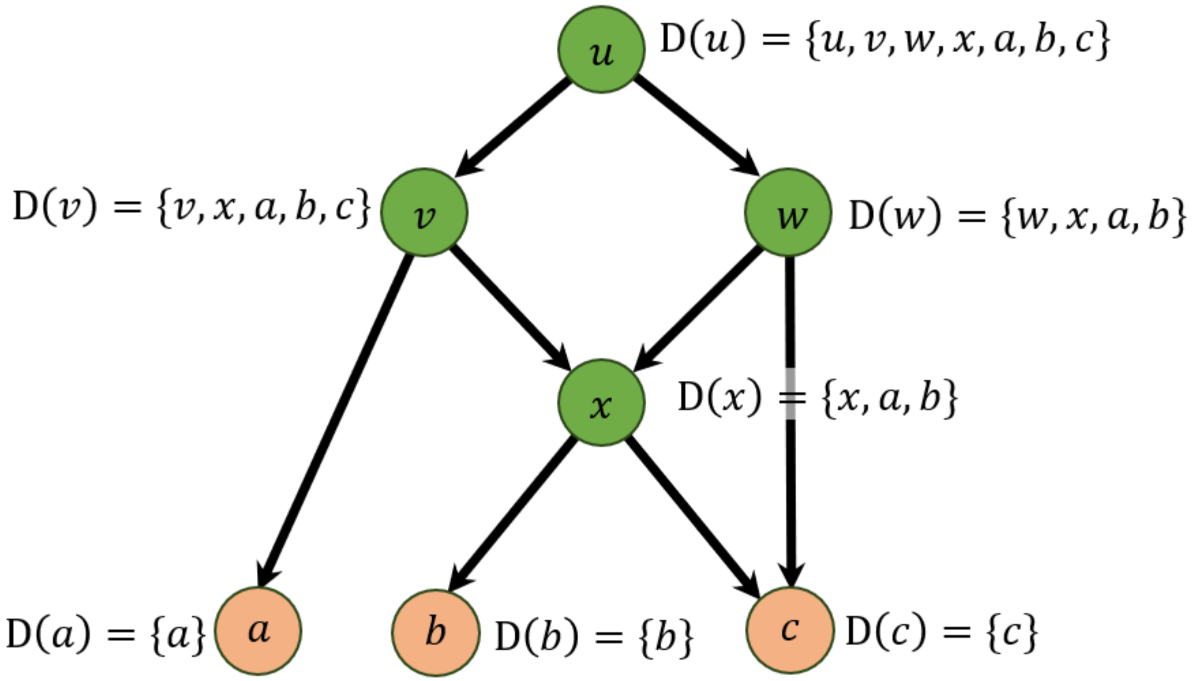
Hasse Diagram



$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

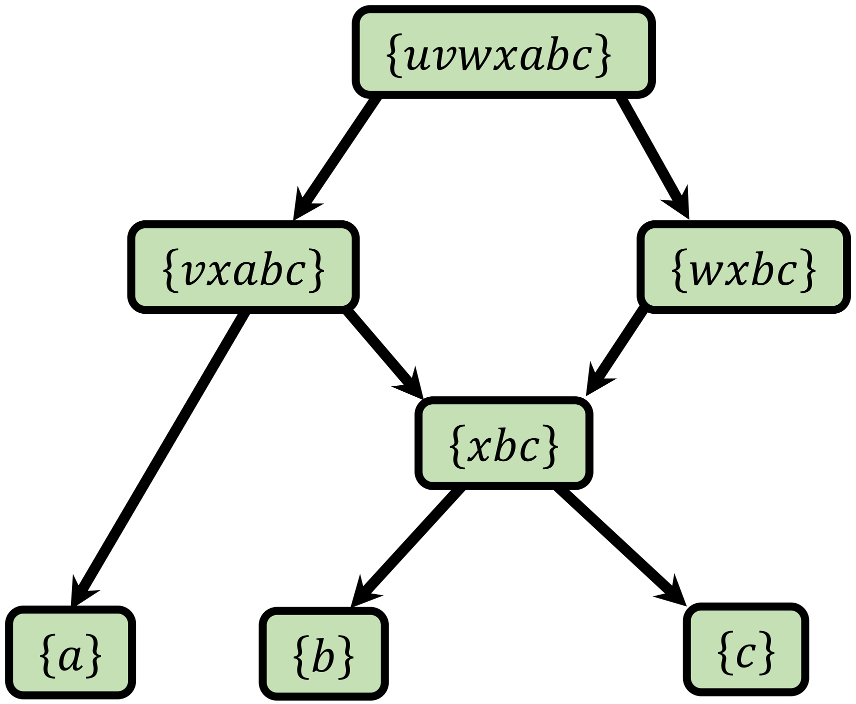


Hasse Diagram

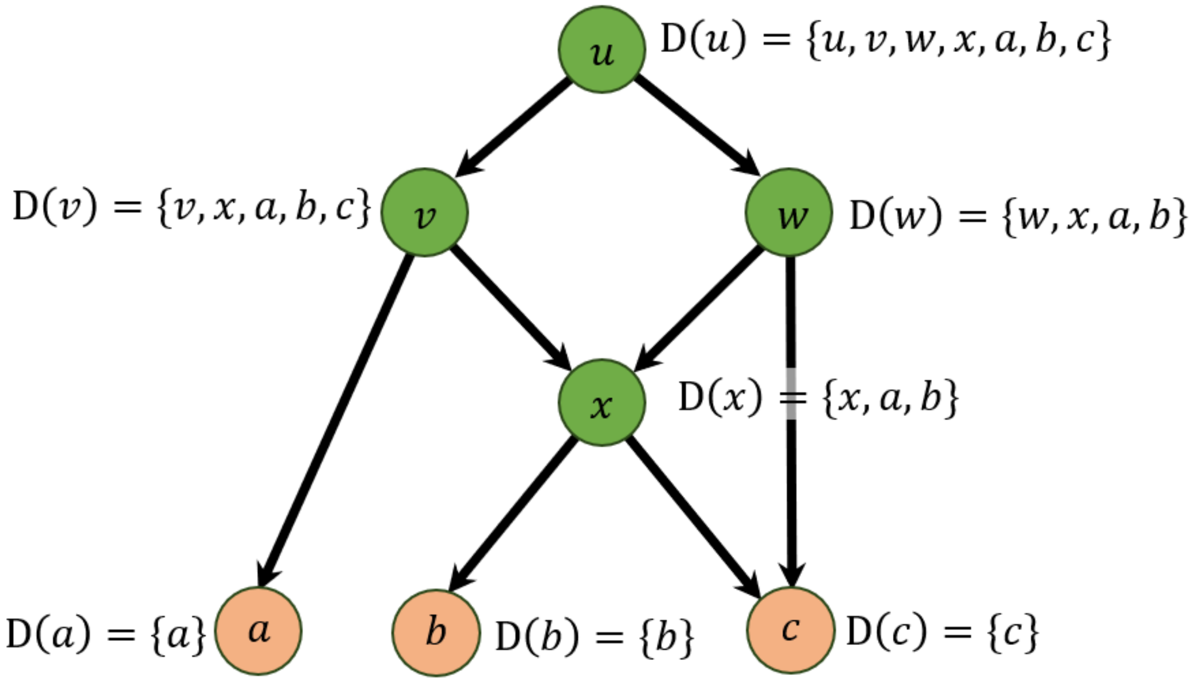


$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

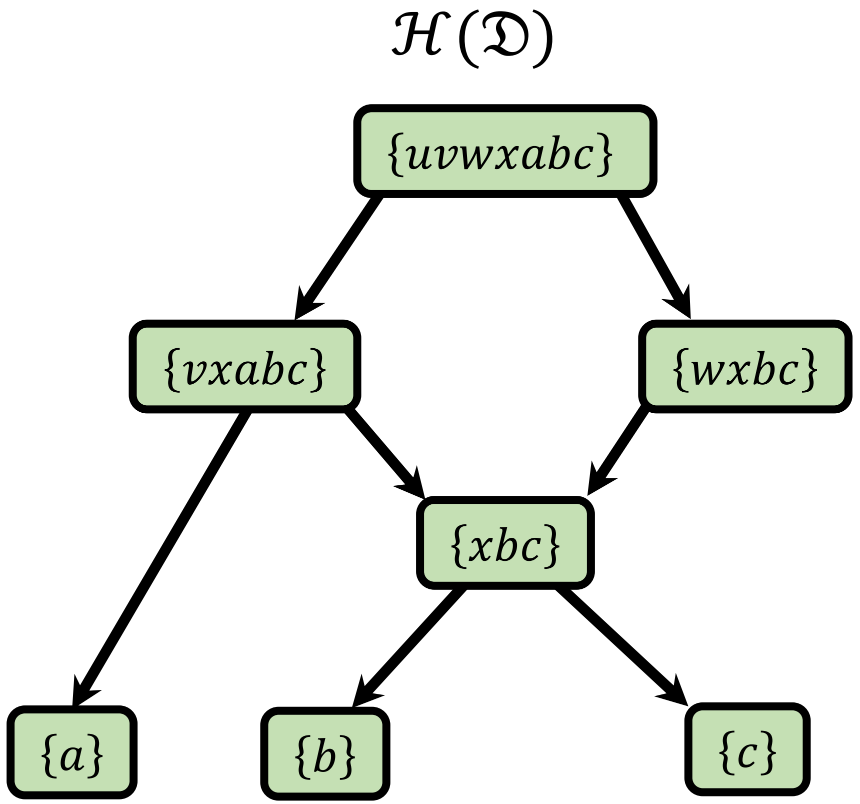
$\mathcal{H}(\mathfrak{D})$



Hasse Diagram

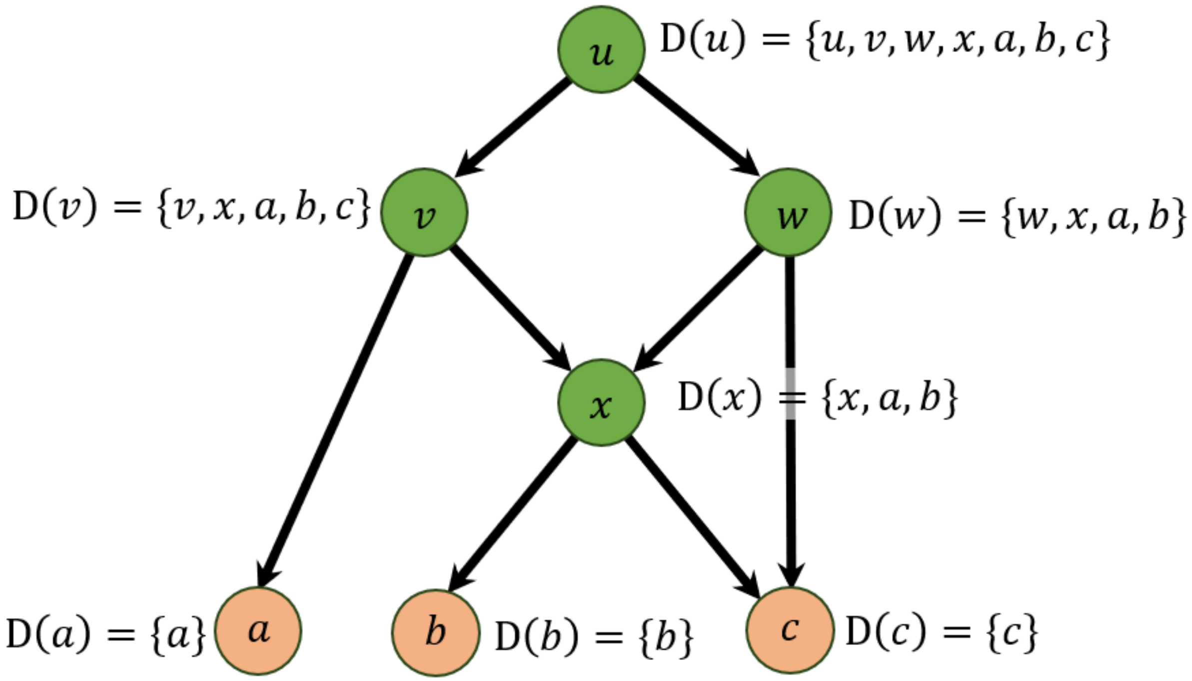


$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$

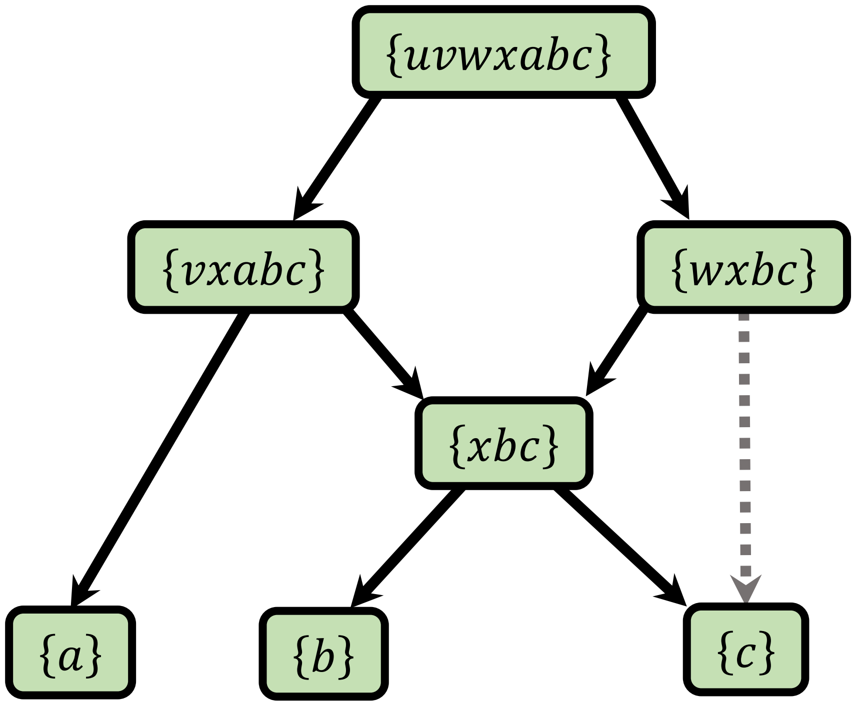


Are G and $\mathcal{H}(\mathfrak{D})$ connected?

Hasse Diagram

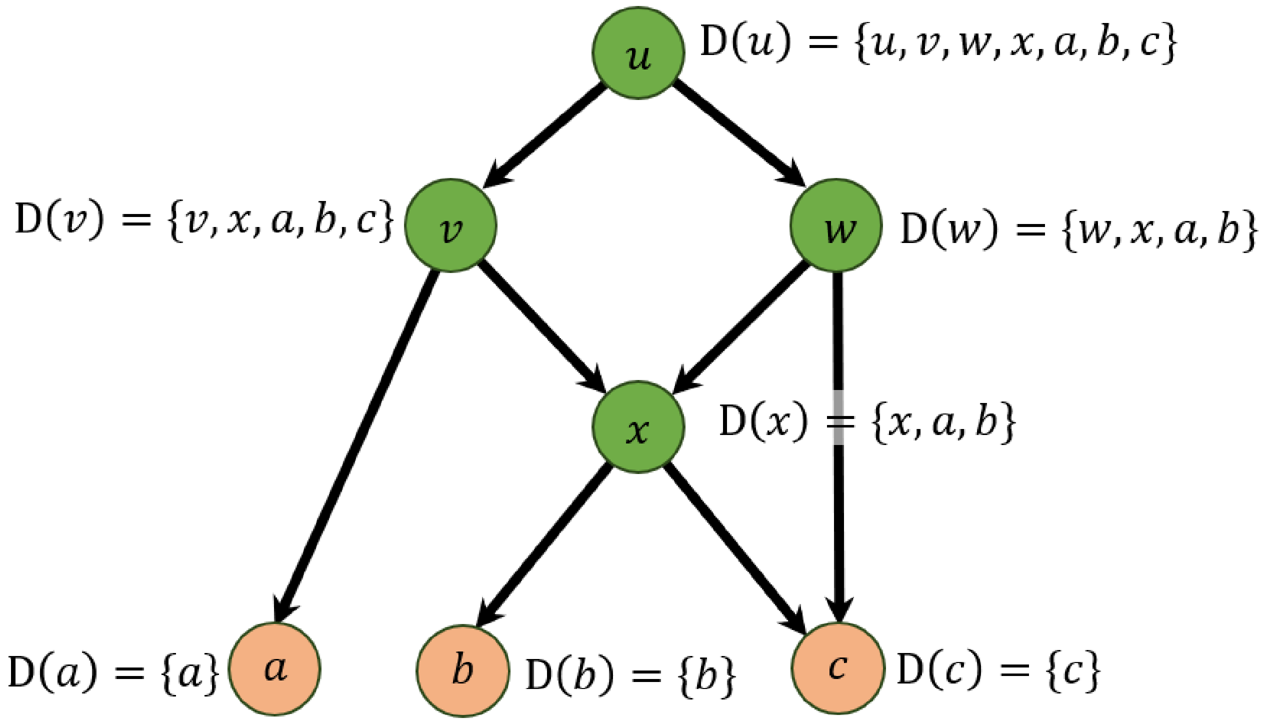


$\mathfrak{D} = \{\{uvwxabc\}, \{vxabc\}, \{wxbc\}, \{xbc\}, \{a\}, \{b\}, \{c\}\}$



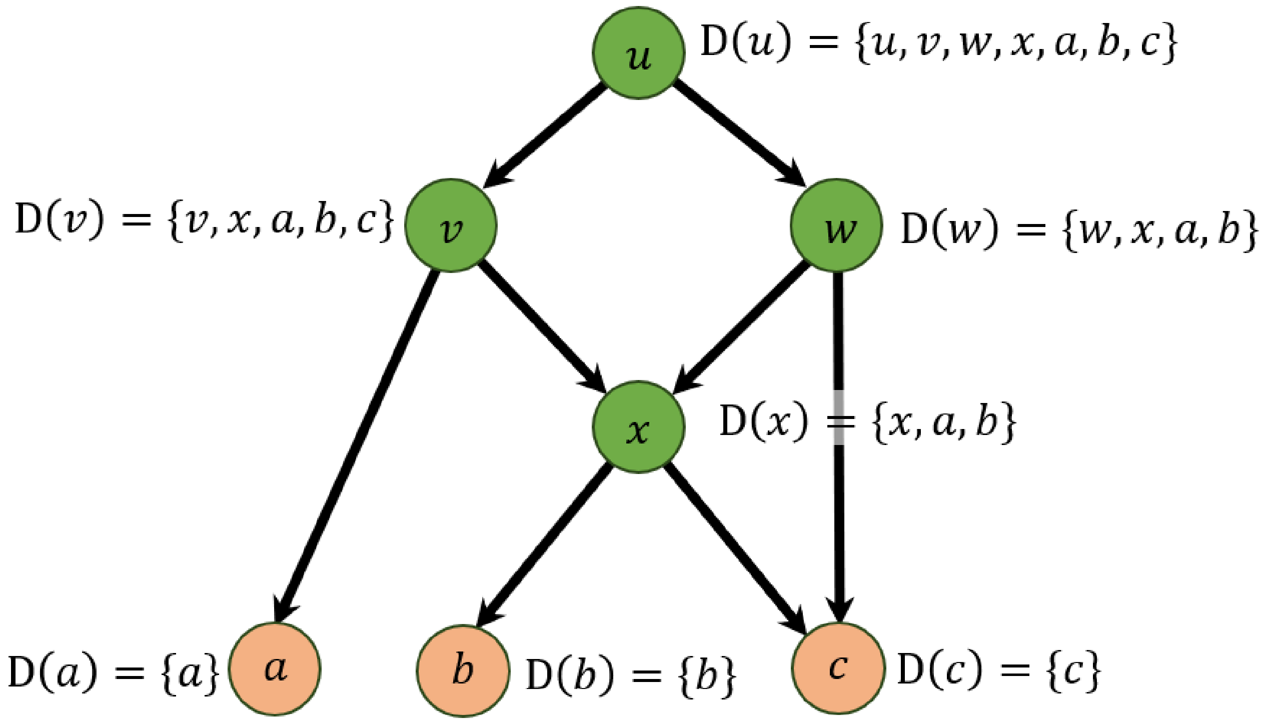
Are G and $\mathcal{H}(\mathfrak{D})$ connected?

