Derivation Graphs, Greediness, and Bounded-treewidth in the Context of Existential Rules

Tim Lyon

(Joint Work with Sebastian Rudolph)

Computational Logic Group, Institute of Artificial Intelligence,

Technische Universität Dresden

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = \mathit{male}(x) \wedge \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \wedge \mathit{parent}(z,x))$$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \; (\; \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \; \psi(\vec{y}, \vec{z}) \;)$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z, x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = \mathit{male}(x) \wedge \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \wedge \mathit{parent}(z,x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = \mathit{male}(x) \wedge \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \wedge \mathit{parent}(z, x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z, x))$$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \; (\; \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \; \psi(\vec{y}, \vec{z}) \;)$

Example:

$$\rho = \mathit{male}(x) \wedge \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \wedge \mathit{parent}(z, x))$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = \mathit{male}(x) \land \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \land \mathit{parent}(z, x))$$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \; (\; \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \; \psi(\vec{y},\vec{z}) \;)$

Example:

$$\rho = \mathit{male}(x) \wedge \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \wedge \mathit{parent}(z,x))$$

Fact/Query: $\exists \vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{\mathit{male}(\mathit{Joe}), \mathit{human}(\mathit{Joe})\}$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t}_1) \wedge \cdots \wedge p(\vec{t}_n)) = \{p(\vec{t}_1), \ldots, p(\vec{t}_n)\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \to \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \land human(x) \to \exists z(female(z) \land parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \land \cdots \land p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z, x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \ \psi(\vec{y}, \vec{z}) \ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z, x))$$

Fact/Query: $\exists \vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \ \psi(\vec{y},\vec{z}) \ )$

Example:

$$\rho = male(x) \land human(x) \rightarrow \exists z(female(z) \land parent(z,x))$$

Fact/Query: $\exists \vec{x}(p(\vec{t_1}) \land \cdots \land p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \; (\; \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \; \psi(\vec{y}, \vec{z}) \;)$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z, x))$$

Fact/Query: $\exists \vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \to \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \land human(x) \to \exists z(female(z) \land parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t}_1) \land \cdots \land p(\vec{t}_n)) = \{p(\vec{t}_1), \ldots, p(\vec{t}_n)\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \; (\; \phi(\vec{x},\vec{y}) \rightarrow \exists \vec{z} \; \psi(\vec{y},\vec{z}) \;)$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists \vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \land human(x) \rightarrow \exists z(female(z) \land parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t}_1) \land \cdots \land p(\vec{t}_n)) = \{p(\vec{t}_1), \ldots, p(\vec{t}_n)\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}),\dots,p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0,(\rho_0,h_0,F_1),\dots,(\rho_n,h_n,F_{n+1})$

Example: Let $h(x) = Joe$.

1. $F_0 = \{male(Joe), human(Joe)\}$

Preliminaries

Existential Rule: $\forall \vec{x} \forall \vec{y} \; (\; \phi(\vec{x}, \vec{y}) \rightarrow \exists \vec{z} \; \psi(\vec{y}, \vec{z}) \;)$

Example:

$$\rho = \mathit{male}(x) \wedge \mathit{human}(x) \rightarrow \exists z(\mathit{female}(z) \wedge \mathit{parent}(z, x))$$

Fact/Query: $\exists \vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{\mathit{male}(\mathit{Joe}), \mathit{human}(\mathit{Joe})\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Example: Let $h(x) = \mathit{Joe}$.

1. $F_0 = \{\mathit{male}(\mathit{Joe}), \mathit{human}(\mathit{Joe})\}$
2. $F_1 = \{\mathit{male}(\mathit{Joe}), \mathit{human}(\mathit{Joe}), \mathit{female}(z), \mathit{parent}(z, \mathit{Joe})\}$

Preliminaries

Existential Rule: $\forall\vec{x}\forall\vec{y}\ (\ \phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\ \psi(\vec{y},\vec{z})\ )$

Example:

$$\rho = male(x) \wedge human(x) \rightarrow \exists z(female(z) \wedge parent(z,x))$$

Fact/Query: $\exists\vec{x}(p(\vec{t_1}) \wedge \cdots \wedge p(\vec{t_n})) = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$

Example:

$$F = \{male(Joe), human(Joe)\}$$

Derivation: $F_0, (\rho_0, h_0, F_1), \ldots, (\rho_n, h_n, F_{n+1})$

Example: Let $h(x) = Joe$.

1. $F_0 = \{male(Joe), human(Joe)\}$
2. $F_1 = \{male(Joe), human(Joe), female(z), parent(z, Joe)\}$

$\mathcal{D} = F_0, (\rho, h, F_1)$.

A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

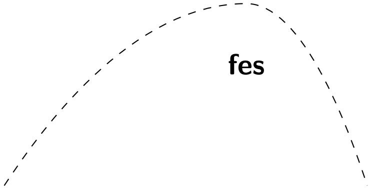
Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

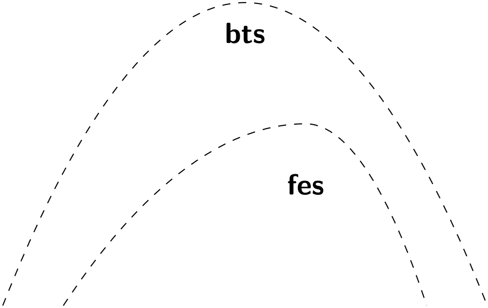


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

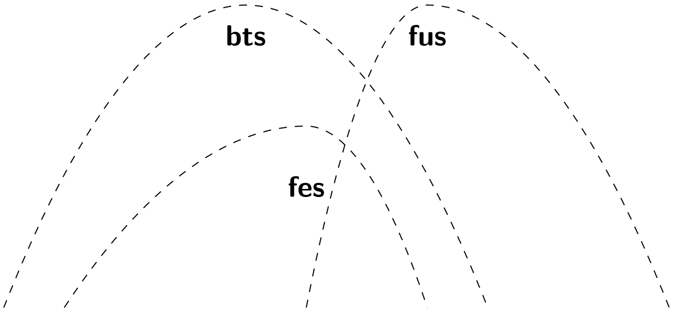


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

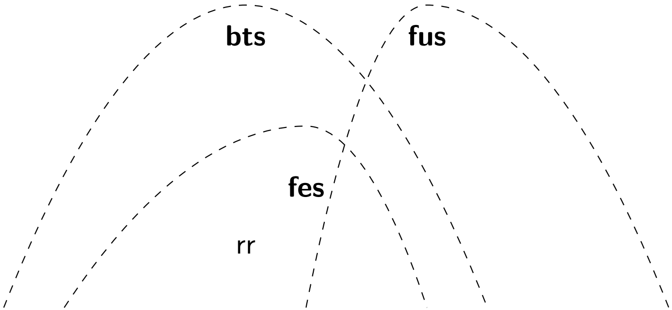


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

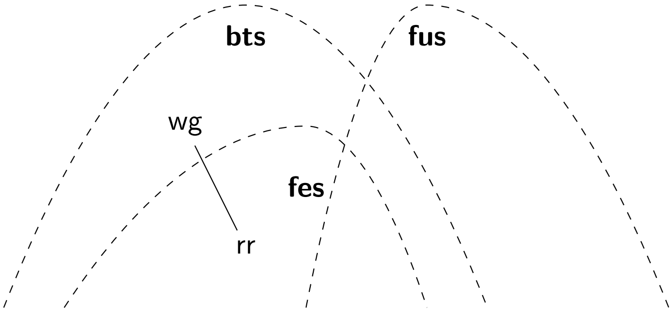


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

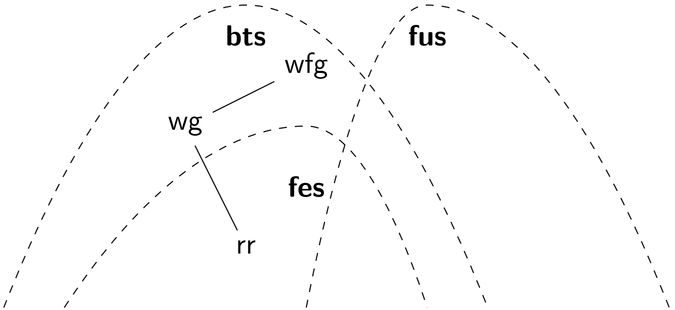


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

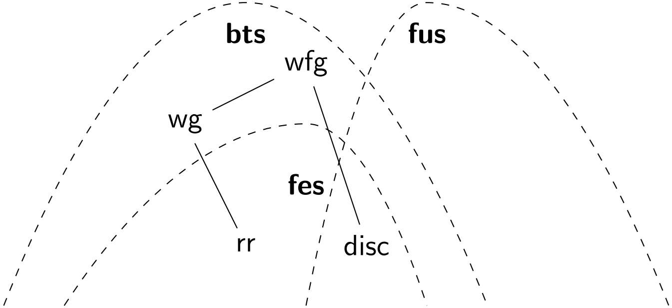


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

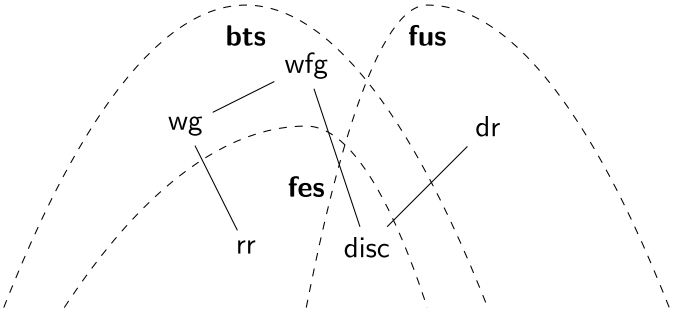


A Jungle of Decidable Classes

Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability

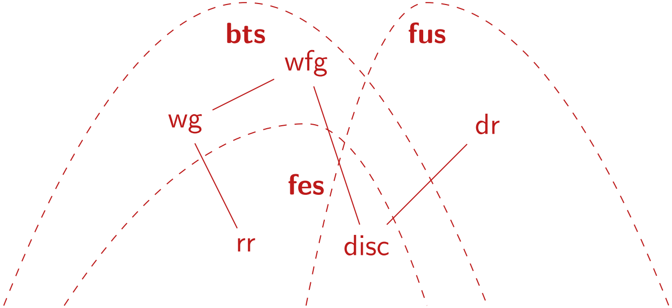


A Jungle of Decidable Classes

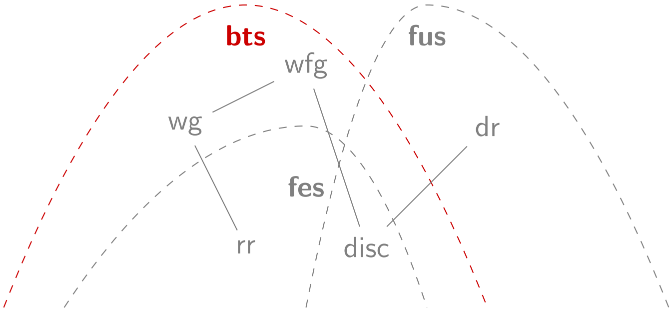
Query Entailment: Let F be a fact and $\mathcal{R}$ be a set of existential rules. Given query $Q = \{p(\vec{t_1}), \ldots, p(\vec{t_n})\}$, does $(F, \mathcal{R}) \models Q$?