Some Descendant Clusters Properties



Some Descendant Clusters Properties

Let G be a DAG, then:

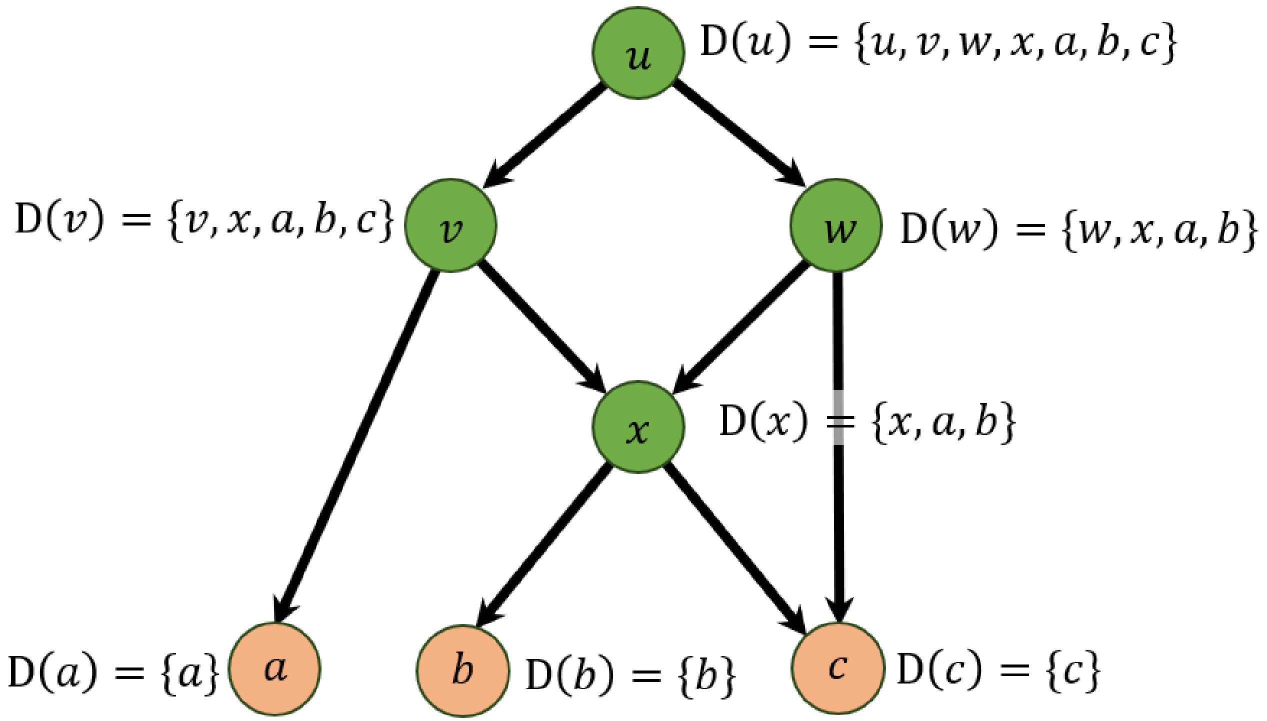


Some Descendant Clusters Properties

Let G be a DAG, then:

(*i*) $D(u) = D(v)$ if and only if $u = v$

→ Assume $D(u) = D(v), u \neq v$
→ $u \in D(v), v \in D(u)$
→ paths $u \dots v, v \dots u$ in G
→ G contains a cycle ⚡



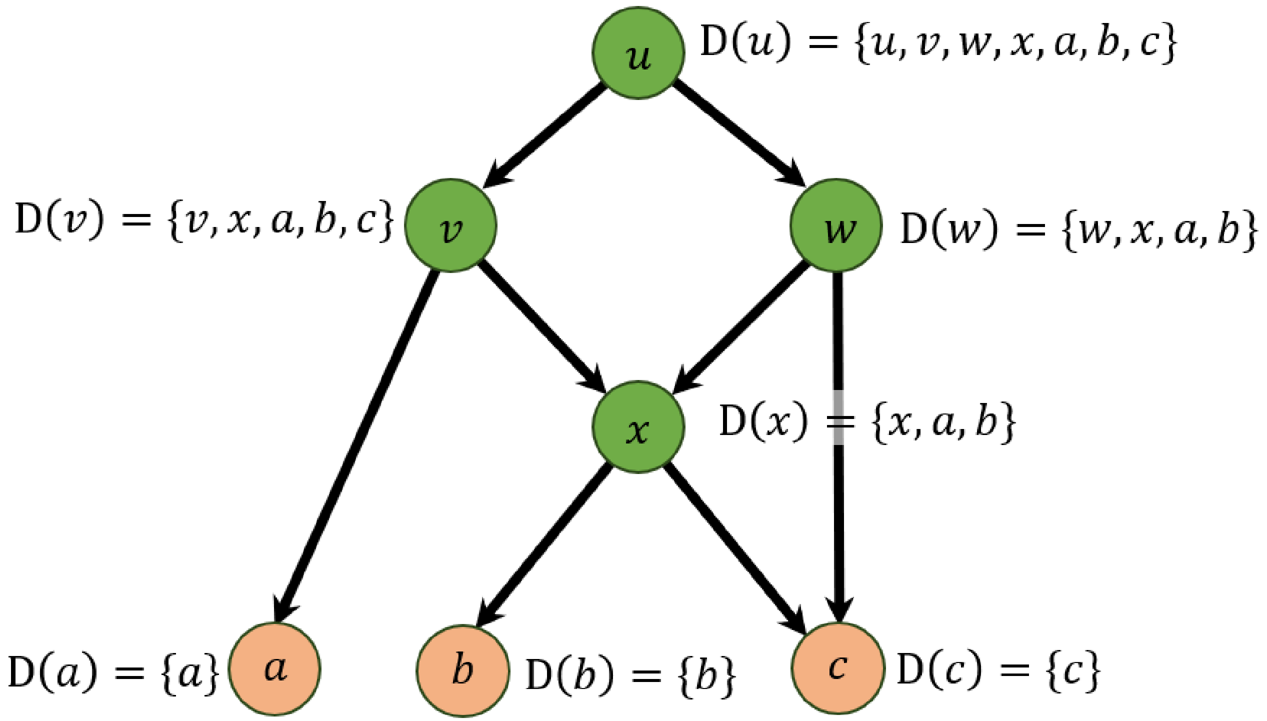
Some Descendant Clusters Properties

Let G be a DAG, then:

(*i*) $\boldsymbol{D(u) = D(v)}$ **if and only if** $\boldsymbol{u = v}$

→ Assume $D(u) = D(v), u \neq v$
→ $u \in D(v), v \in D(u)$
→ paths $u \dots v, v \dots u$ in G
→ G contains a cycle ⚡

(ii) $\boldsymbol{v \in D(u)}$ **if and only if** $\boldsymbol{D(v) \subseteq D(u)}$



Some Descendant Clusters Properties

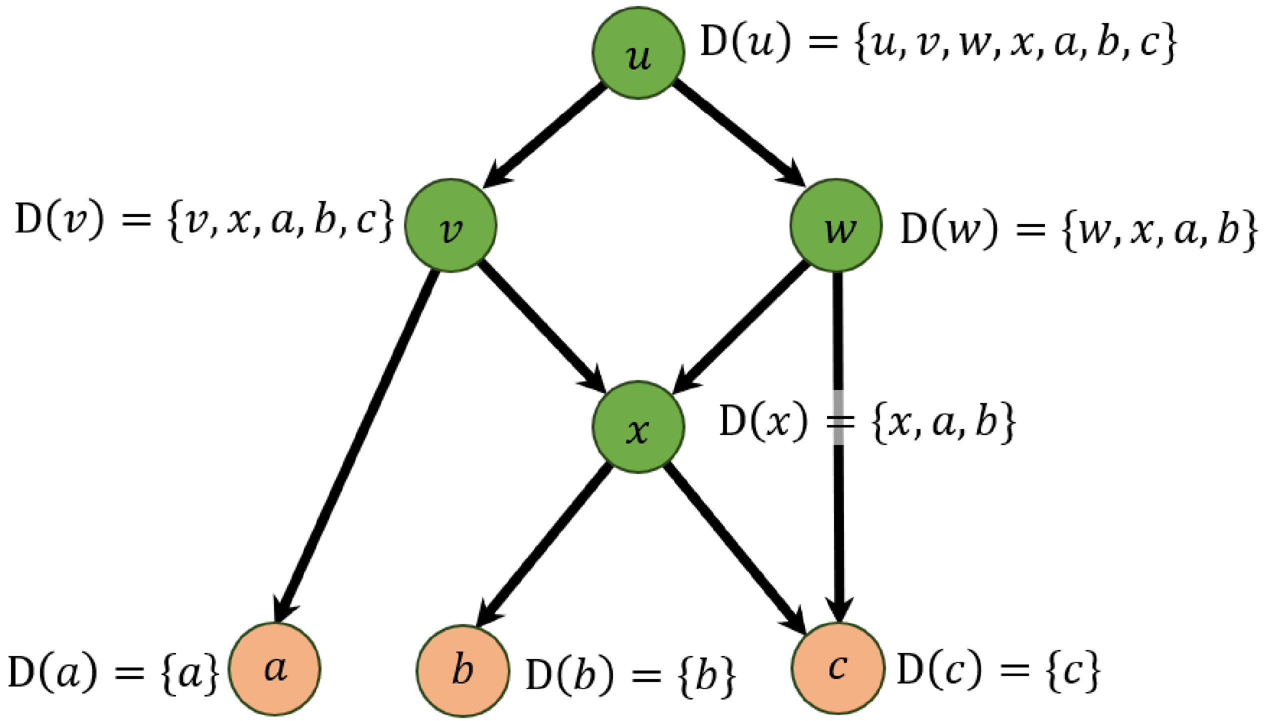
Let G be a DAG, then:

(*i*) $D(u) = D(v)$ if and only if $u = v$

→ Assume $D(u) = D(v), u \neq v$
→ $u \in D(v), v \in D(u)$
→ paths $u \dots v, v \dots u$ in G
→ G contains a cycle ⚡

(ii) $v \in D(u)$ if and only if $D(v) \subseteq D(u)$

→ $v \in D(u)$, then there is path $u \dots v$.
  → $\forall x \in D(v)$ there is path $v \dots x$.
  → $\forall x \in D(v)$ there is path $u \dots v \dots x$ (acyclicity)
  → consequently $D(v) \subseteq D(u)$.



Some Descendant Clusters Properties

Let G be a DAG, then:

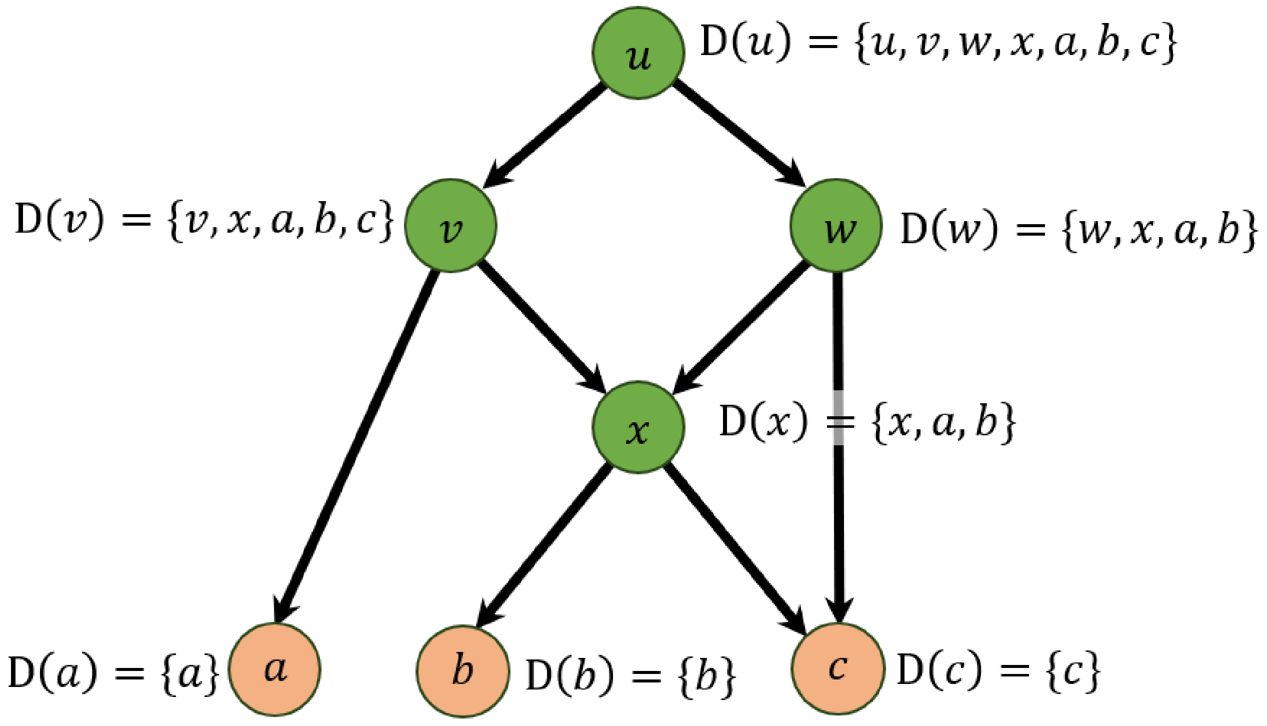
(*i*) $D(u) = D(v)$ if and only if $u = v$

→ Assume $D(u) = D(v), u \neq v$
→ $u \in D(v), v \in D(u)$
→ paths $u \dots v, v \dots u$ in G
→ G contains a cycle ⚡

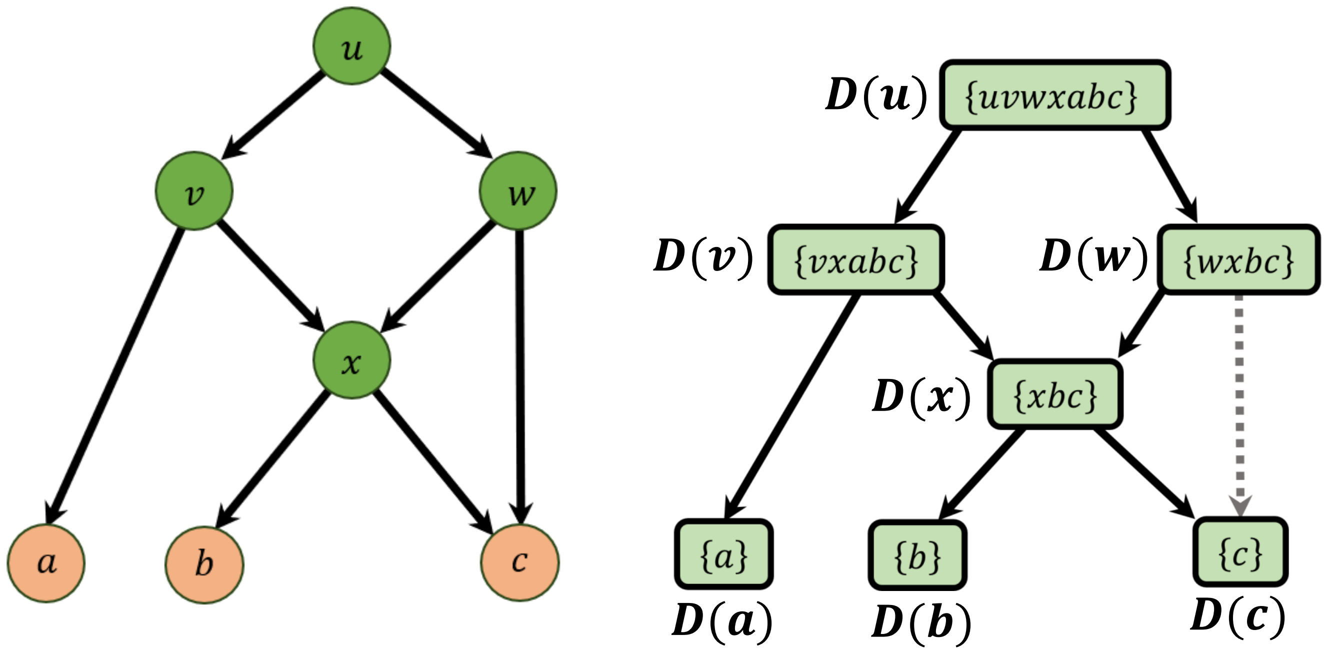
(ii) $v \in D(u)$ if and only if $D(v) \subseteq D(u)$

→ $v \in D(u)$, then there is path $u \dots v$.
  → $\forall x \in D(v)$ there is path $v \dots x$.
  → $\forall x \in D(v)$ there is path $u \dots v \dots x$ (acyclicity)
  → consequently $D(v) \subseteq D(u)$.

→ $D(v) \subseteq D(u)$
  → $v \in D(v), u \in D(u)$ by definition
  → $D(v) \subseteq D(u)$ implies $v \in D(u)$

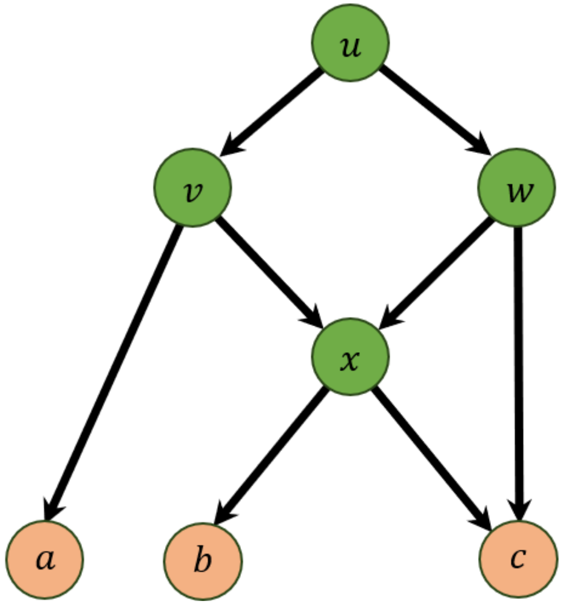


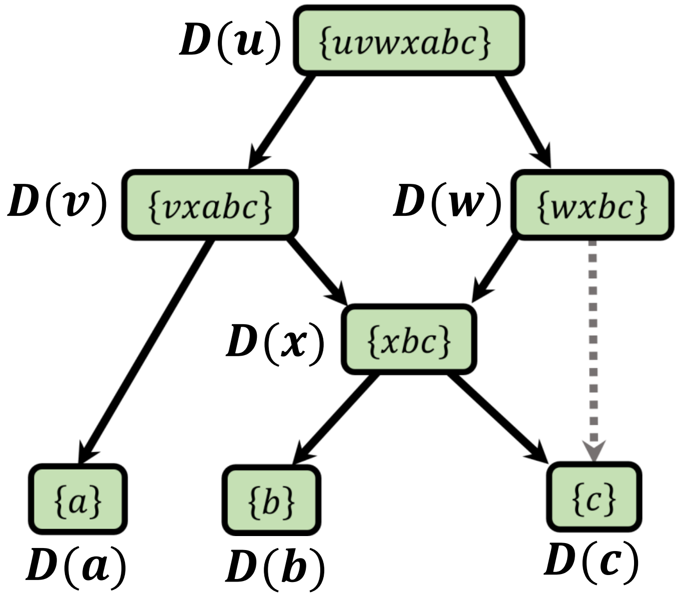
Hasse Diagram Isomorphism



Hasse Diagram Isomorphism

$(i)\ D(u) = D(v)$ *if and only if* $u = v$

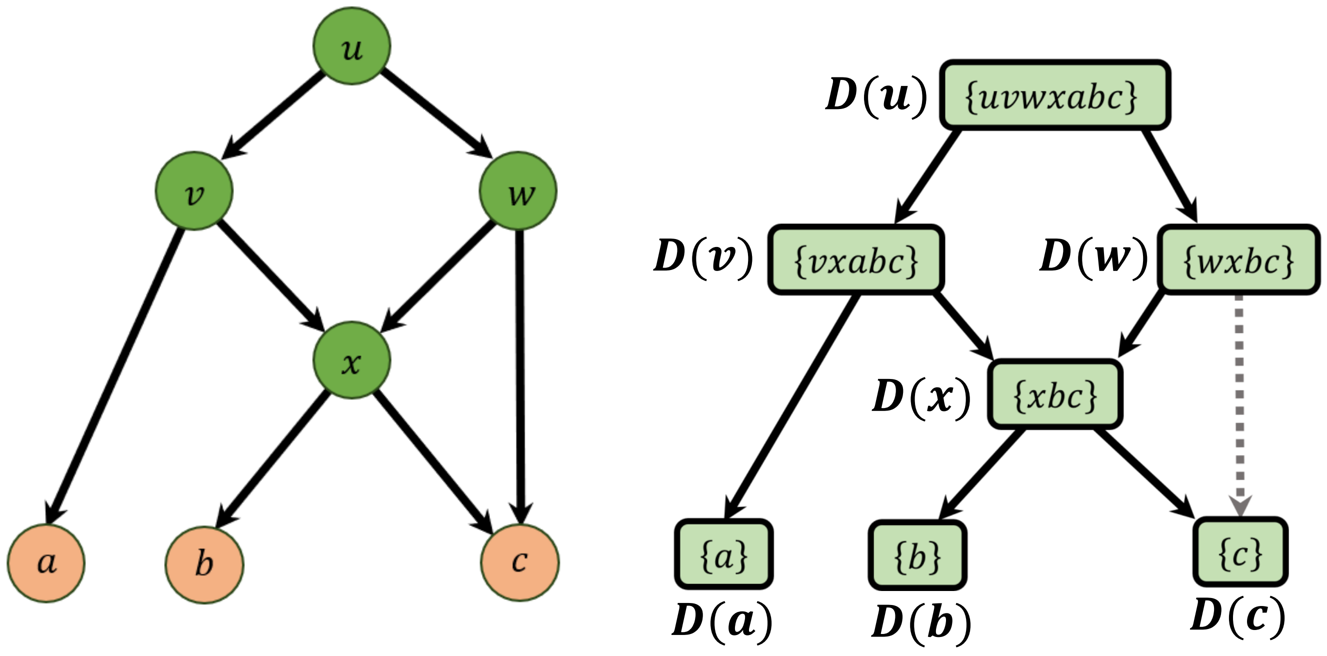




Hasse Diagram Isomorphism

$\boldsymbol{(i)\ D(u) = D(v)\ if\ and\ only\ if\ u = v}$

- (i) yields bijection between $V(G)$ and $\mathfrak{D}_G$
  → $\phi: V(G) \to \mathfrak{D},\ \phi(u) \coloneqq D(u)$