Issue: Undecidable!

Solution: Restrict $\mathcal{R} \rightsquigarrow$ Decidability



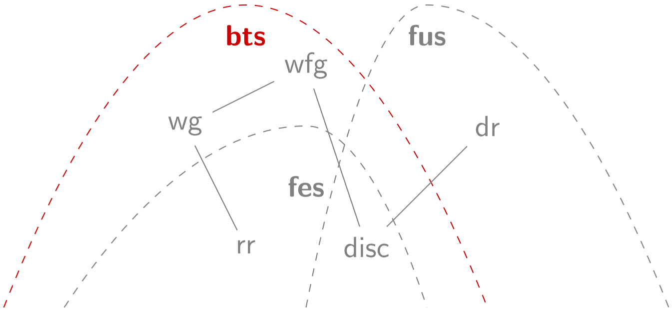
Goal of This Project



Goals:

- Focus: BTS, Derivation Graphs, and Greediness
- Relationships b/w decidable classes
- Generalize existing notions

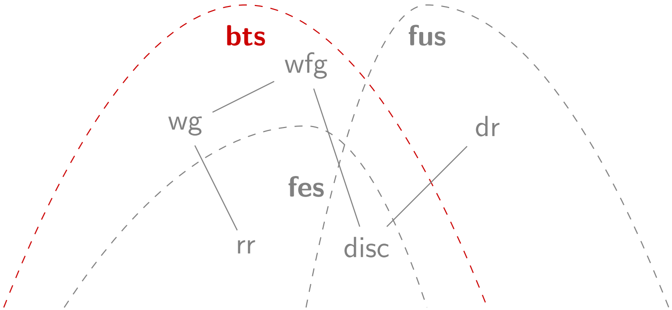
Goal of This Project



Goals:

- Focus: BTS, Derivation Graphs, and Greediness
- Relationships b/w decidable classes
- Generalize existing notions

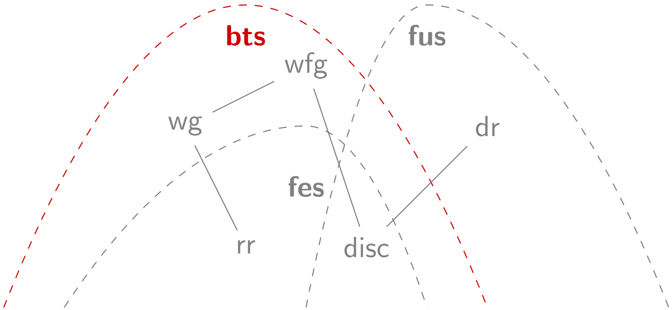
Goal of This Project



Goals:

- Focus: BTS, Derivation Graphs, and Greediness
- Relationships b/w decidable classes
- Generalize existing notions

Goal of This Project



Goals:

- Focus: BTS, Derivation Graphs, and Greediness
- Relationships b/w decidable classes
- Generalize existing notions

Tree Decomposition

Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$

Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$

Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$

Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

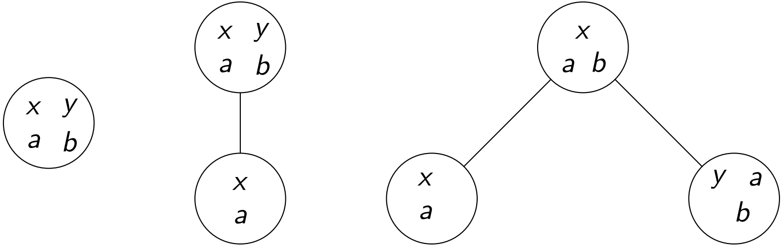
- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

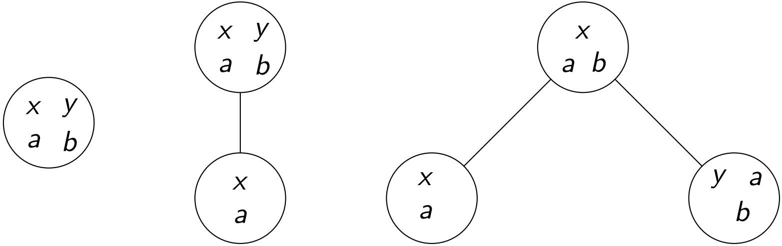


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \dots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