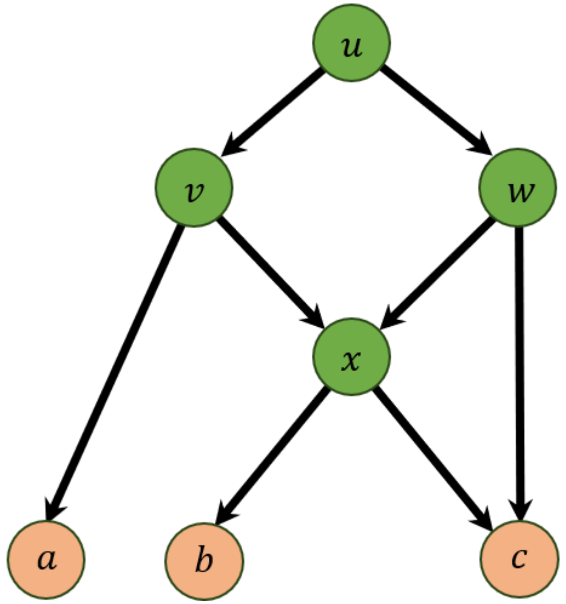


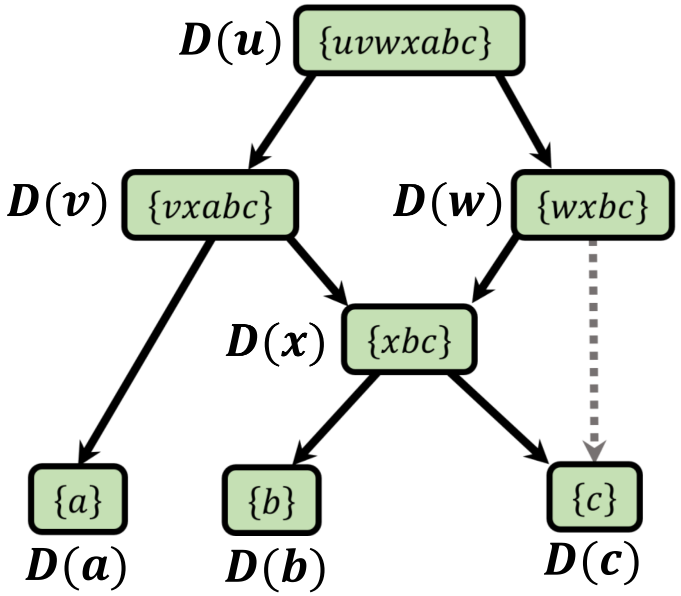
Hasse Diagram Isomorphism

$(\boldsymbol{i})\ \boldsymbol{D(u) = D(v)\ if\ and\ only\ if\ u = v}$

- (i) yields bijection between $V(G)$ and $\mathfrak{D}_G$
  → $\phi: V(G) \to \mathfrak{D},\ \ \phi(u) \coloneqq D(u)$

$(\boldsymbol{ii})\ \boldsymbol{v \in D(u)}$ **if and only if** $\boldsymbol{D(v) \subseteq D(u)}$





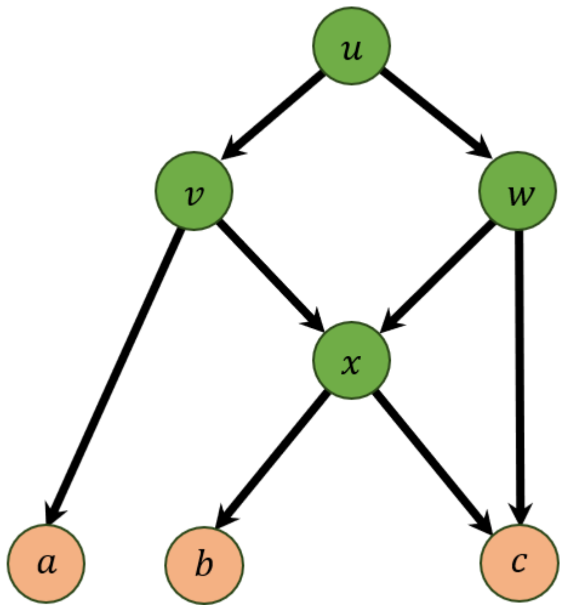
Hasse Diagram Isomorphism

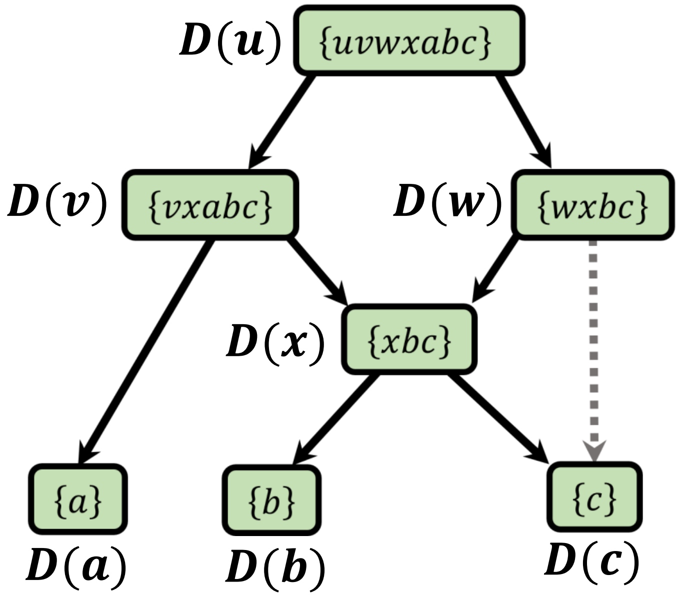
$\boldsymbol{(i)\ D(u) = D(v)\ if\ and\ only\ if\ u = v}$

- (i) yields bijection between $V(G)$ and $\mathfrak{D}_G$
  → $\phi: V(G) \to \mathfrak{D},\ \ \phi(u) \coloneqq D(u)$

$\boldsymbol{(ii)\ v \in D(u)}$ **if and only if** $\boldsymbol{D(v) \subseteq D(u)}$

- **Let** $\boldsymbol{(u, v) \in E(G)}$**.**
  → if (u, v) is a shortcut, there is a $w \in D(u)$
  with $D(v) \subseteq D(w) \subseteq D(u), \big(D(u), D(v)\big) \notin E(\mathcal{H}(\mathfrak{D}))$
  → if (u, v) is not a shortcut there is no such w
  and $\big(D(u), D(v)\big) \in E(\mathcal{H}(\mathfrak{D}))$





Hasse Diagram Isomorphism

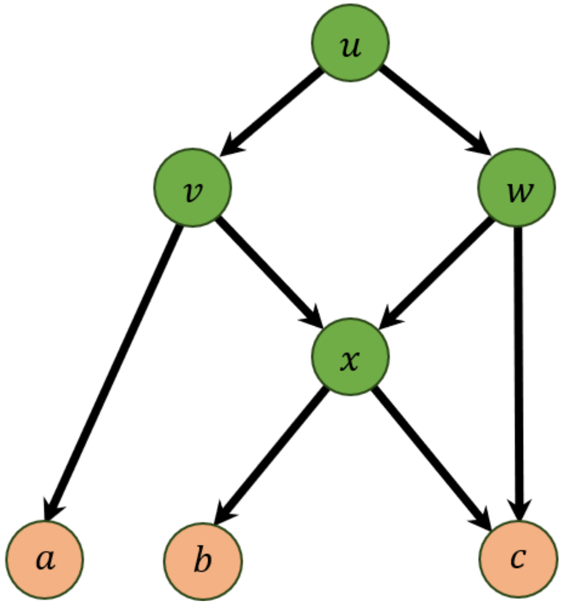
$(i)\ D(u) = D(v)$ *if and only if* $u = v$

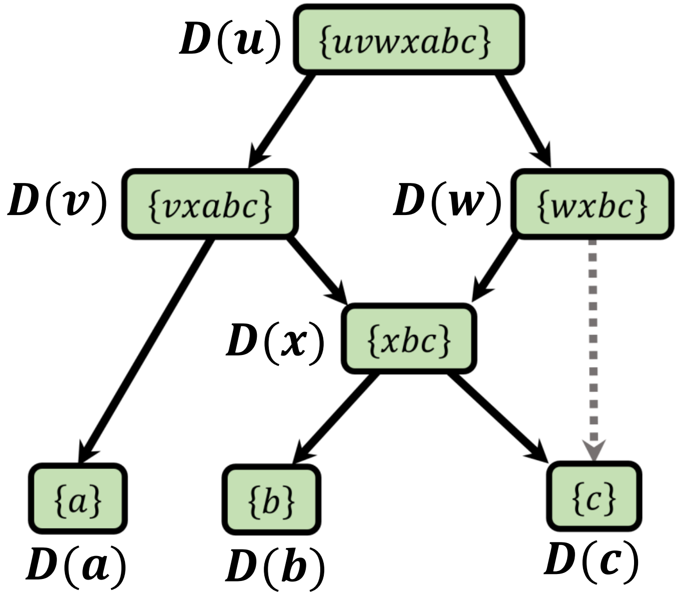
- (i) yields bijection between $V(G)$ and $\mathfrak{D}_G$
  → $\phi: V(G) \to \mathfrak{D},\ \phi(u) \coloneqq D(u)$

$(ii)\ v \in D(u)$ if and only if $D(v) \subseteq D(u)$

- **Let $(u, v) \in E(G)$.**
  → if (u, v) is a shortcut, there is a $w \in D(u)$ with $D(v) \subseteq D(w) \subseteq D(u), \big(D(u), D(v)\big) \notin E(\mathcal{H}(\mathfrak{D}))$
  → if (u, v) is not a shortcut there is no such w and $\big(D(u), D(v)\big) \in E(\mathcal{H}(\mathfrak{D}))$

- **Let, $\big(D(u), D(v)\big) \in E\big(\mathcal{H}(\mathfrak{D})\big)$**
  → $\nexists w$ distinct from u, v with $D(v) \subseteq D(w) \subseteq D(u)$
  → $D(v) \subseteq D(u)$ and path $P = u \dots v$ in G
  → $|P| > 1$ then $P = u \dots w \dots v$ and $D(v) \subseteq D(w) \subseteq D(u)$ ⚡
  → $|P|=1$ and $(u, v) \in E(G)$





Hasse Diagram Isomorphism

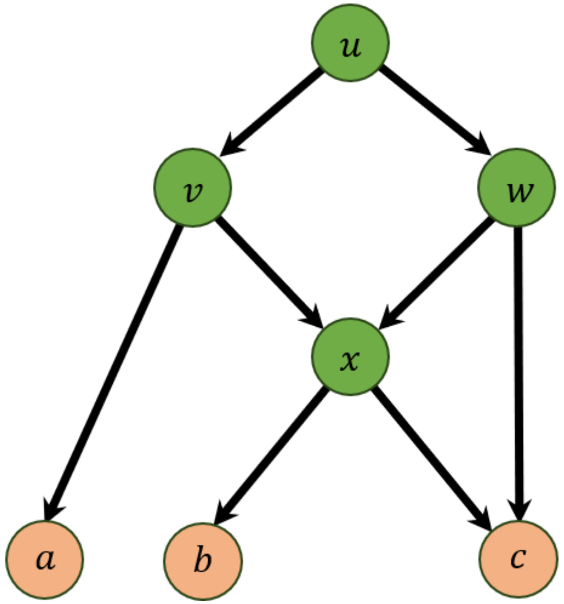
(i) $D(u) = D(v)$ *if and only if* $u = v$

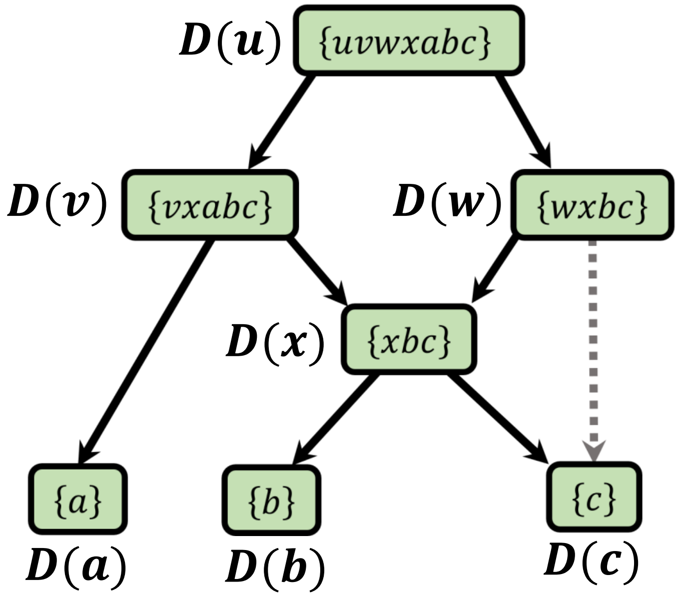
- (i) yields bijection between $V(G)$ and $\mathfrak{D}_G$
  → $\phi: V(G) \to \mathfrak{D},\ \phi(u) \coloneqq D(u)$

(ii) $v \in D(u)$ **if and only if** $D(v) \subseteq D(u)$

- **Let** $(u, v) \in E(G)$.
  → if (u, v) is a shortcut, there is a $w \in D(u)$ with $D(v) \subseteq D(w) \subseteq D(u), \big(D(u), D(v)\big) \notin E(\mathcal{H}(\mathfrak{D}))$
  → if (u, v) is not a shortcut there is no such w and $\big(D(u), D(v)\big) \in E(\mathcal{H}(\mathfrak{D}))$

- **Let,** $\big(D(u), D(v)\big) \in E\big(\mathcal{H}(\mathfrak{D})\big)$
  → $\nexists w$ distinct from u, v with $D(v) \subseteq D(w) \subseteq D(u)$
  → $D(v) \subseteq D(u)$ and path $P = u \dots v$ in G
  → $|P| > 1$ then $P = u \dots w \dots v$ and $D(v) \subseteq D(w) \subseteq D(u)$ ⚡
  → $|P|=1$ and $(u, v) \in E(G)$





G is isomorphic to $\mathcal{H}(\mathfrak{D})$ **(minus some shortcuts)**
or:
the transitive reduction of G is isomorphic to $\mathcal{H}(\mathfrak{D})$.

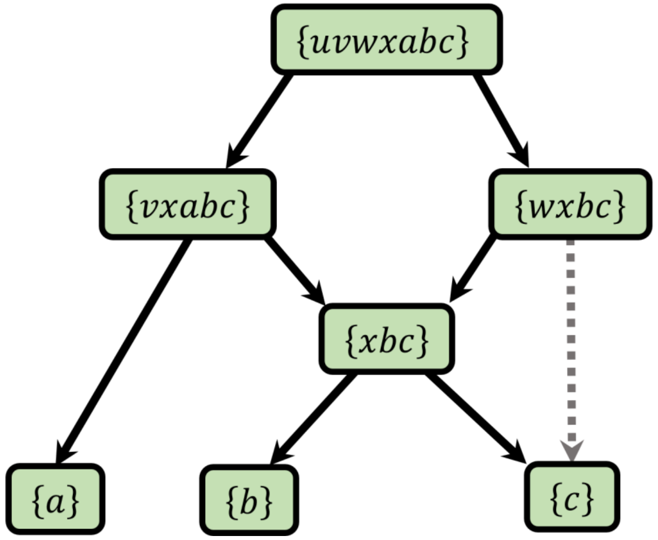
Descendant Clusters

Descendant Clusters

- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?

Descendant Clusters

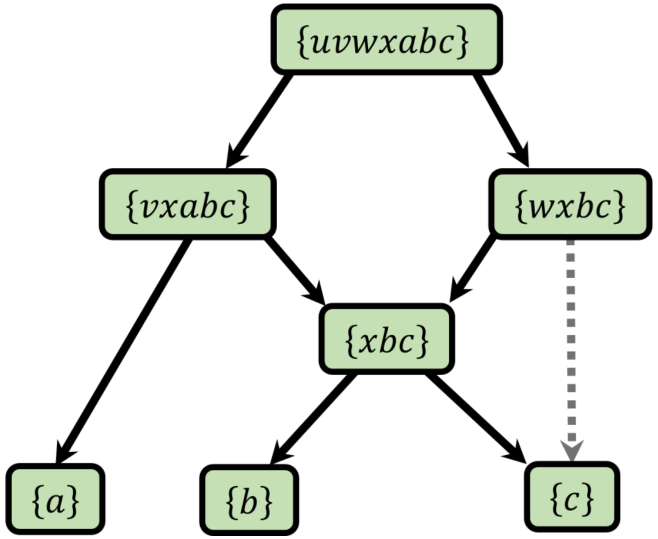
- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?



Descendant Clusters

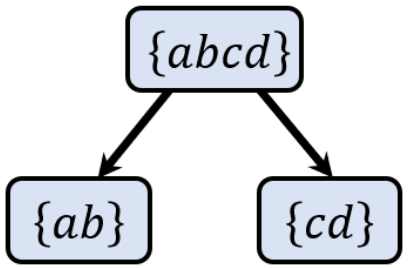
- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?

- Not the case for every set system.
  → Maybe we need **more sets**?
  (at least as many sets as "vertices")
  → Maybe we need **all singletons**?
  → Maybe we need **...**

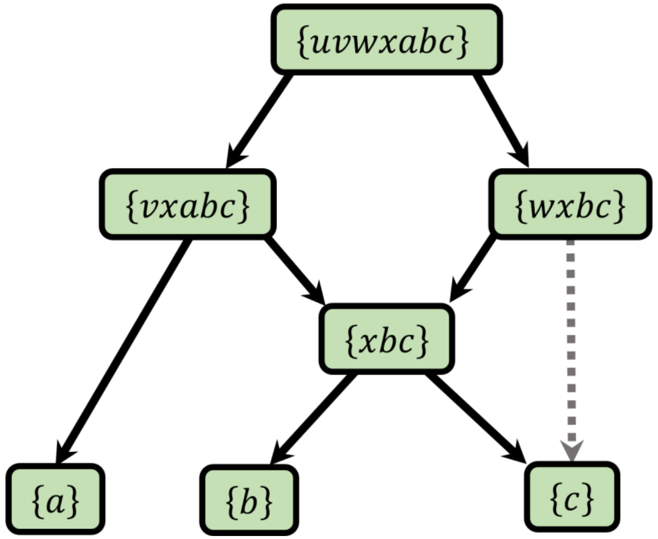


Descendant Clusters

- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?



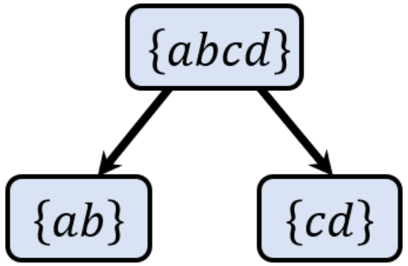
- Not the case for every set system.
  → Maybe we need **more sets**?
  (at least as many sets as "vertices")
  → Maybe we need **all singletons**?
  → Maybe we need **...**

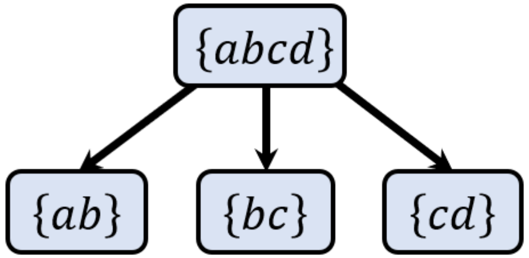


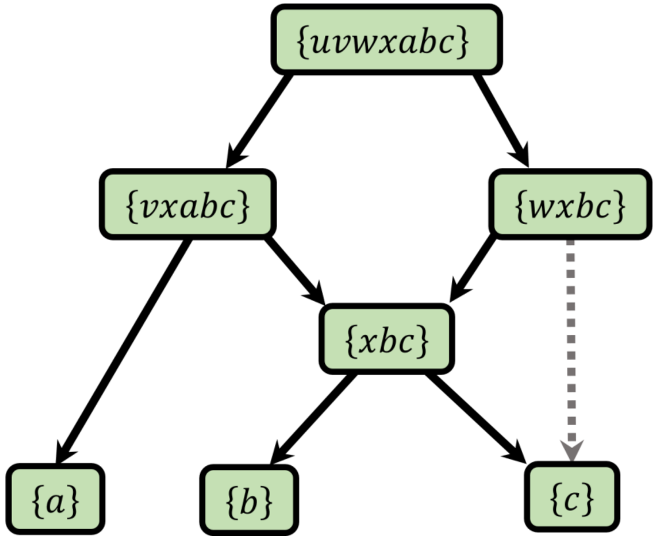
Descendant Clusters

- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?

- Not the case for every set system.
  → Maybe we need **more sets**?
  (at least as many sets as “vertices”)
  → Maybe we need **all singletons**?
  → Maybe we need **...**



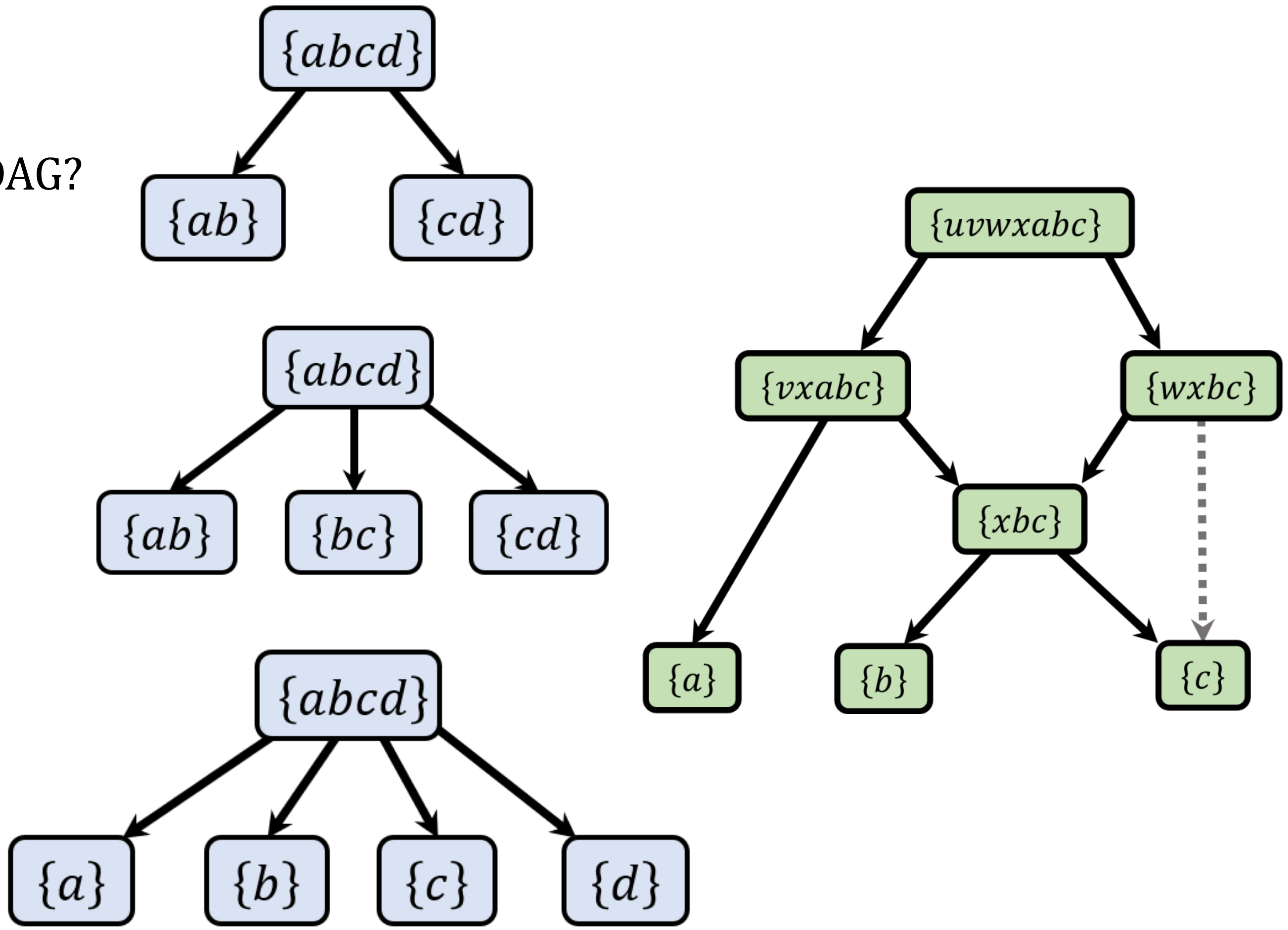




Descendant Clusters

- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?

- Not the case for every set system.
  → Maybe we need **more sets**?
  (at least as many sets as "vertices")
  → Maybe we need **all singletons**?
  → Maybe we need **...**

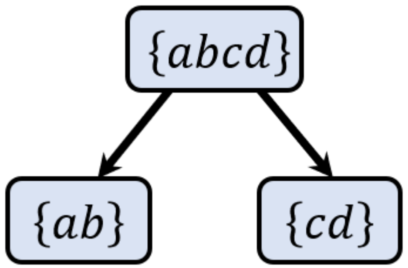


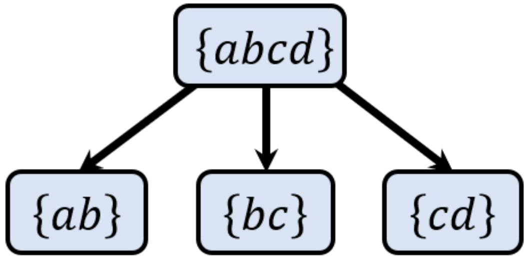
Descendant Clusters

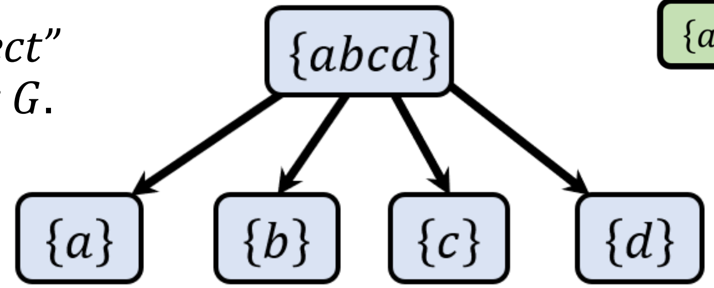
- Given a set system $\mathfrak{S}$:
  → when is $\mathfrak{S}$ equal to the descendant clusters of a DAG?

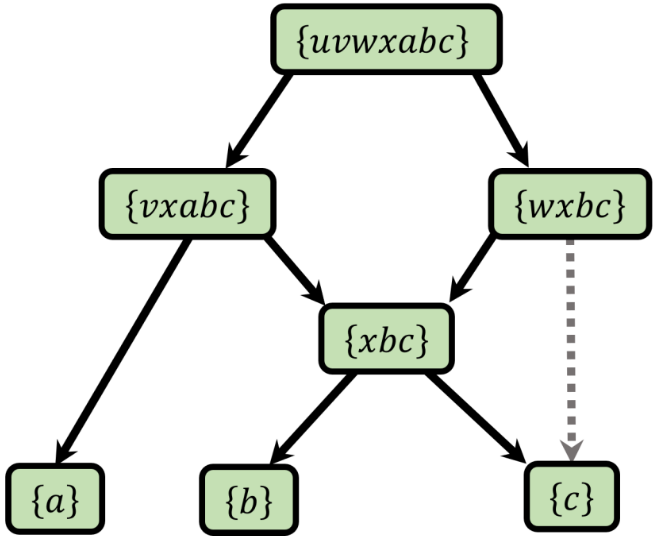
- Not the case for every set system.
  → Maybe we need **more sets**?
  (at least as many sets as "vertices")
  → Maybe we need **all singletons**?
  → Maybe we need **...**

- Maybe we start with a function that finds the *"correct"* descendant set in $\mathfrak{D}$ for a vertex without consulting G.









D-Snake! $\widetilde{D}$

D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{U}$
  → $\boldsymbol{\widetilde{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

D-Snake! $\widetilde{D}$

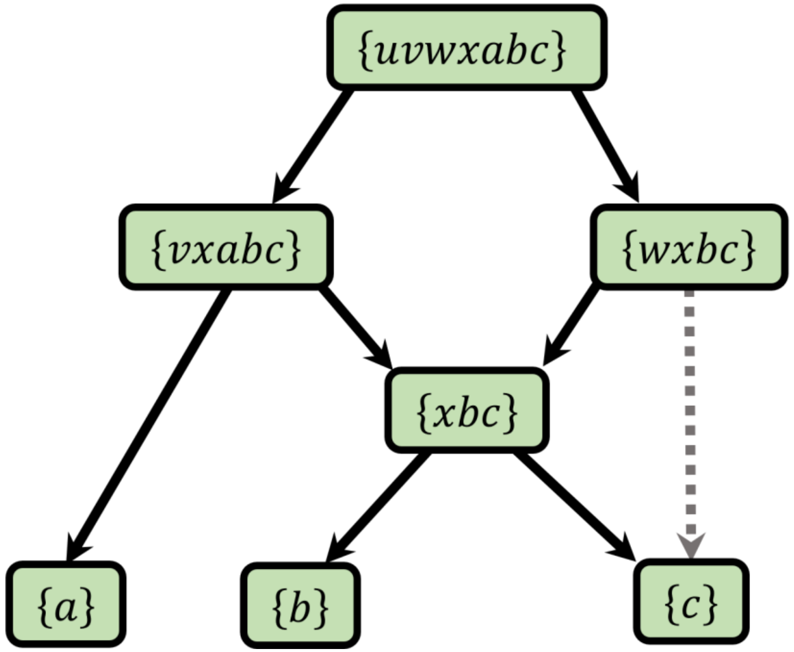
- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U}'$ **for all** $\boldsymbol{U}' \in \boldsymbol{U}$
  → $\boldsymbol{\widetilde{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

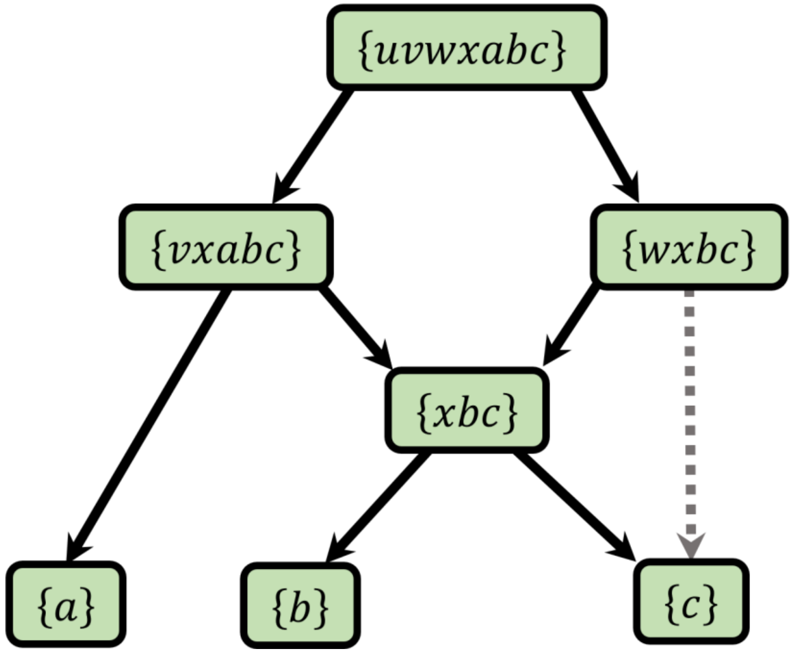


D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u)$

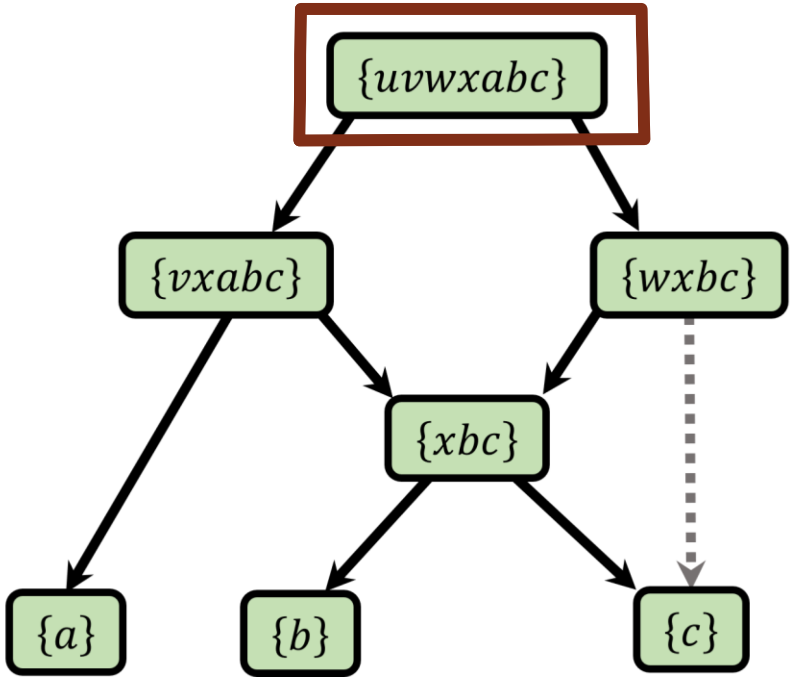


D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) := \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u)$

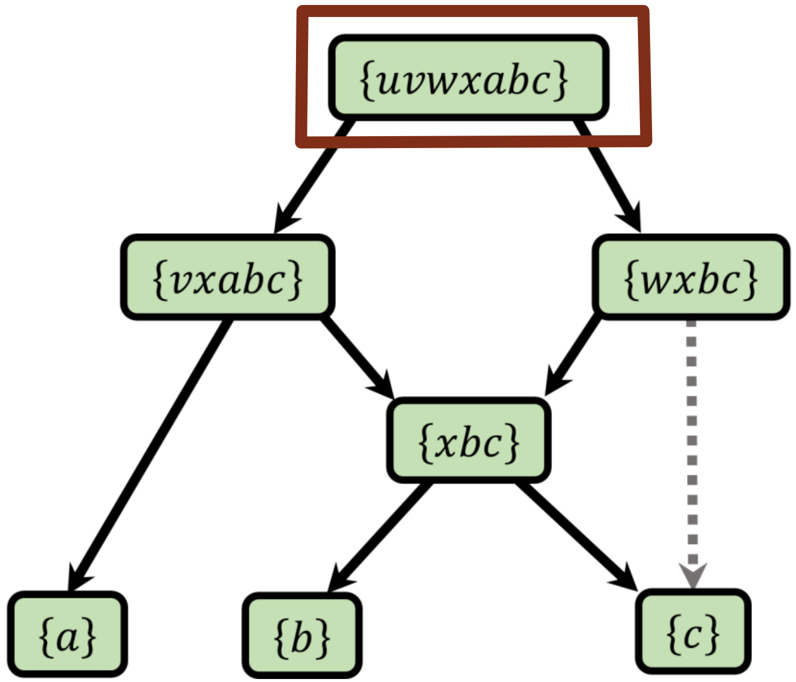


D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

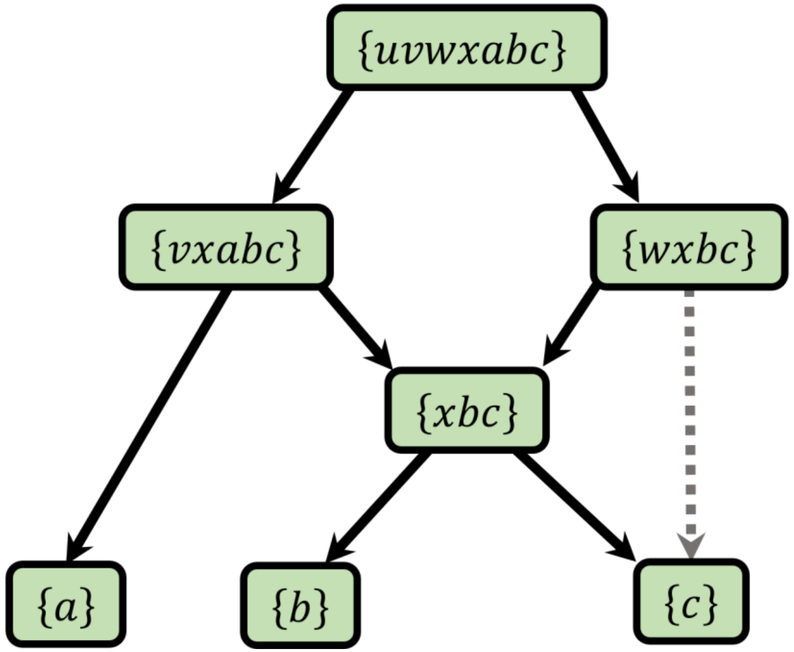


D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\widetilde{\boldsymbol{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{U}$
  → $\widetilde{\boldsymbol{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$



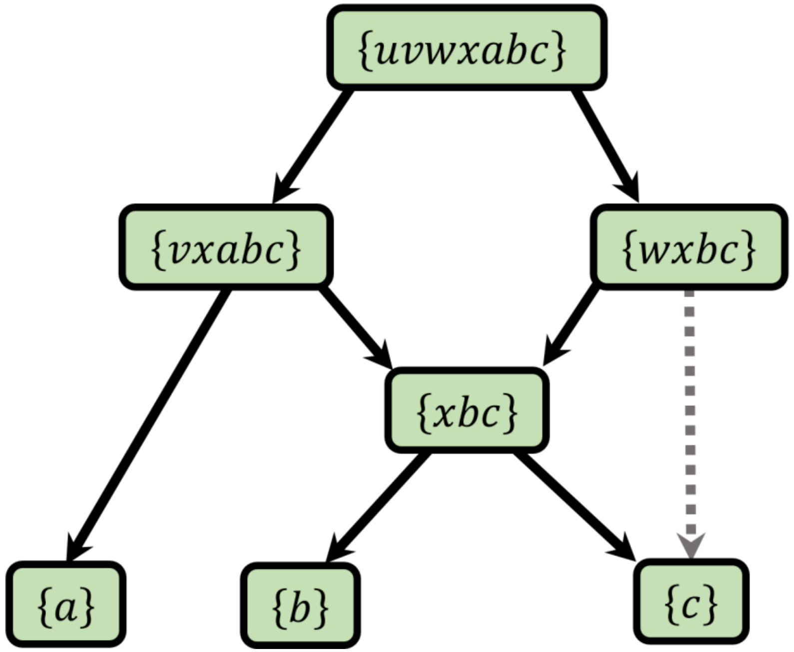
D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v)$



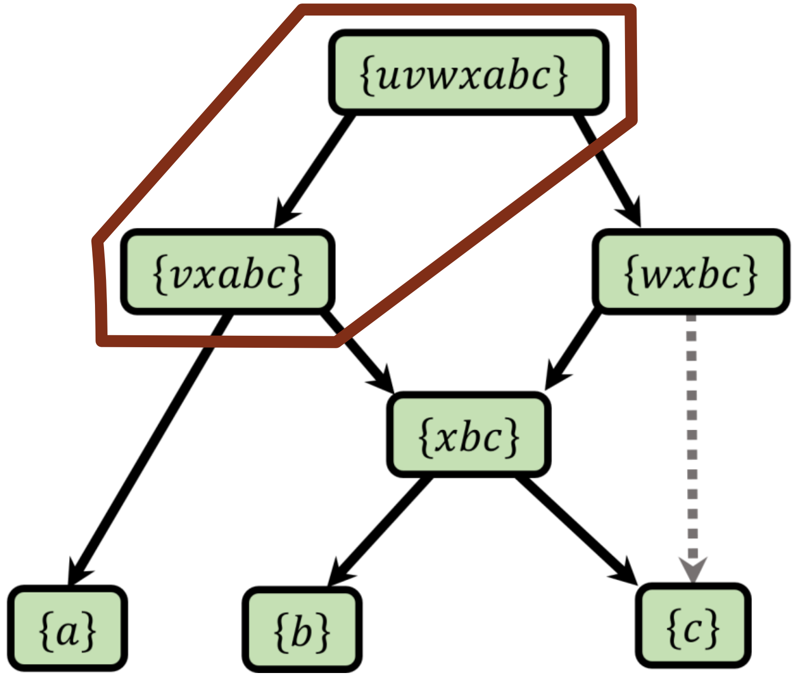
D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{U}$
  → $\boldsymbol{\widetilde{D}(u)} \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v)$



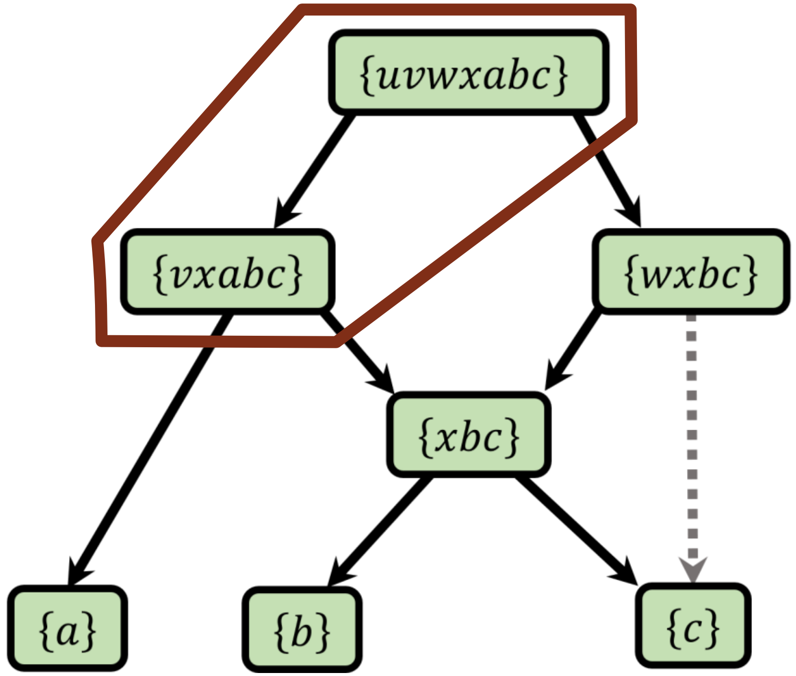
D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$