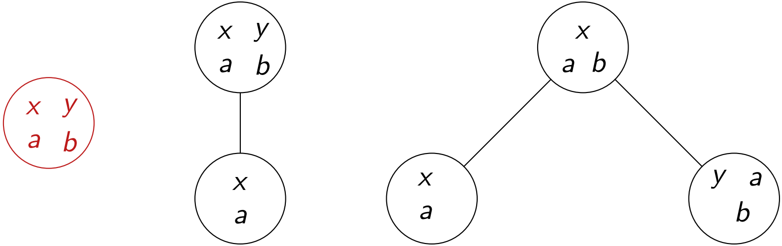


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

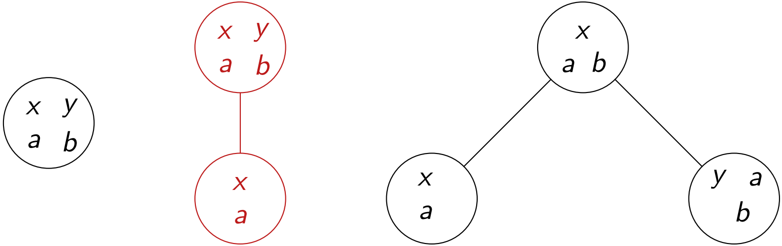


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

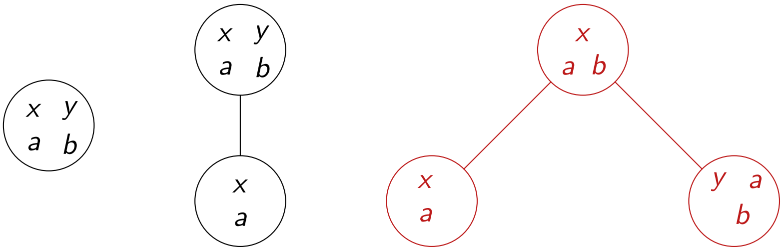


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

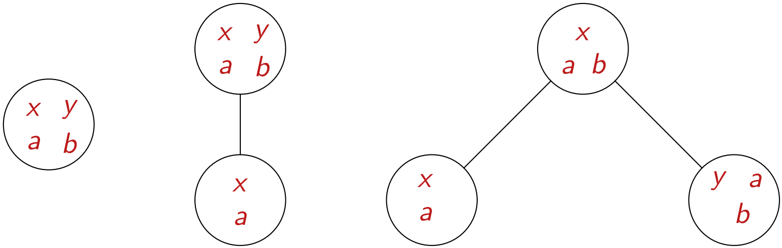


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

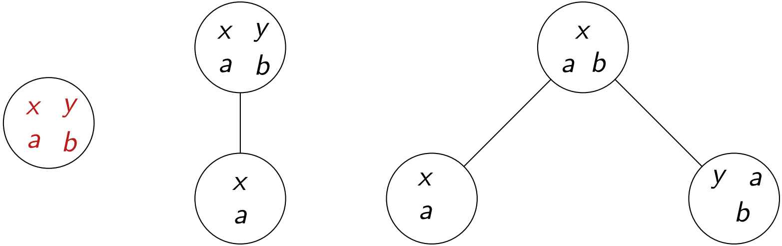


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

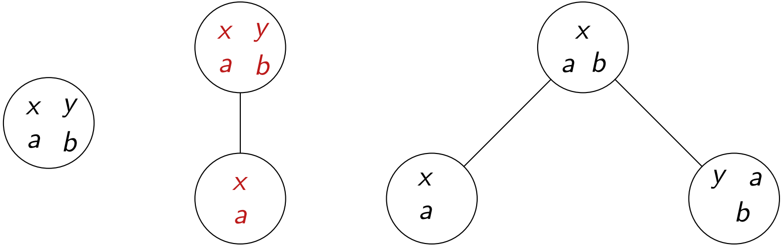


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

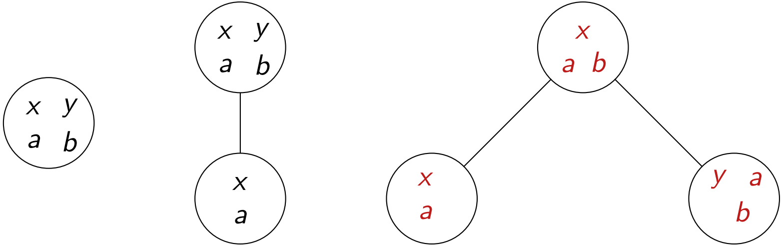


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

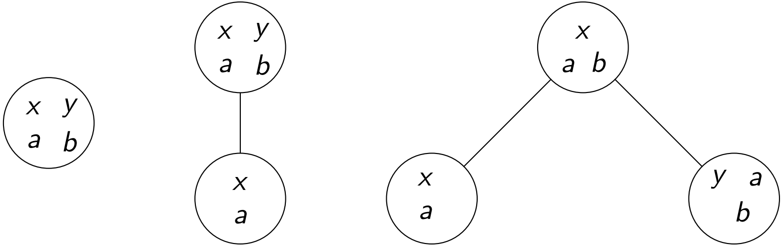


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

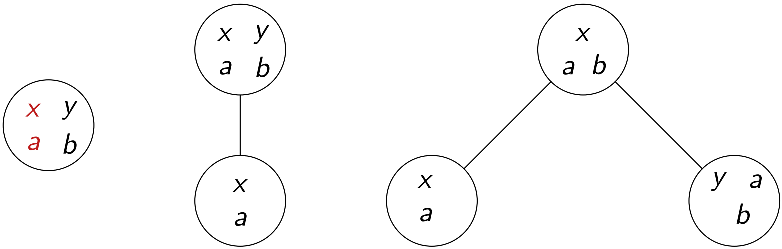


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq \mathit{term}(F)$
- $\bigcup_i X_i = \mathit{term}(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