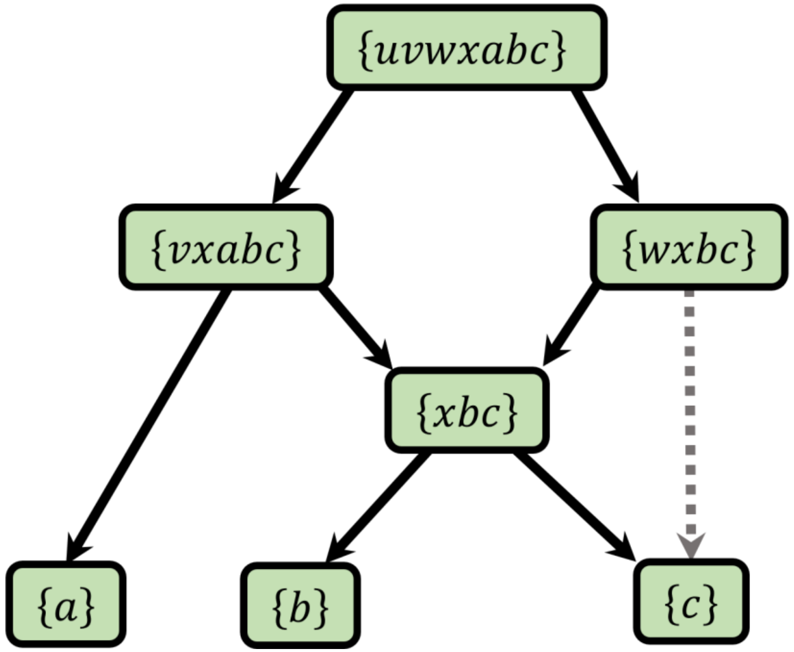
D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{U}$
  → $\boldsymbol{\widetilde{D}(u) := \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$



D-Snake! $\widetilde{D}$

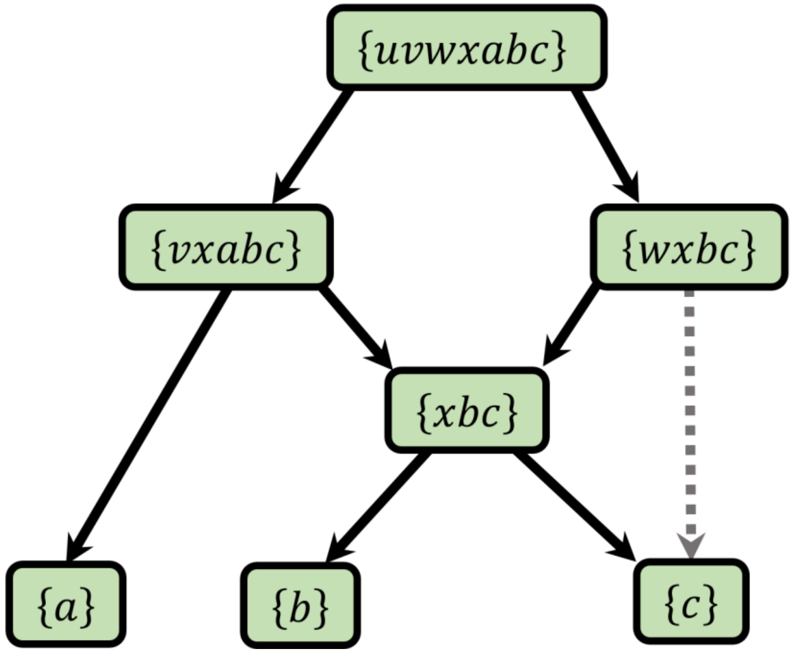
- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) := \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w)$



D-Snake! $\widetilde{D}$

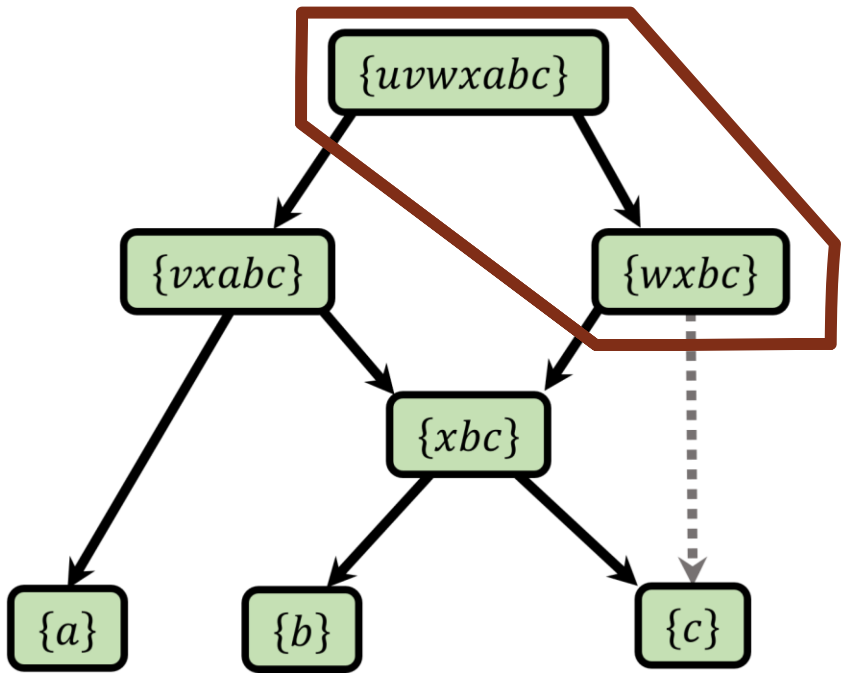
- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{U}$
  → $\boldsymbol{\widetilde{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w)$



D-Snake! $\widetilde{D}$

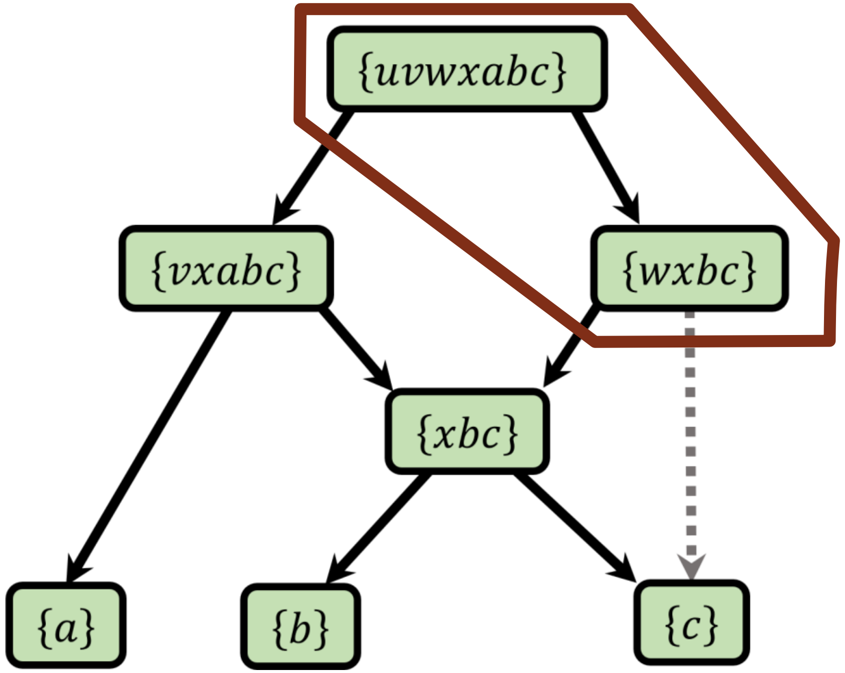
- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) := \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$



D-Snake! $\widetilde{D}$

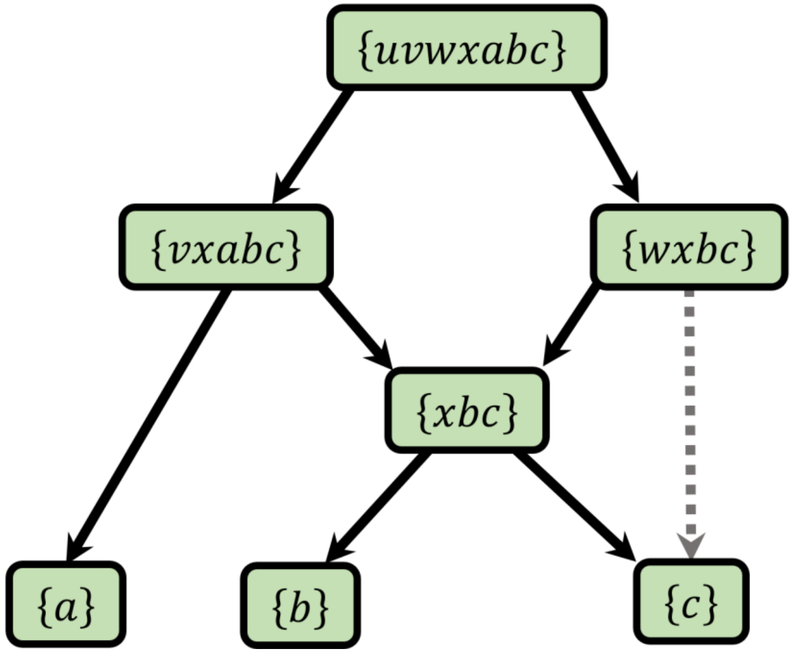
- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u)} \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) := \mathrm{U}}$

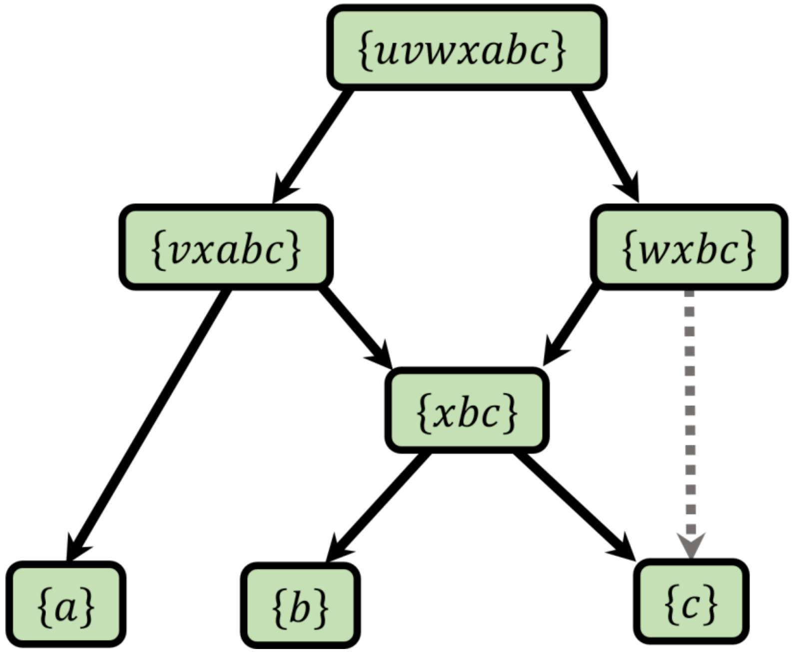
- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x)$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u)} \coloneqq \mathbf{U}$

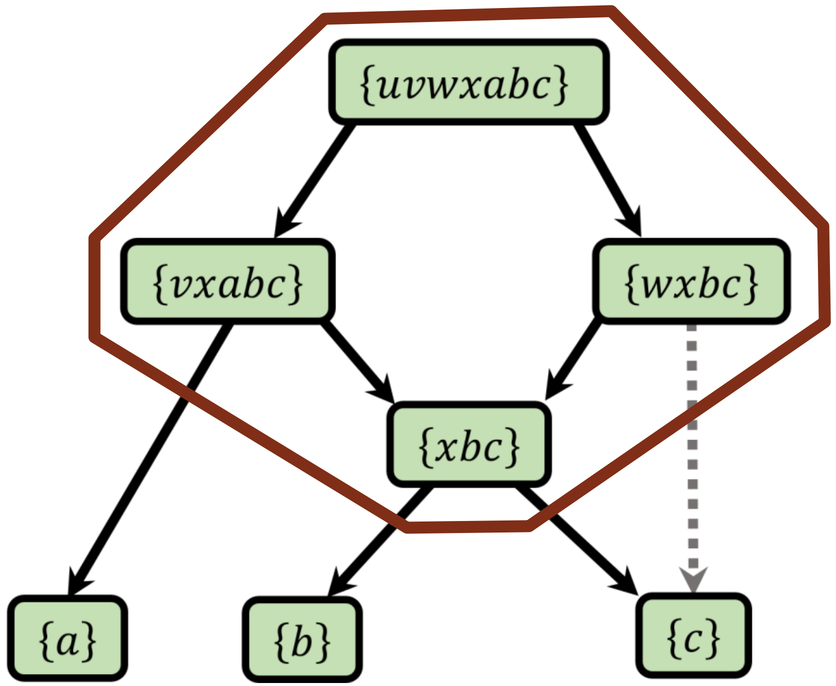
- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x)$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\widetilde{\boldsymbol{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U}'$ **for all** $\boldsymbol{U}' \in \boldsymbol{U}$
  → $\widetilde{\boldsymbol{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

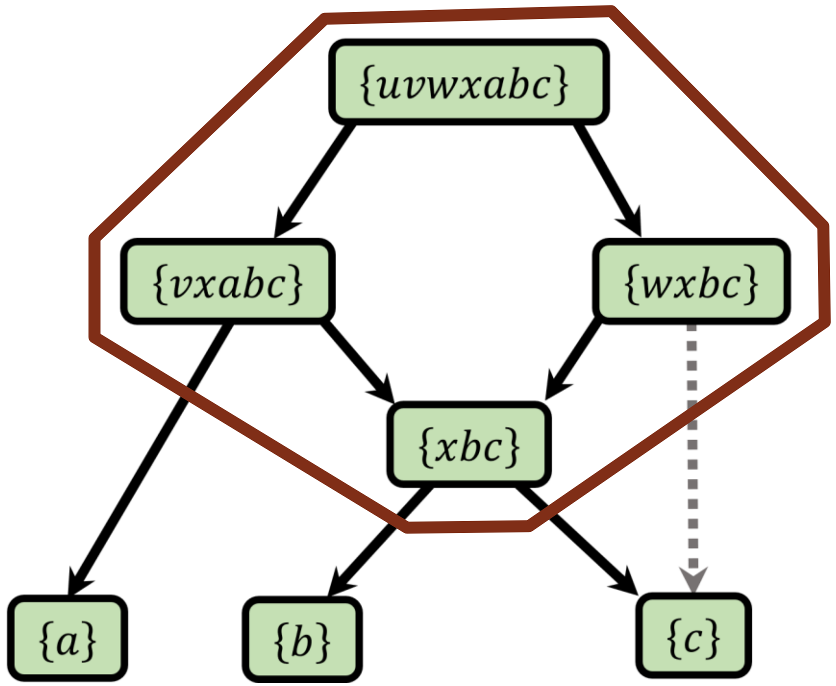
- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\widetilde{\boldsymbol{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U}'$ **for all** $\boldsymbol{U}' \in \boldsymbol{U}$
  → $\widetilde{\boldsymbol{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

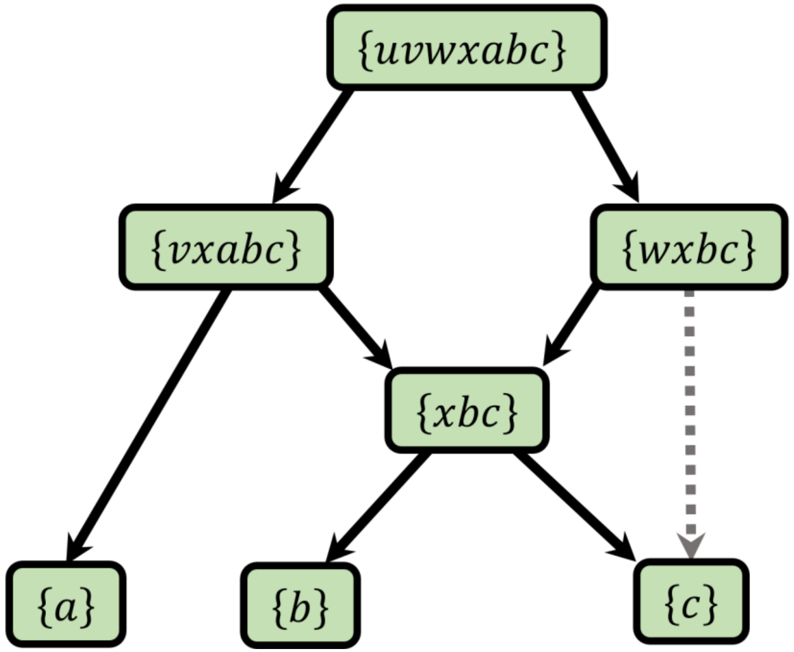
- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{U}$
  → $\boldsymbol{\widetilde{D}(u)} \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

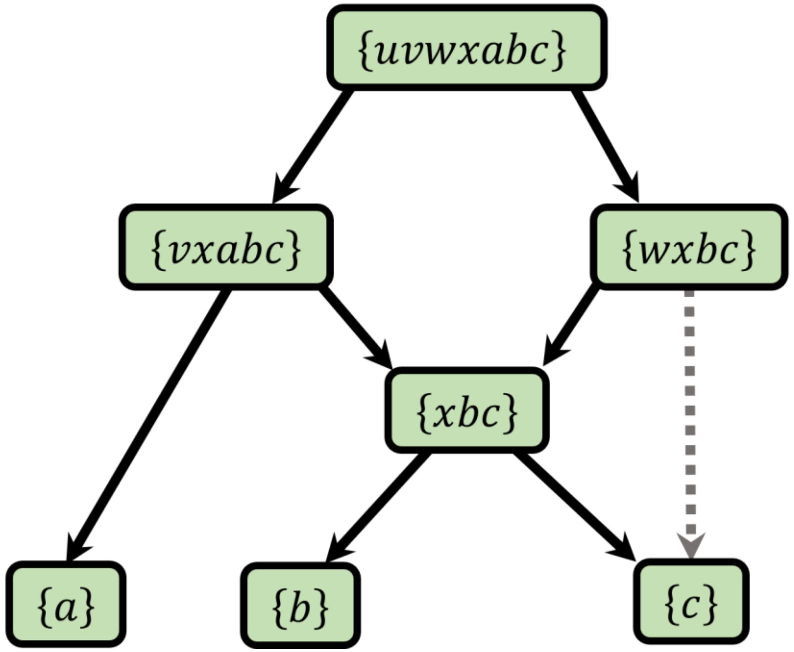
$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a)$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathbf{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

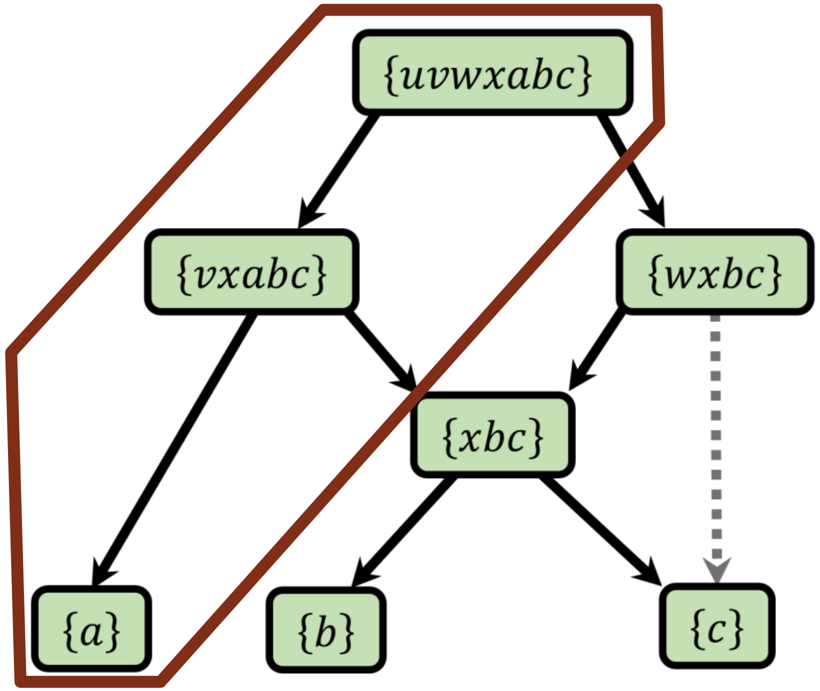
$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a)$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

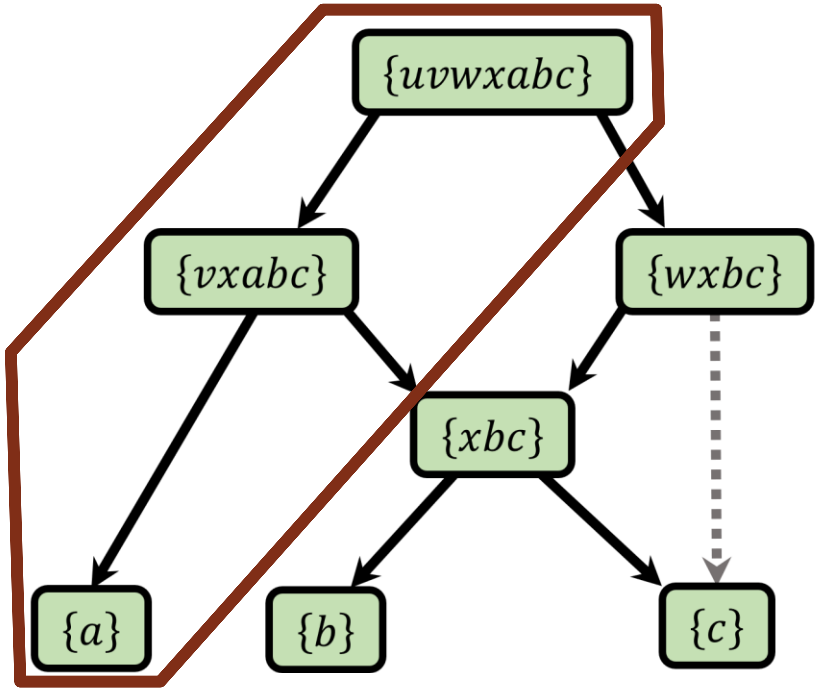
$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathbf{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