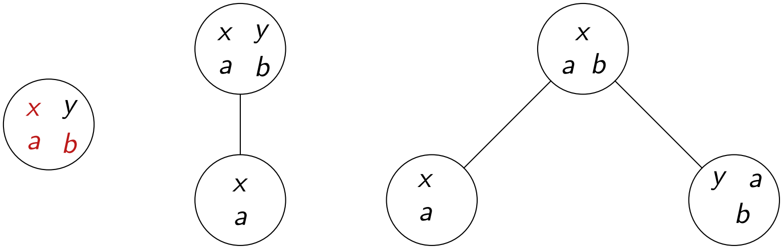


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

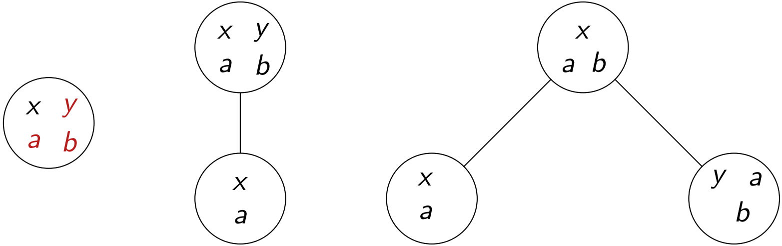


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

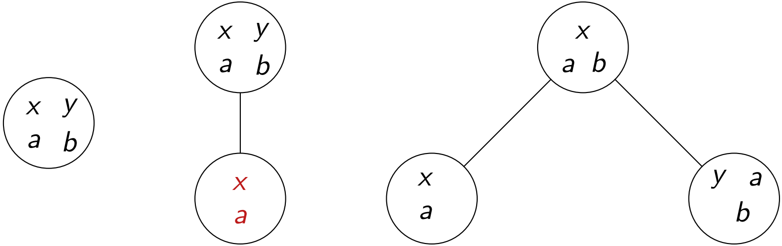


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

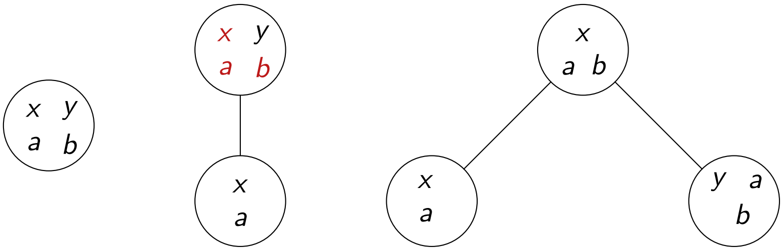


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

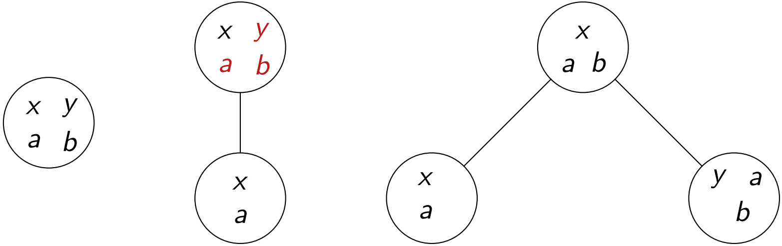


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

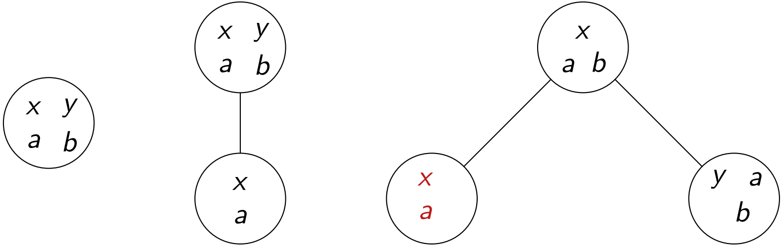


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

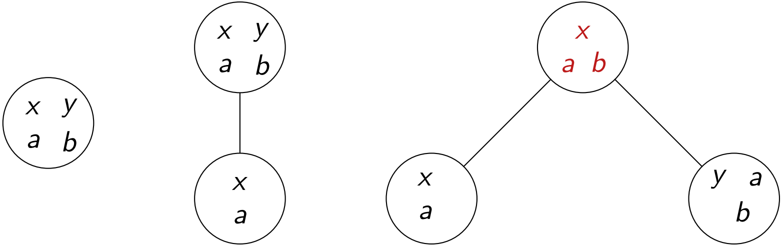


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \dots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \upharpoonright t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