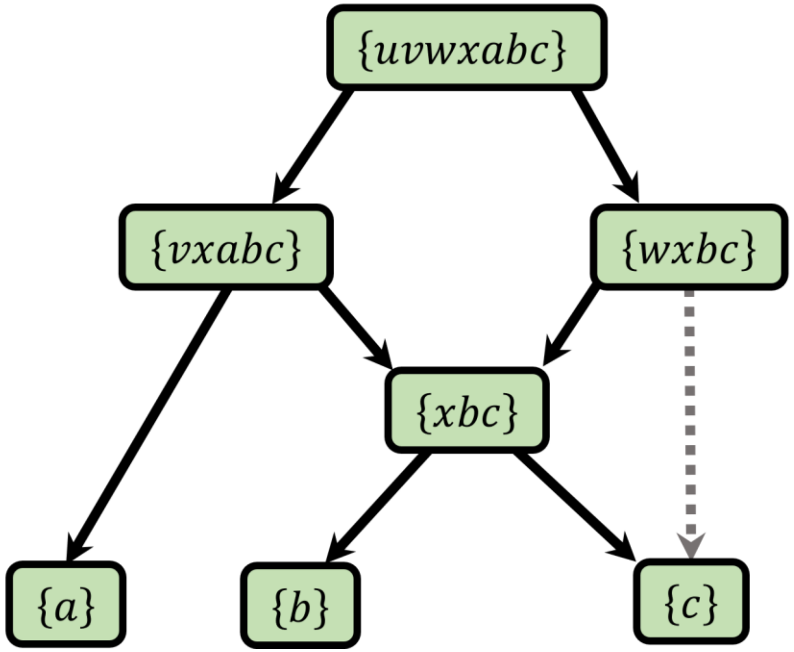
$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

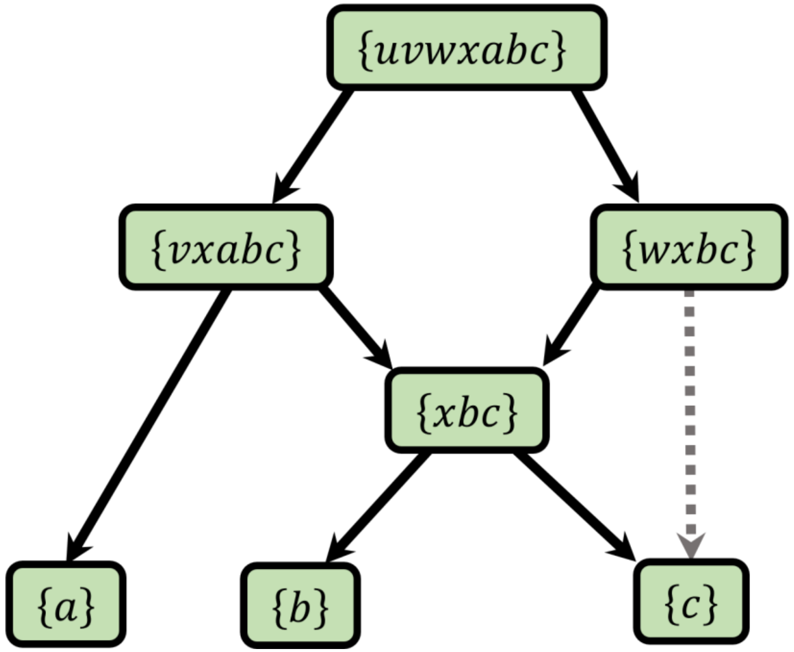
$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b)$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

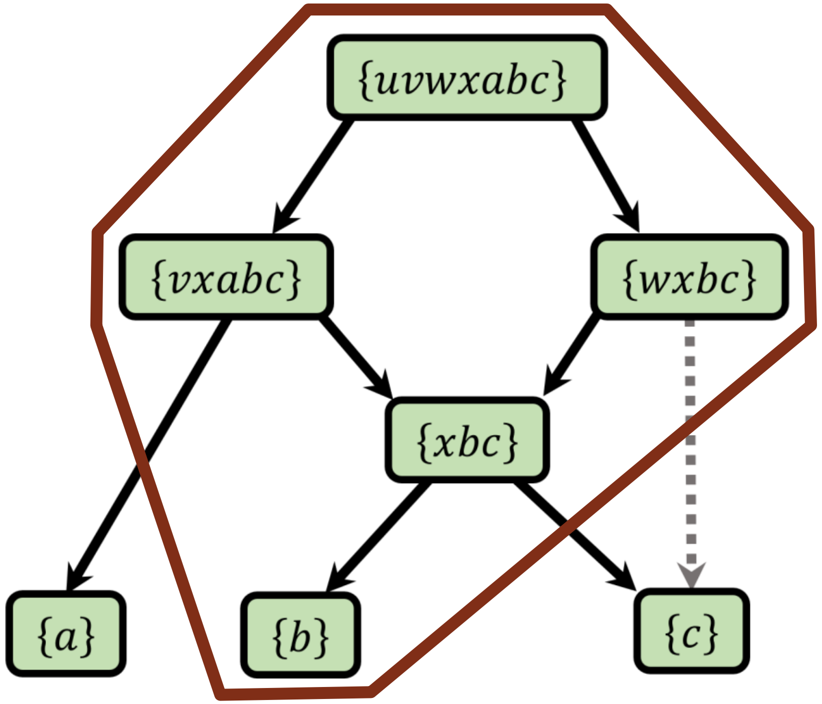
$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b)$



D-Snake! $\widetilde{D}$

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U} = \{A \in \mathfrak{S} \mid u \in A\}}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in U}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

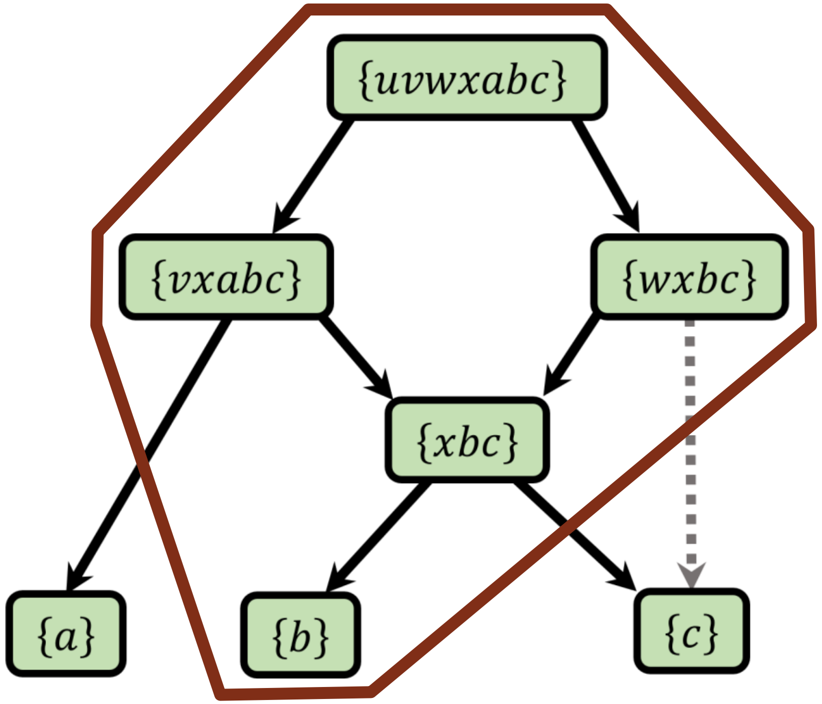
$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b) = \{b\}$



D-Snake!

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U \subseteq U'}$ **for all** $\boldsymbol{U' \in \widetilde{U}}$
  → $\boldsymbol{\widetilde{D}(u) \coloneqq \mathrm{U}}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

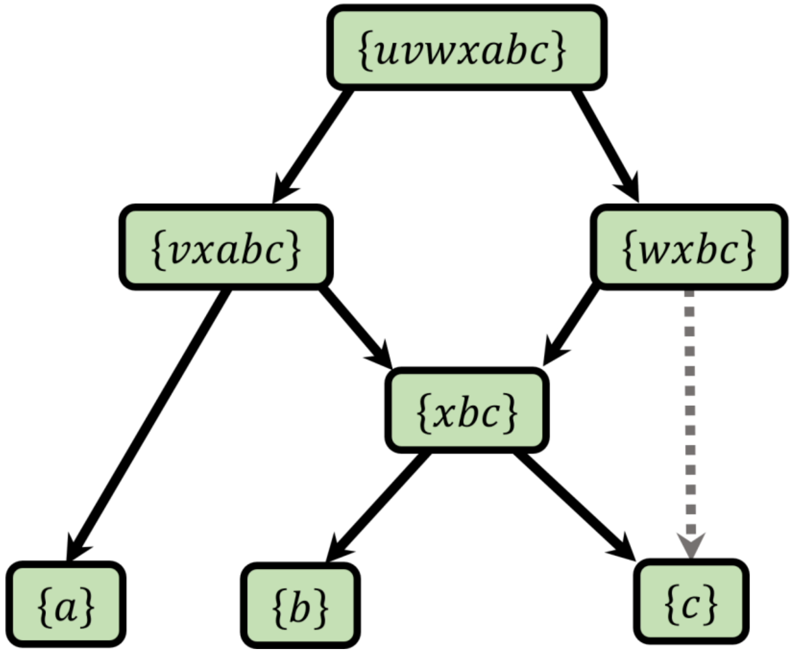
$\widetilde{D}(v) = \{vxabc\}$

$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b) = \{b\}$



D-Snake!

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\boldsymbol{\widetilde{U}} = \{\boldsymbol{A} \in \boldsymbol{\mathfrak{S}} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U'}$ **for all** $\boldsymbol{U'} \in \boldsymbol{\widetilde{U}}$
  → $\boldsymbol{\widetilde{D}(u)} \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

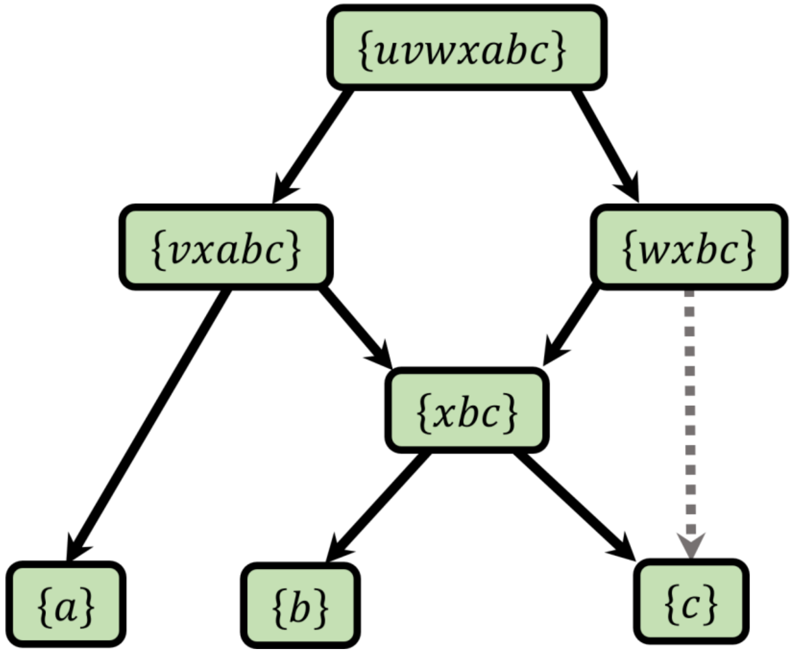
$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b) = \{b\}$

$\widetilde{D}(c)$



D-Snake!

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\widetilde{\boldsymbol{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U}'$ **for all** $\boldsymbol{U}' \in \widetilde{\boldsymbol{U}}$
  → $\widetilde{\boldsymbol{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

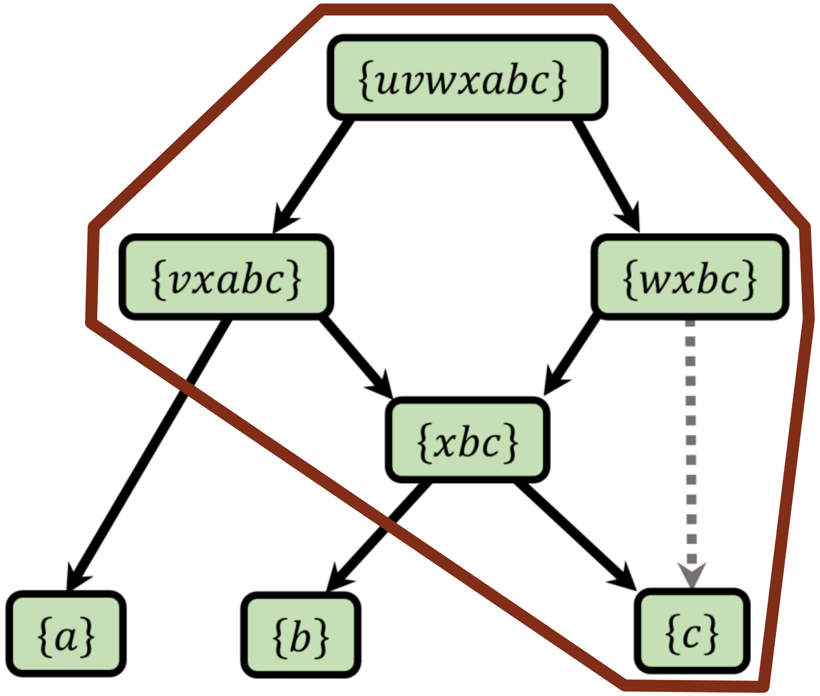
$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b) = \{b\}$

$\widetilde{D}(c)$



D-Snake!

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\widetilde{\boldsymbol{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U}'$ **for all** $\boldsymbol{U}' \in \widetilde{\boldsymbol{U}}$
  → $\widetilde{\boldsymbol{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

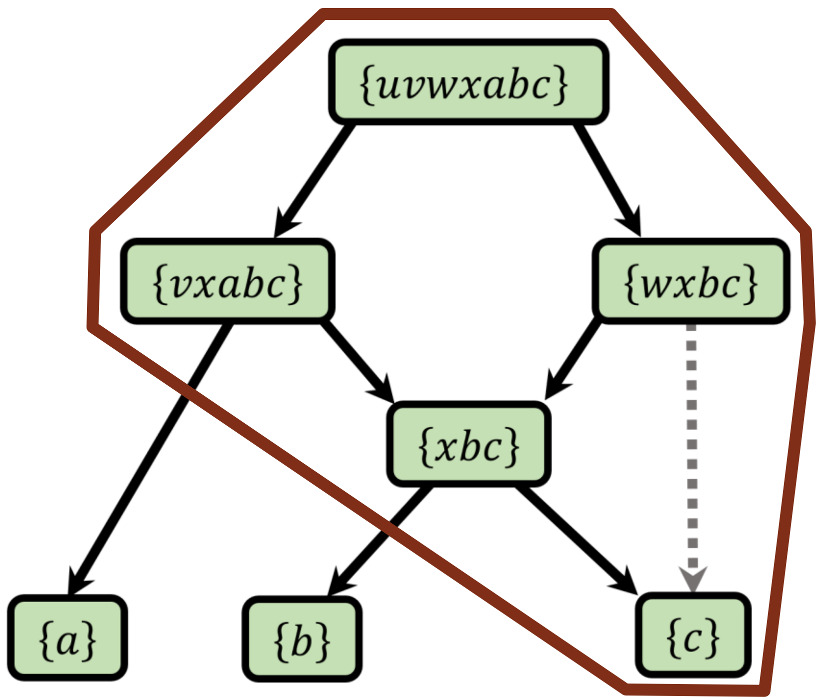
$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b) = \{b\}$

$\widetilde{D}(c) = \{c\}$



D-Snake!

- For set system $\mathfrak{S}$ over X and $u \in X$:
  → derive $\widetilde{\boldsymbol{U}} = \{\boldsymbol{A} \in \mathfrak{S} \mid \boldsymbol{u} \in \boldsymbol{A}\}$
  (all sets in $\mathfrak{S}$ that contain u)
  → find "minimal" $U \in \widetilde{U}$ such that
  $\boldsymbol{U} \subseteq \boldsymbol{U}'$ for all $\boldsymbol{U}' \in \widetilde{\boldsymbol{U}}$
  → $\widetilde{\boldsymbol{D}}(\boldsymbol{u}) \coloneqq \mathbf{U}$

- Corresponds to a **minimal element**, or sink, in the by $\widetilde{U}$ induced subgraph of $\mathcal{H}(\mathfrak{S})$ ($\mathcal{H}(\mathfrak{S})[\widetilde{U}]$)

$\widetilde{D}(u) = \{uvwxabc\}$

$\widetilde{D}(v) = \{vxabc\}$

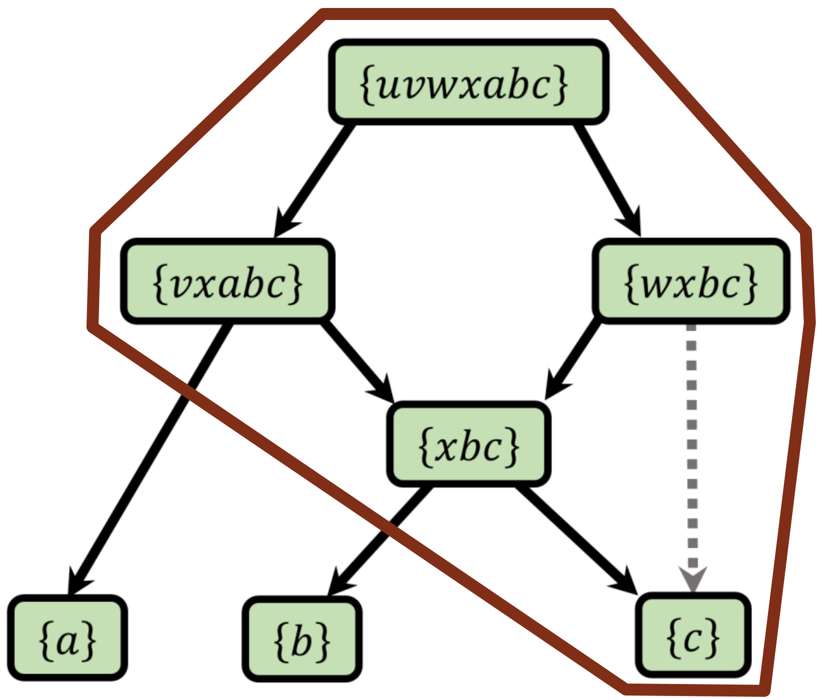
$\widetilde{D}(w) = \{wxbc\}$

$\widetilde{D}(x) = \{xbc\}$

$\widetilde{D}(a) = \{a\}$

$\widetilde{D}(b) = \{b\}$

$\widetilde{D}(c) = \{c\}$



Let $\boldsymbol{G}$ be a DAG and $\mathfrak{D}$ be its descendant cluster. Then, $\boldsymbol{D}(\boldsymbol{u}) = \widetilde{\boldsymbol{D}}(\boldsymbol{u})$ and $\boldsymbol{f}: \boldsymbol{V}(\boldsymbol{G}) \to \mathfrak{D}, \boldsymbol{f}(\boldsymbol{u}) \coloneqq \widetilde{\boldsymbol{D}}(\boldsymbol{u})$ is bijective.
(Has also been proved with non-anecdotal arguments)

D-Snake!

D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

D-Snake!

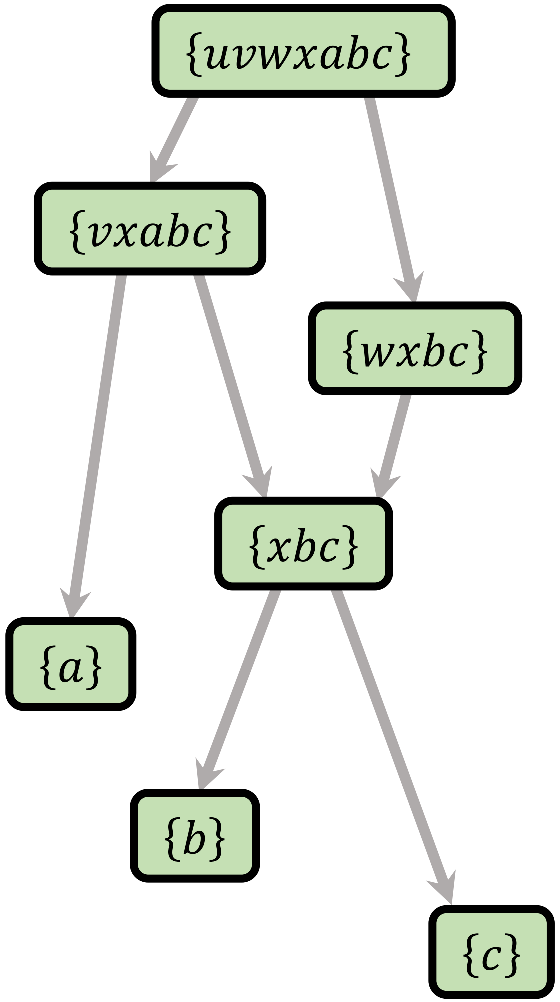
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

(iii) $x \in \widetilde{D}(v)$ **if and only if** $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) := \widetilde{D}(u)$ is a bijection.