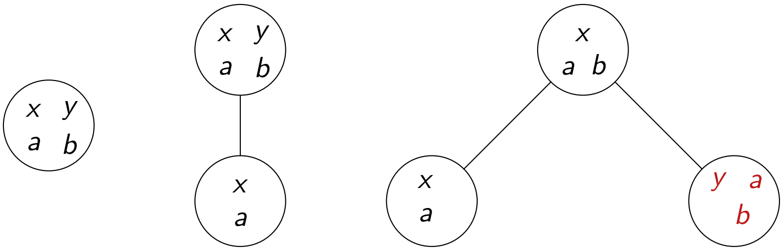


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

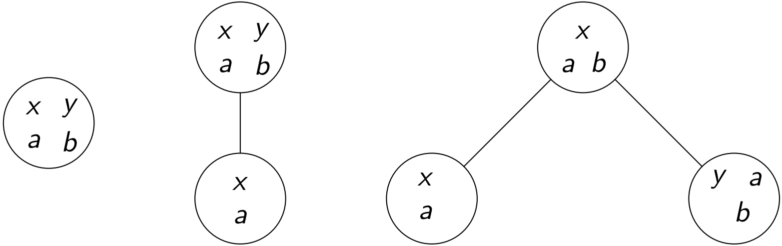


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

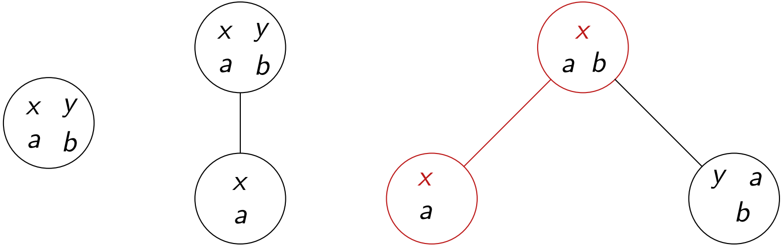


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

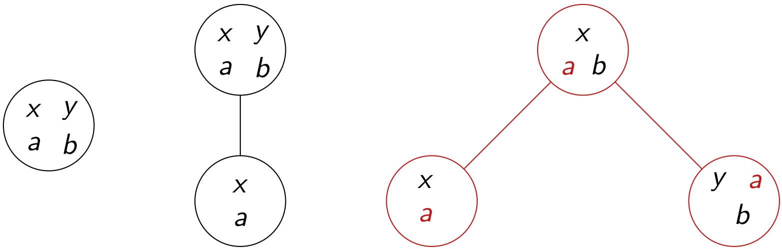


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

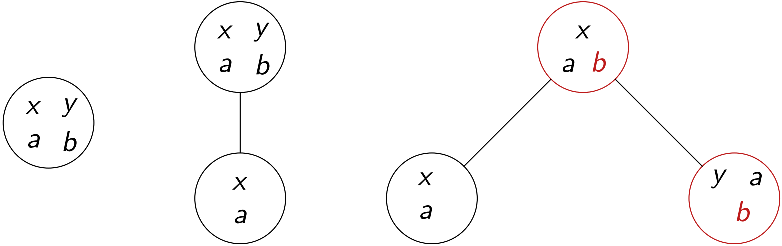


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

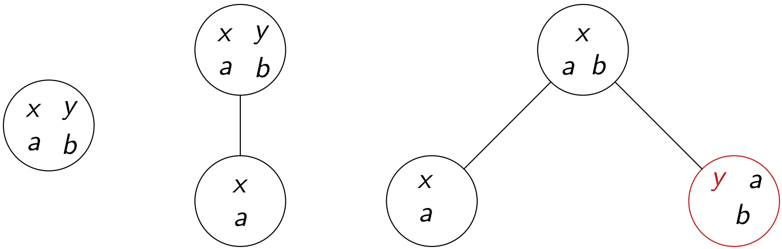


Tree Decomposition

Tree Decomposition: $T(F) = (V, E)$ with $V = \{X_1, \ldots, X_n\}$ and

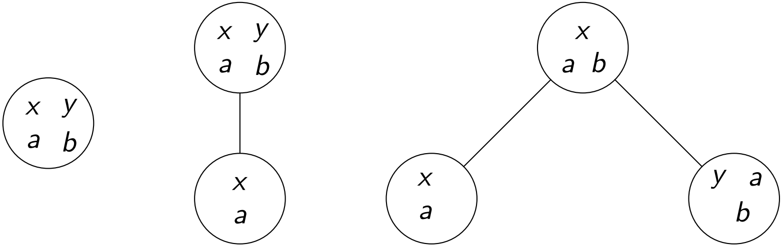
- For each i, $X_i \subseteq term(F)$
- $\bigcup_i X_i = term(F)$
- If $p(\vec{t}) \in F$, then $\exists X_i$ s.t. $\vec{t} \in X_i$
- $T(F) \restriction t$ is connected

Example: Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$



Treewidth

Example (Cont.): Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

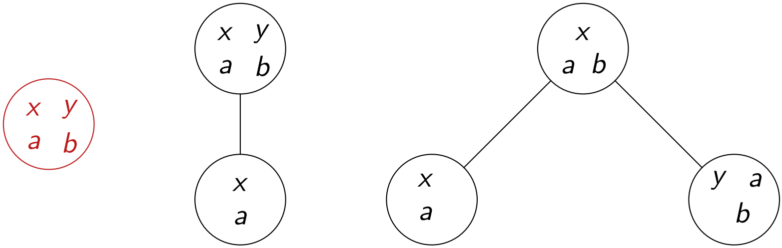


Width: Largest node minus 1.

Treewidth: Smallest width among all tree decompositions.

Treewidth

Example (Cont.): Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

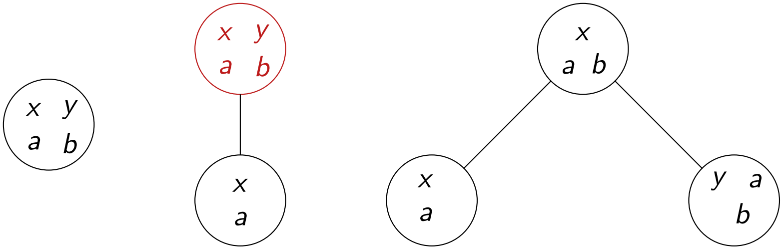


Width: Largest node minus 1.

Treewidth: Smallest width among all tree decompositions.

Treewidth

Example (Cont.): Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

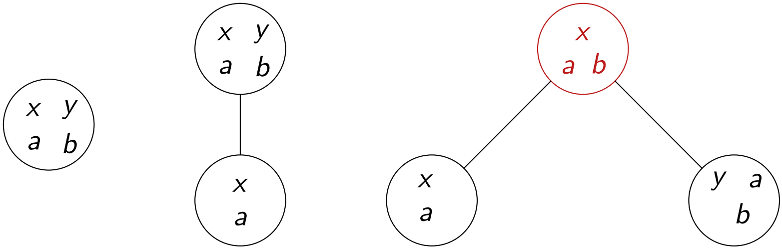


Width: Largest node minus 1.