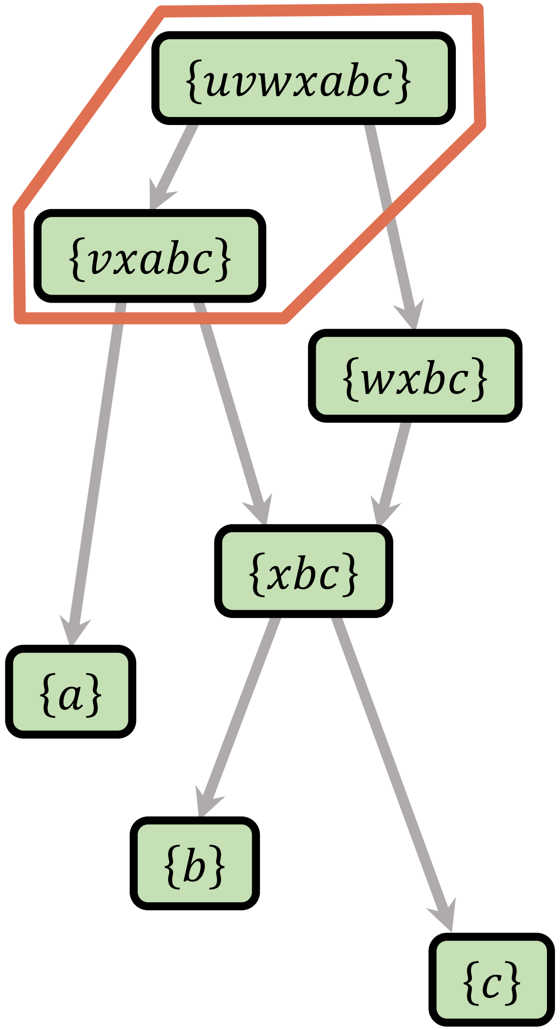
(iii) $x \in \widetilde{D}(v)$ if and only if $\widetilde{D}(x) \subseteq \widetilde{D}(v)$



D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) := \widetilde{D}(u)$ is a bijection.

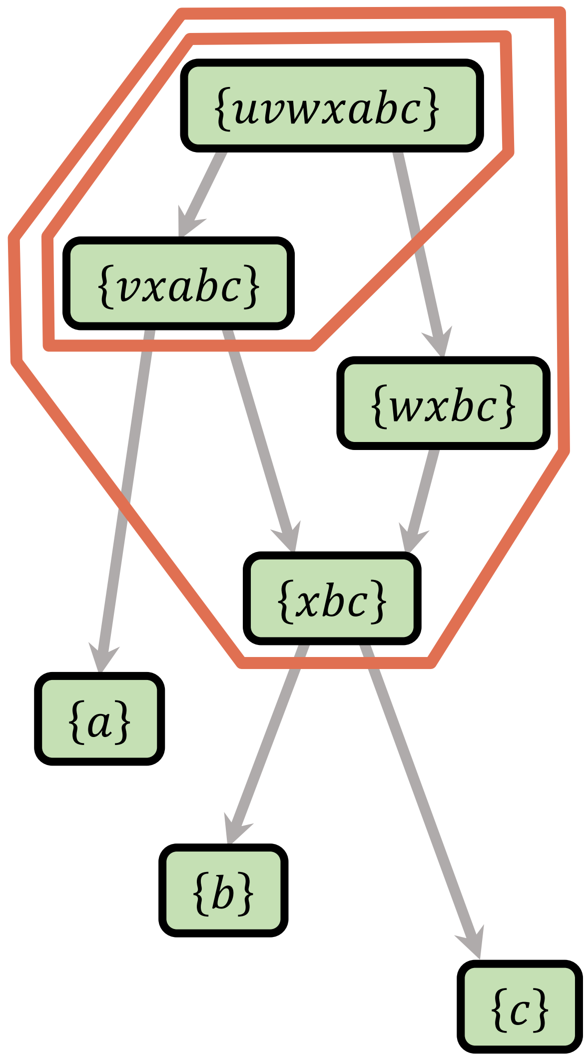
(iii) $x \in \widetilde{D}(v)$ if and only if $\widetilde{D}(x) \subseteq \widetilde{D}(v)$



D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) := \widetilde{D}(u)$ is a bijection.

(iii) $x \in \widetilde{D}(v)$ if and only if $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

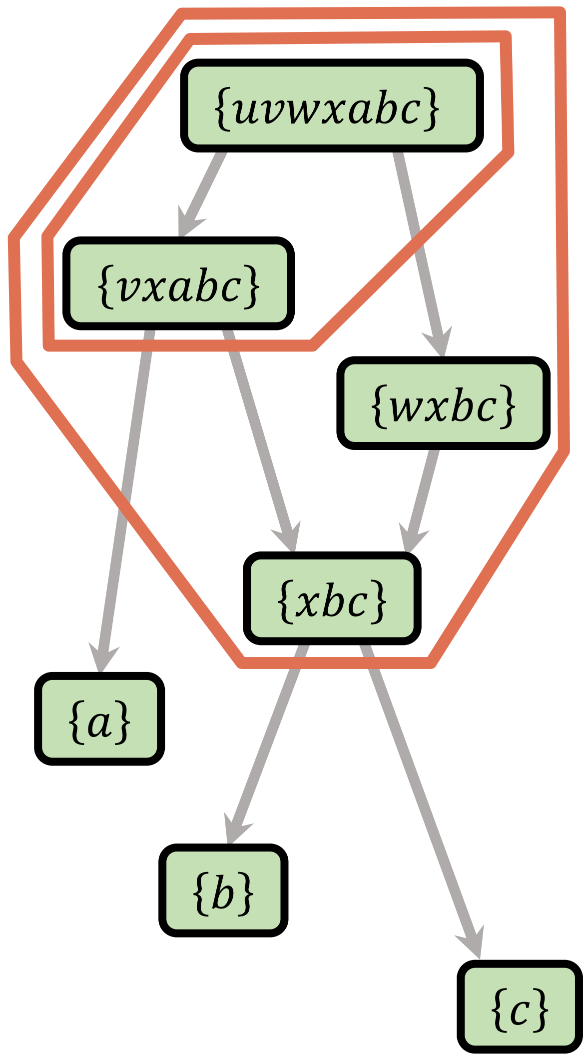


D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x} \in \widetilde{\boldsymbol{D}}(\boldsymbol{v})$ **if and only if** $\widetilde{\boldsymbol{D}}(\boldsymbol{x}) \subseteq \widetilde{\boldsymbol{D}}(\boldsymbol{v})$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal
for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$



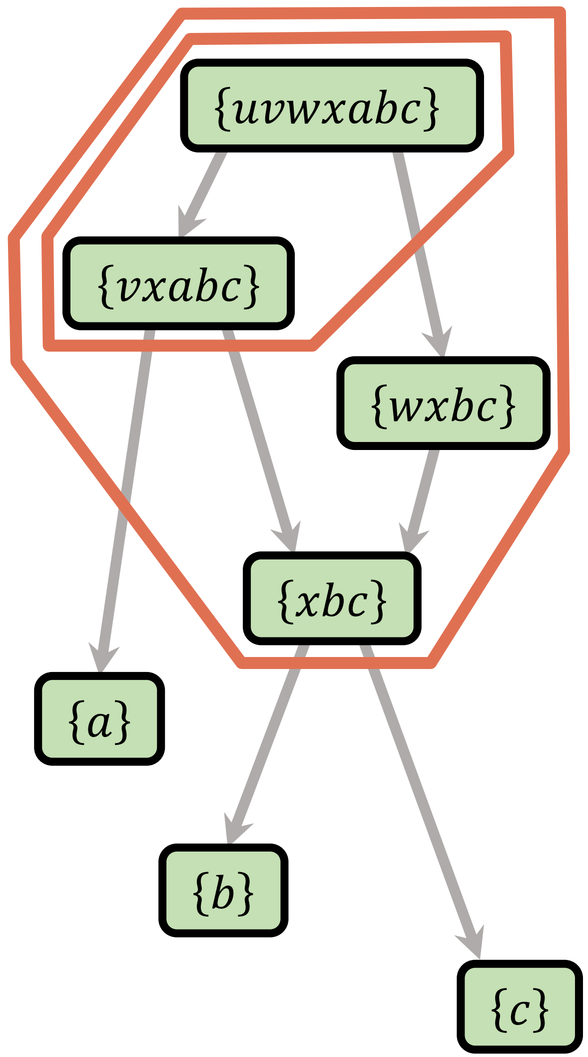
D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x} \in \widetilde{\boldsymbol{D}}(\boldsymbol{v})$ **if and only if** $\widetilde{\boldsymbol{D}}(\boldsymbol{x}) \subseteq \widetilde{\boldsymbol{D}}(\boldsymbol{v})$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal
for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→$\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition,
consequently $v \in \widetilde{D}(u)$



D-Snake!

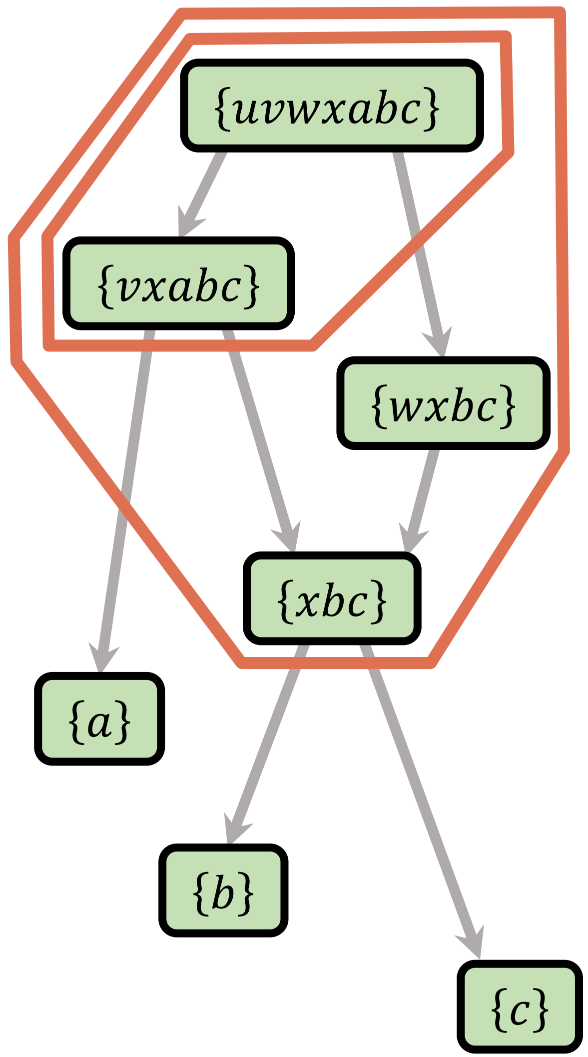
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x} \in \widetilde{\boldsymbol{D}}(\boldsymbol{v})$ **if and only if** $\widetilde{\boldsymbol{D}}(\boldsymbol{x}) \subseteq \widetilde{\boldsymbol{D}}(\boldsymbol{v})$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be $\subseteq$-minimal
for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→$\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition,
consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters



D-Snake!

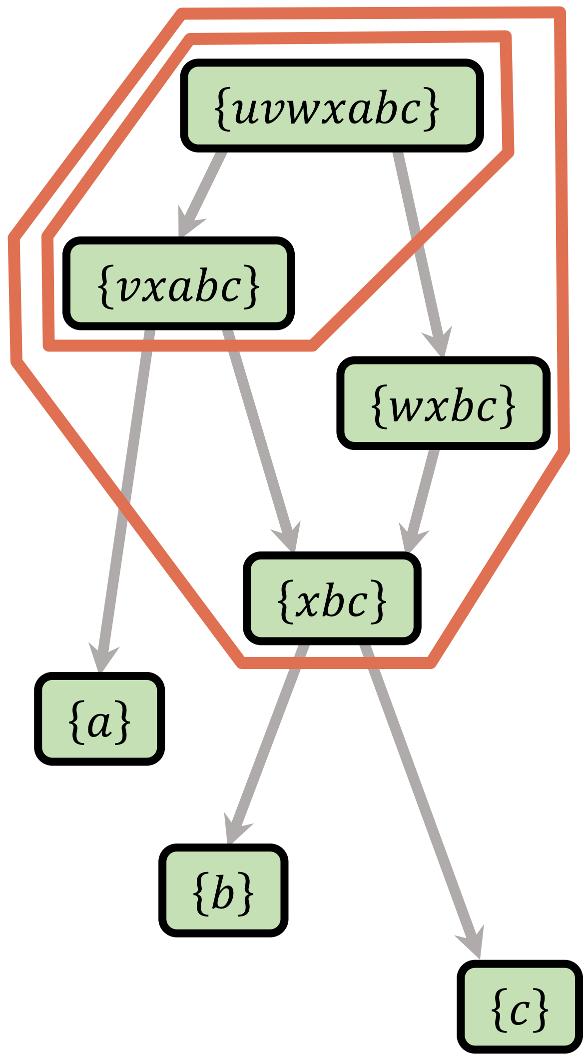
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) := \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→ $x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be $\subseteq$-minimal for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→ $\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition, consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.



D-Snake!

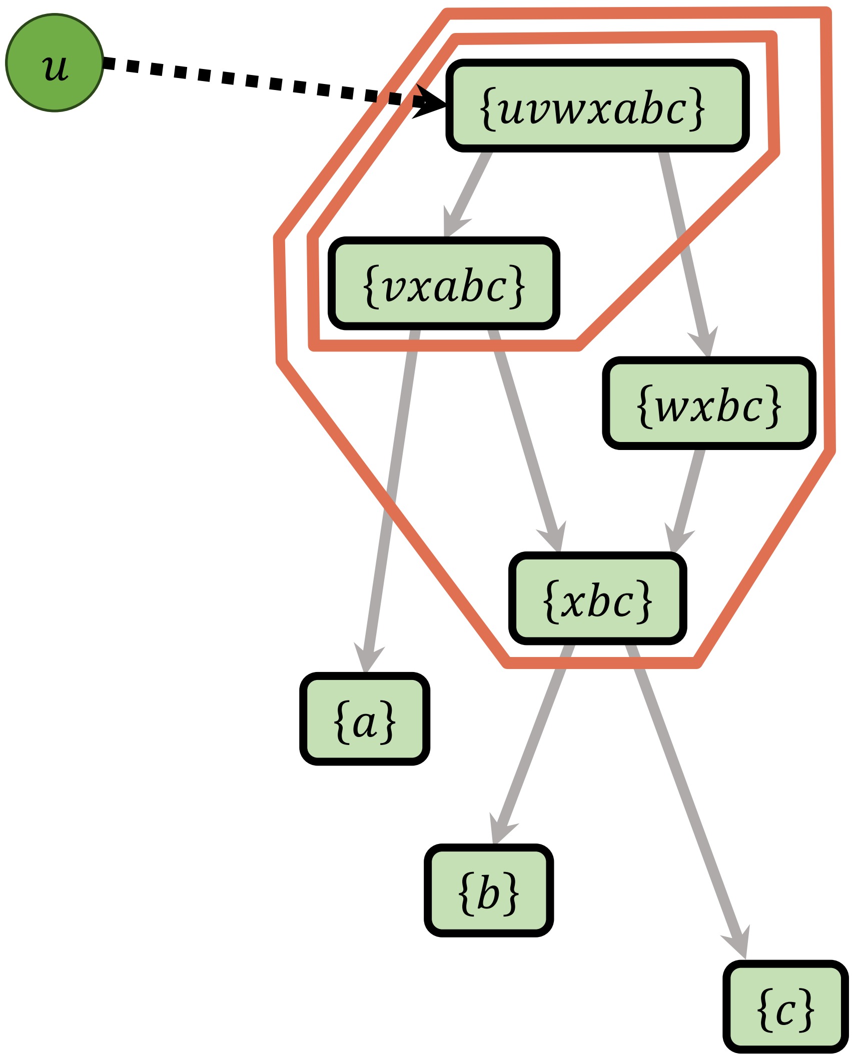
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→$\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition, consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.



D-Snake!

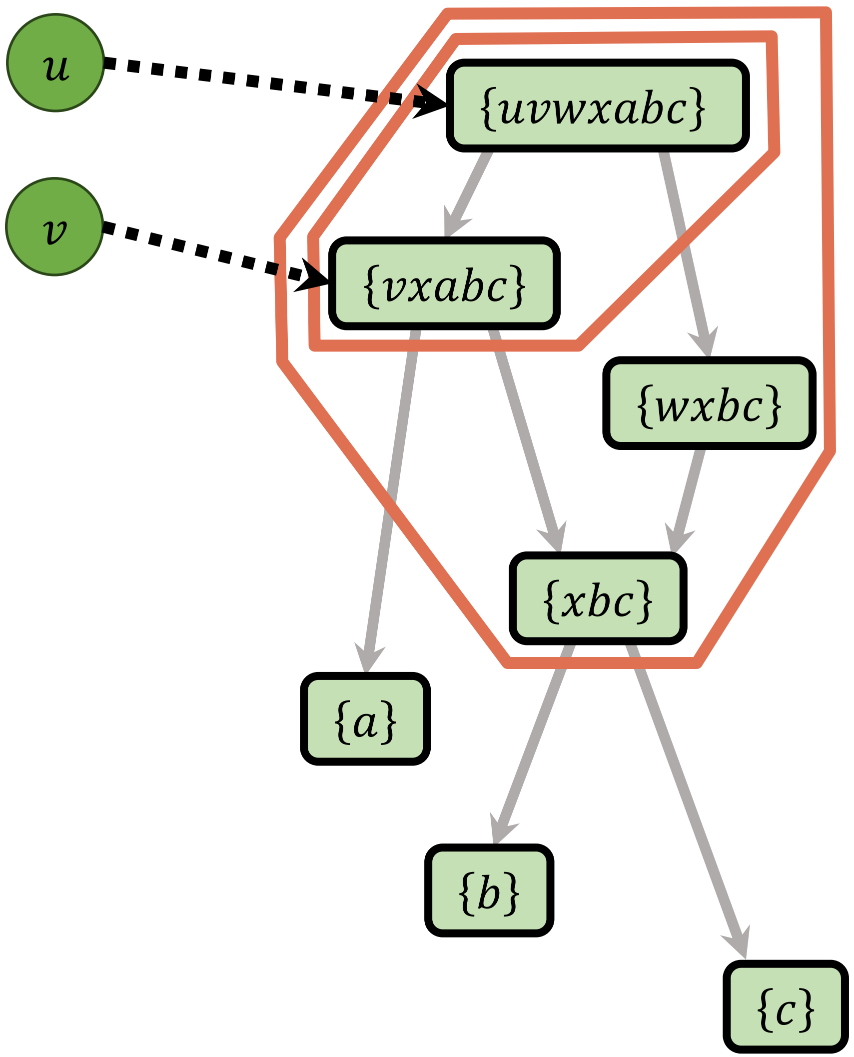
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal
for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→$\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition,
consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.



D-Snake!

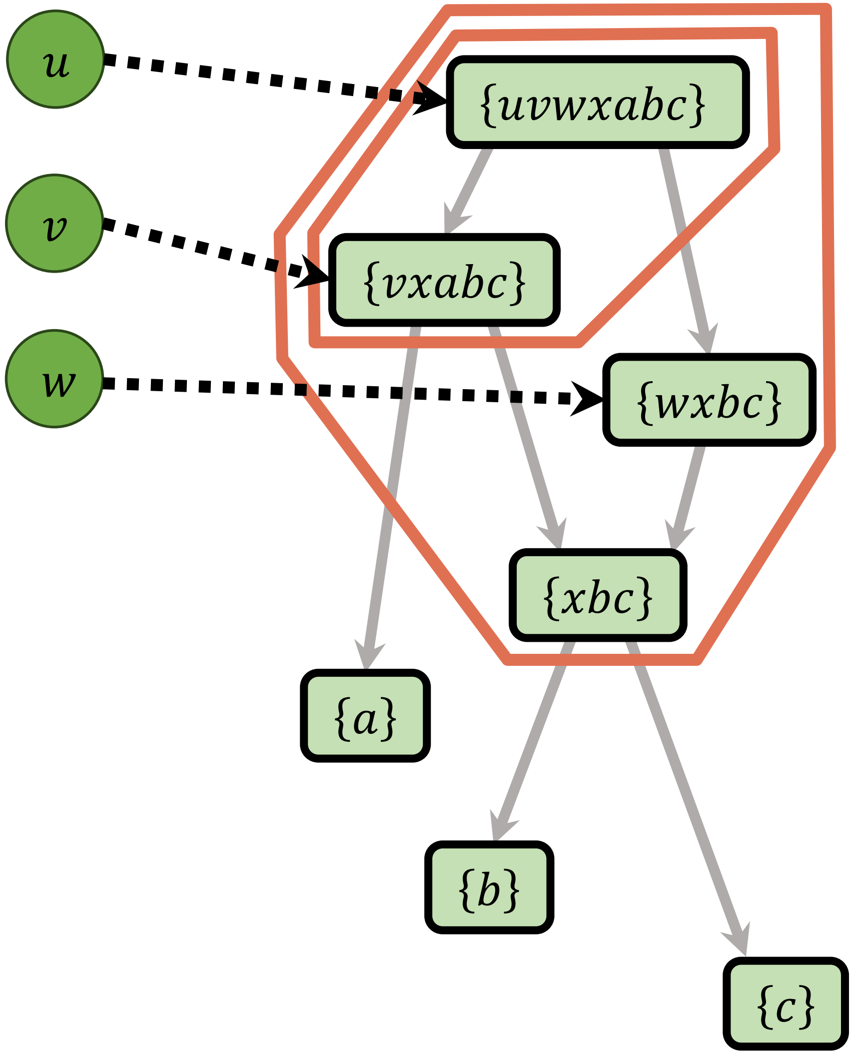
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→$\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition, consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.