Treewidth: Smallest width among all tree decompositions.

Treewidth

Example (Cont.): Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$

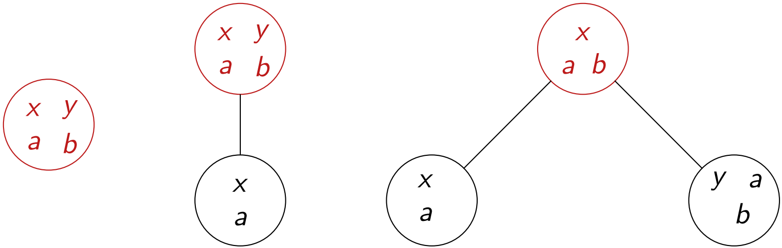


Width: Largest node minus 1.

Treewidth: Smallest width among all tree decompositions.

Treewidth

Example (Cont.): Let $F = \{p(x, a), q(x, a, b), r(y, a, b)\}$



Width: Largest node minus 1.

Treewidth: Smallest width among all tree decompositions.

The Importance of Treewidth

Theorem (Courcelle, 1990)

The first-order theory of a class of first-order structures with bounded treewidth is decidable.

Bounded Treewidth Sets: A rule set $\mathcal{R}$ is **bts** iff for any fact F, $(F, \mathcal{R})$ has a universal model of bounded treewidth.

Universal Model: A model for a FO theory that has a homomorphism into every other model.

Theorem (Baget et al. 2011)

If $\mathcal{R}$ is **bts**, *then query entailment is decidable.*

Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

Alternative Formulation of BTS: **adbts**

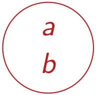
All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

Example: $F = \{p(a, b)\}$, $\mathcal{R} = \{p(x, y) \to p(y, z)\}$, $n = 1$.

Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

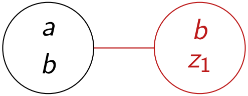
Example: $F = \{p(a, b)\}$, $\mathcal{R} = \{p(x, y) \rightarrow p(y, z)\}$, $n = 1$.



Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

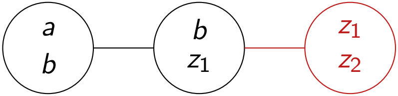
Example: $F = \{p(a, b)\}$, $\mathcal{R} = \{p(x, y) \rightarrow p(y, z)\}$, $n = 1$.



Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

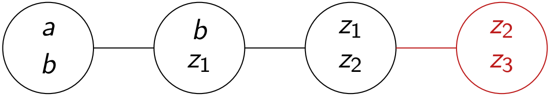
Example: $F = \{p(a, b)\}$, $\mathcal{R} = \{p(x, y) \to p(y, z)\}$, $n = 1$.



Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

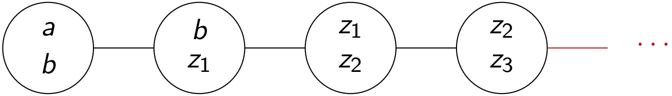
Example: $F = \{p(a, b)\}$, $\mathcal{R} = \{p(x, y) \rightarrow p(y, z)\}$, $n = 1$.



Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

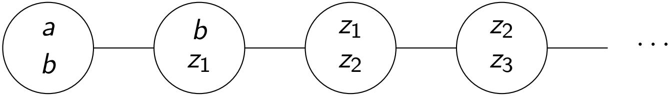
Example: $F = \{p(a, b)\}$, $\mathcal{R} = \{p(x, y) \rightarrow p(y, z)\}$, $n = 1$.



Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

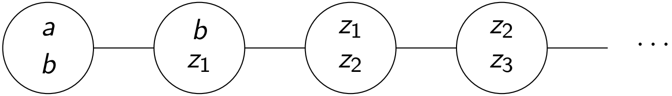
Example: $F = \{p(a,b)\}$, $\mathcal{R} = \{p(x,y) \to p(y,z)\}$, $n = 1$.



Alternative Formulation of BTS: **adbts**

All-Derivation Bounded Treewidth Sets: $\mathcal{R}$ is **adbts** iff for any F, there exists an $n \in \mathbb{N}$, if F' is derivable from F, then $tw(F') \leq n$.

Example: $F = \{p(a,b)\}$, $\mathcal{R} = \{p(x,y) \rightarrow p(y,z)\}$, $n = 1$.



Finding: **adbts** is a *proper* subset of **bts**.

Theorem

adbts $\subsetneq$ **bts**

Derivation Graphs

Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Derivation Graphs

Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Example: $F = \{p(x,y)\}$ and

$\mathcal{R} = \{$

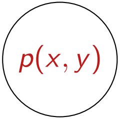
1. $p(x,y) \rightarrow q(y,z)$,
2. $p(x,y) \land q(y,z) \rightarrow p(y,y)$,
3. $p(y,y) \rightarrow q(y,y)$,
4. $q(x,x) \land p(y,z) \rightarrow r(x,y,z)$.

$\}$

Derivation Graphs

Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Example: $F = \{p(x,y)\}$ and



$\mathcal{R} = \{$

1. $p(x,y) \rightarrow q(y,z)$,
2. $p(x,y) \wedge q(y,z) \rightarrow p(y,y)$,
3. $p(y,y) \rightarrow q(y,y)$,
4. $q(x,x) \wedge p(y,z) \rightarrow r(x,y,z)$.