D-Snake!

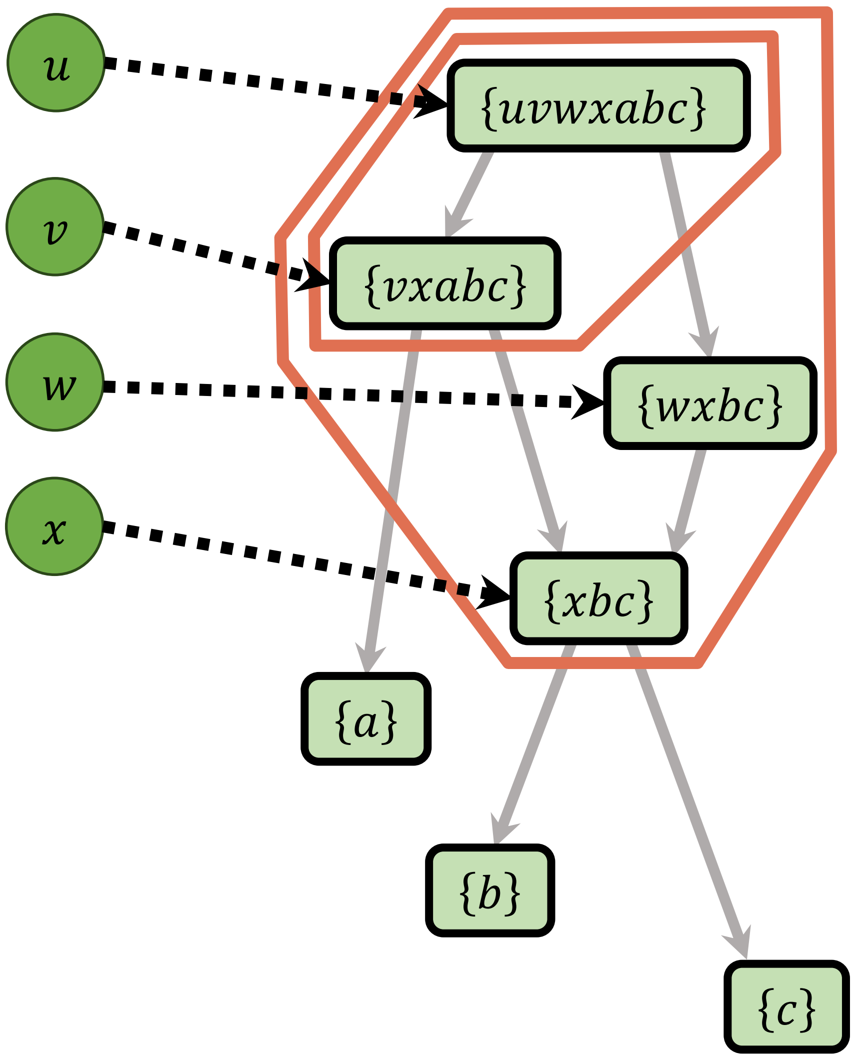
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→ $x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be $\subseteq$-minimal for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→ $\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition, consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.



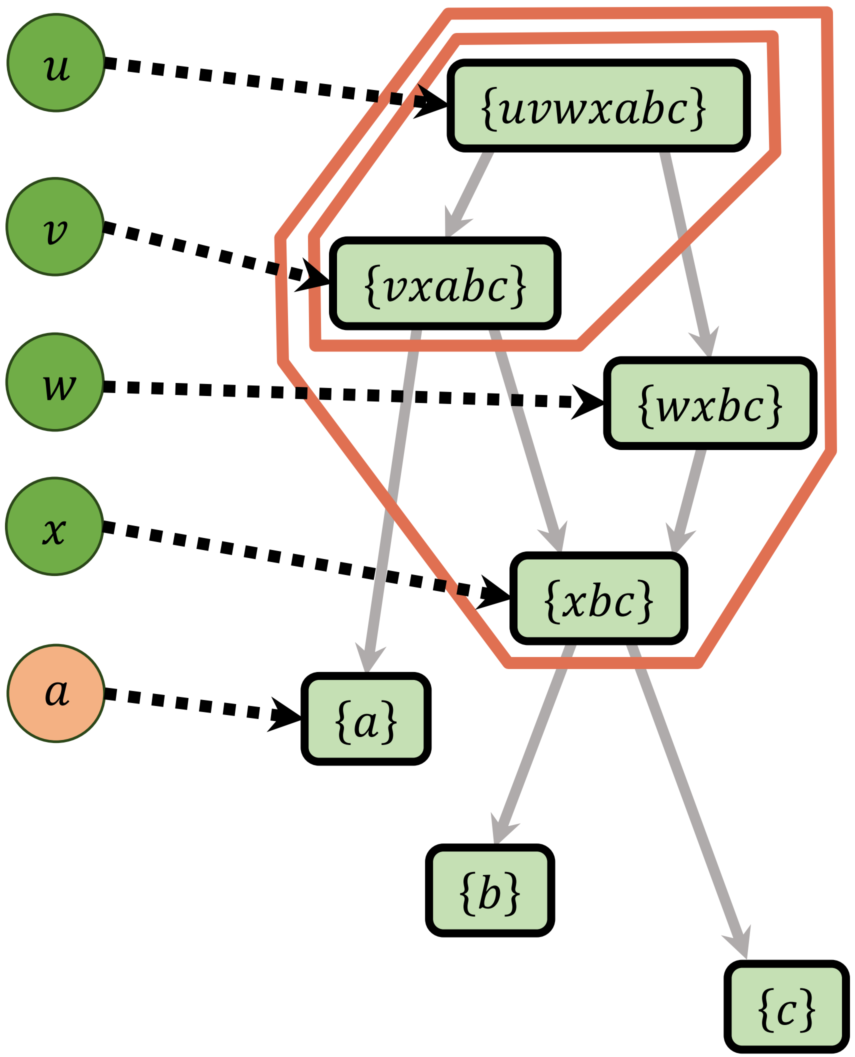
D-Snake!

(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→ $x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→ $\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition, consequently $v \in \widetilde{D}(u)$



- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.

D-Snake!

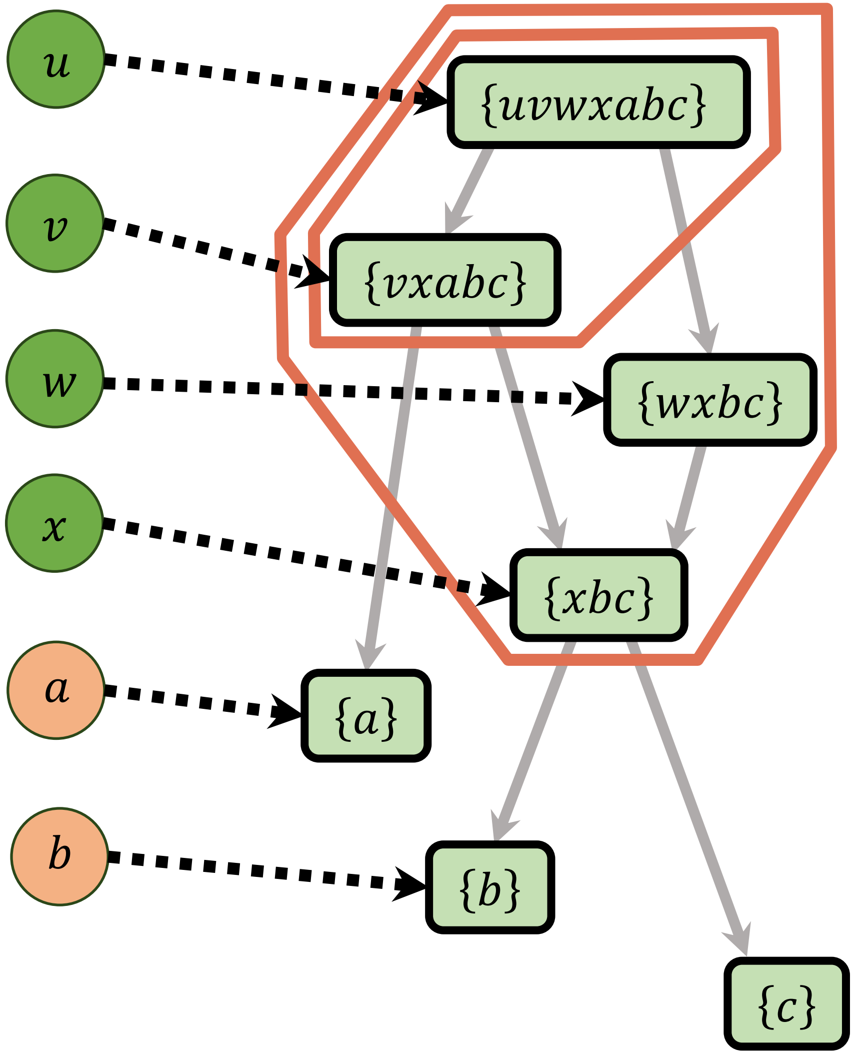
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) \coloneqq \widetilde{D}(u)$ is a bijection.

$(\boldsymbol{iii})$ $\boldsymbol{x \in \widetilde{D}(v)}$ **if and only if** $\boldsymbol{\widetilde{D}(x) \subseteq \widetilde{D}(v)}$

→$x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be ⊆-minimal
for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→$\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition,
consequently $v \in \widetilde{D}(u)$

- $(\boldsymbol{iii})$ gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *"missing"* any elements.



D-Snake!

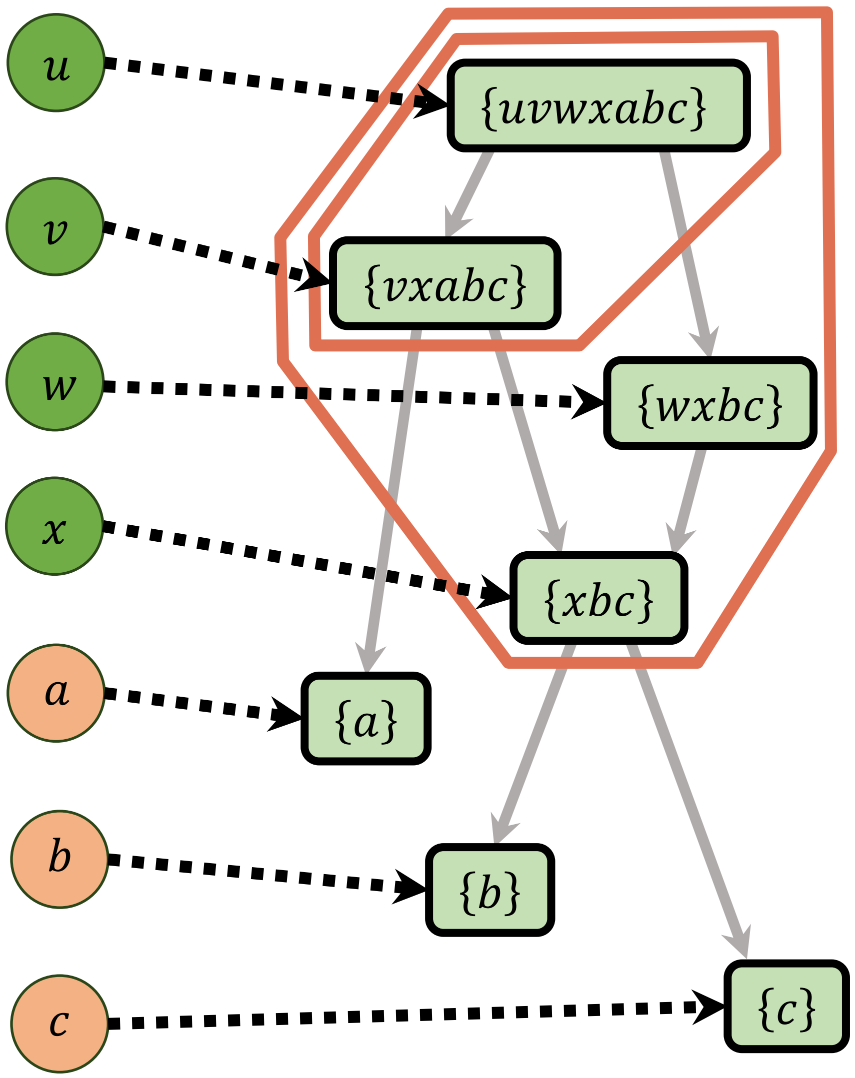
(THM) Let $\mathfrak{S}$ be a set system over X. Then, $\mathfrak{S} = \mathfrak{D}$ of a directed graph G if and only if $f: X \to \mathfrak{S}, f(u) := \widetilde{D}(u)$ is a bijection.

(iii) $x \in \widetilde{D}(v)$ **if and only if** $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→ $x \in \widetilde{D}(v)$. $\widetilde{D}(x)$ has to be $\subseteq$-minimal for all $U \in \mathfrak{S}$ with x $\in U$ hence $\widetilde{D}(x) \subseteq \widetilde{D}(v)$

→ $\widetilde{D}(x) \subseteq \widetilde{D}(v)$. $x \in \widetilde{D}(x)$ and $v \in \widetilde{D}(v)$ by definition, consequently $v \in \widetilde{D}(u)$

- (iii) gives us the general properties of descendant clusters
- Bijectivity of f ensures we are not *“missing”* any elements.



Outlook

Outlook

- Can we somewhat robustly derive DAGs for *"incomplete"* $\mathfrak{D}$?
  → Might only described by constrains like $lca(x, y) \preccurlyeq lca(x, y, z)$, …

Outlook

- Can we somewhat robustly derive DAGs for *“incomplete”* $\mathfrak{D}$?
  → Might only described by constrains like $lca(x, y) \preccurlyeq lca(x, y, z)$, …

- Characterization of $\mathfrak{D}$ of DAG graph classes with certain global properties
  → e.g. global last-common-ancestor networks

Outlook

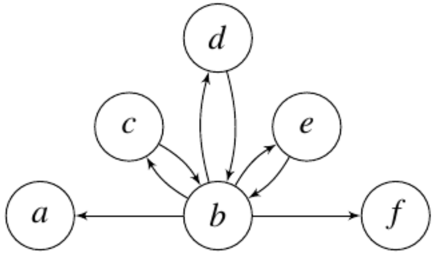
- Can we somewhat robustly derive DAGs for *"incomplete"* $\mathfrak{D}$?
  → Might only described by constrains like $lca(x, y) \preccurlyeq lca(x, y, z)$, ...
- Characterization of $\mathfrak{D}$ of DAG graph classes with certain global properties
  → e.g. global last-common-ancestor networks
- How does $\mathfrak{D}$ behave for undirected graphs?
  → strongly connected components can most likely only be recovered as blocks (cliques)

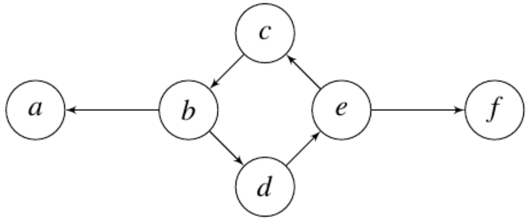
Outlook

- Can we somewhat robustly derive DAGs for *“incomplete”* $\mathfrak{D}$?
  → Might only described by constrains like $lca(x, y) \preccurlyeq lca(x, y, z)$, …

- Characterization of $\mathfrak{D}$ of DAG graph classes with certain global properties
  → e.g. global last-common-ancestor networks

- How does $\mathfrak{D}$ behave for undirected graphs?
  → strongly connected components can most likely only be recovered as blocks (cliques)

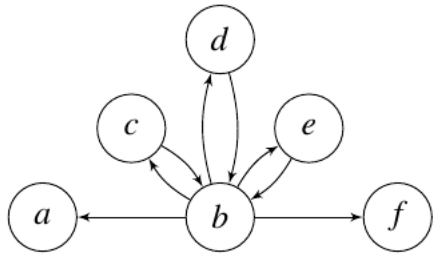




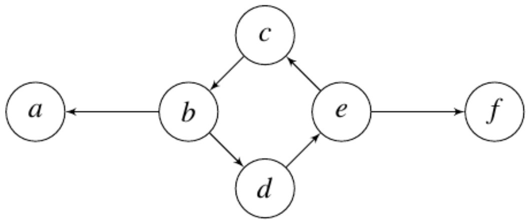
Outlook

- Can we somewhat robustly derive DAGs for *"incomplete"* $\mathfrak{D}$?
  → Might only described by constrains like $lca(x, y) \preccurlyeq lca(x, y, z)$, …

- Characterization of $\mathfrak{D}$ of DAG graph classes with certain global properties
  → e.g. global last-common-ancestor networks



- How does $\mathfrak{D}$ behave for undirected graphs?
  → strongly connected components can most likely only be recovered as blocks (cliques)



- Somewhat coincides with the all-path transit function for DAGs:
  $A(u, v) = D(u) \cap P(v)$ (where P denotes all predecessors of v)

Thank you!

Marc Hellmuth

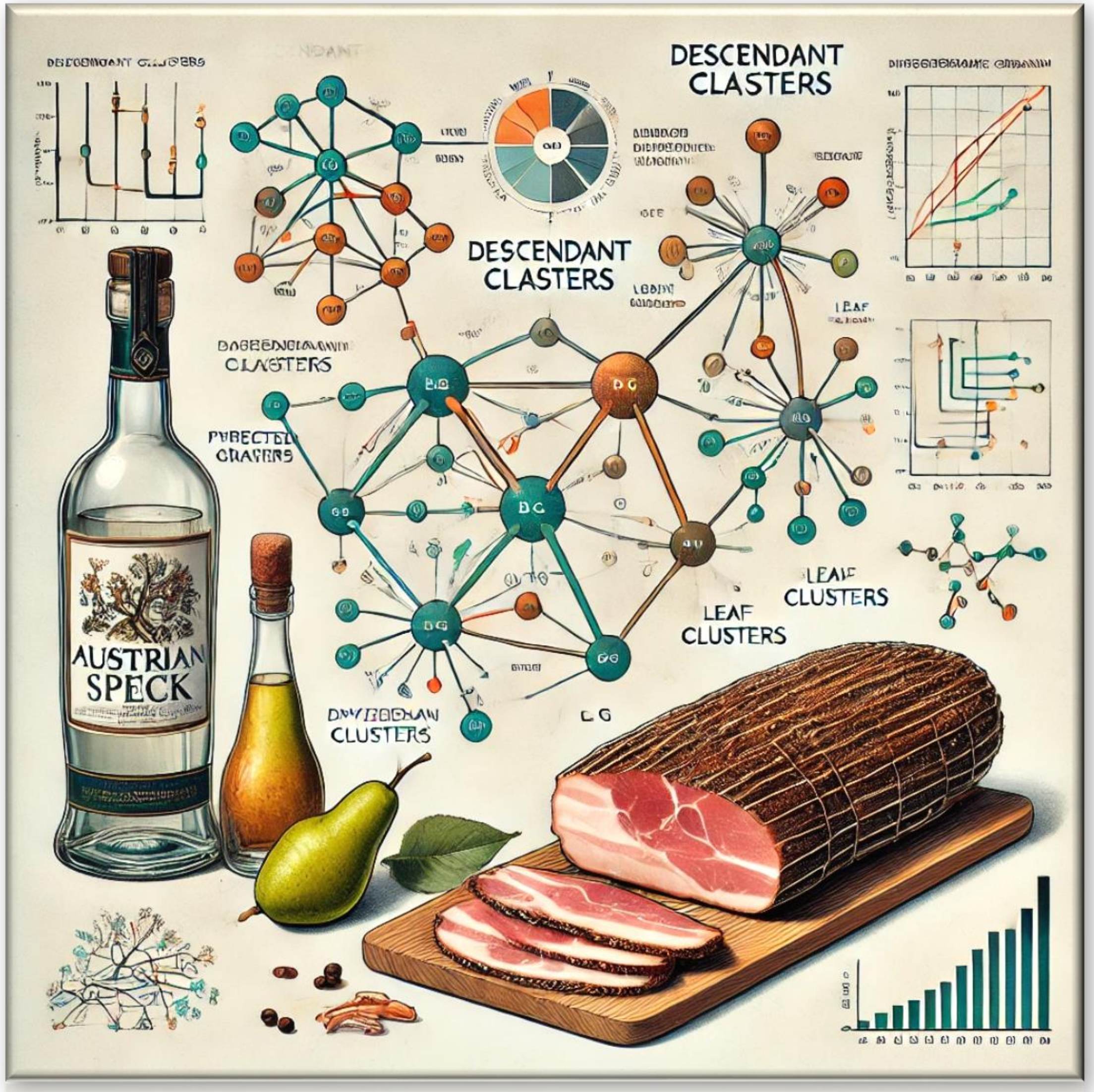
Peter F. Stadler

&

Everyone attending!

“Here is your **corrected graphical abstract**, now with **Austrian Speck accurately depicted as a large piece of cured bacon** on the right side and the **bottle of clear pear liquor on the left**. The scientific focus remains intact while subtly and carefully integrating the conference setting.”

~ChatGPT & Dall-E



Leaf extended DAG