$\}$

Derivation Graphs

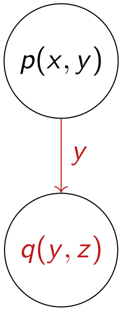
Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Example: $F = \{p(x,y)\}$ and

$\mathcal{R} = \{$

1. $p(x,y) \to q(y,z)$,
2. $p(x,y) \wedge q(y,z) \to p(y,y)$,
3. $p(y,y) \to q(y,y)$,
4. $q(x,x) \wedge p(y,z) \to r(x,y,z)$.

$\}$



Derivation Graphs

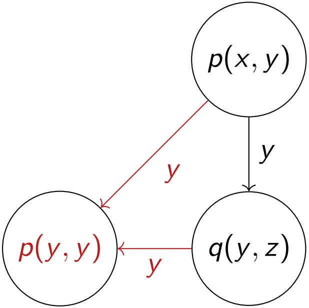
Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Example: $F = \{p(x, y)\}$ and

$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(y, z)$,
2. $p(x, y) \wedge q(y, z) \rightarrow p(y, y)$,
3. $p(y, y) \rightarrow q(y, y)$,
4. $q(x, x) \wedge p(y, z) \rightarrow r(x, y, z)$.

$\}$



Derivation Graphs

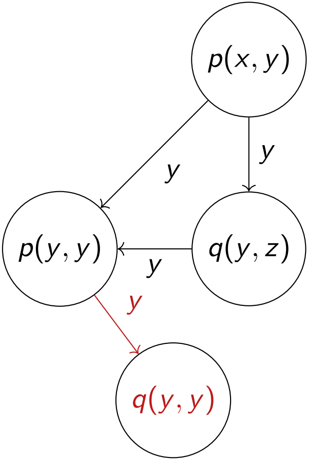
Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Example: $F = \{p(x, y)\}$ and

$\mathcal{R} = \{$

1. $p(x, y) \to q(y, z)$,
2. $p(x, y) \wedge q(y, z) \to p(y, y)$,
3. $p(y, y) \to q(y, y)$,
4. $q(x, x) \wedge p(y, z) \to r(x, y, z)$.

$\}$



Derivation Graphs

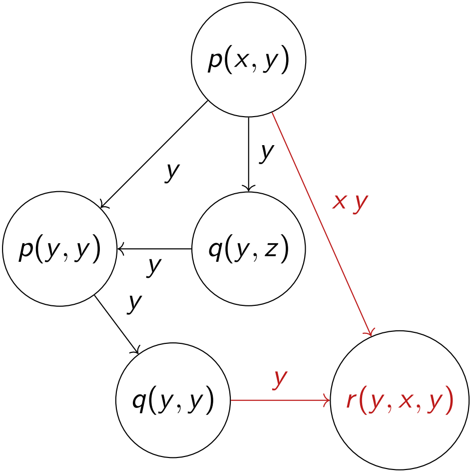
Derivation Graph (Baget et al. 2011): A DAG that keeps track of how facts are derived.

Example: $F = \{p(x,y)\}$ and

$\mathcal{R} = \{$

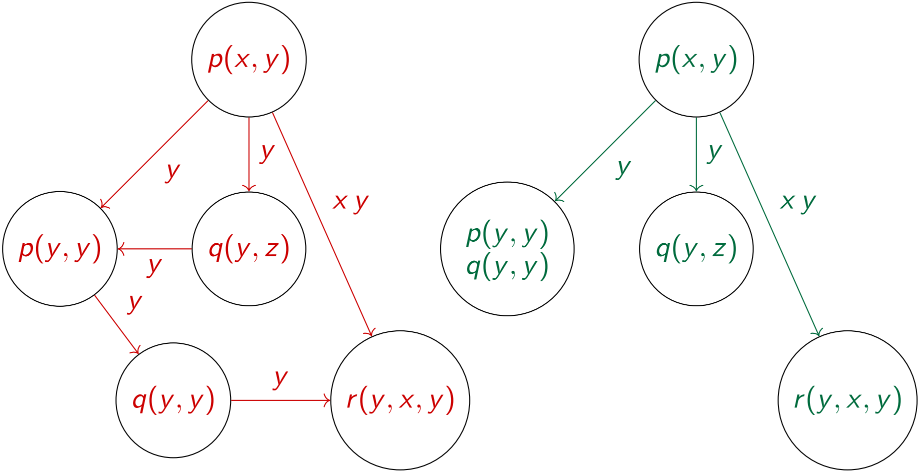
1. $p(x,y) \to q(y,z)$,
2. $p(x,y) \wedge q(y,z) \to p(y,y)$,
3. $p(y,y) \to q(y,y)$,
4. $q(x,x) \wedge p(y,z) \to r(x,y,z)$.

$\}$



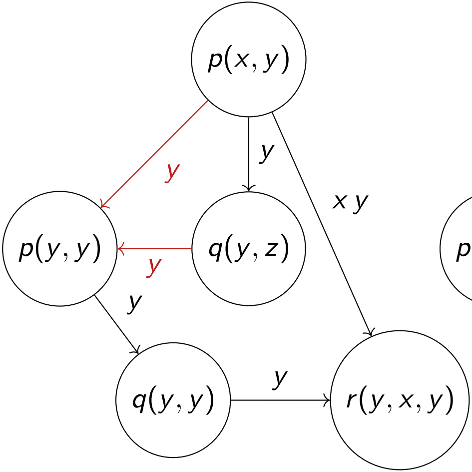
Derivation Graphs and Tree Decompositions

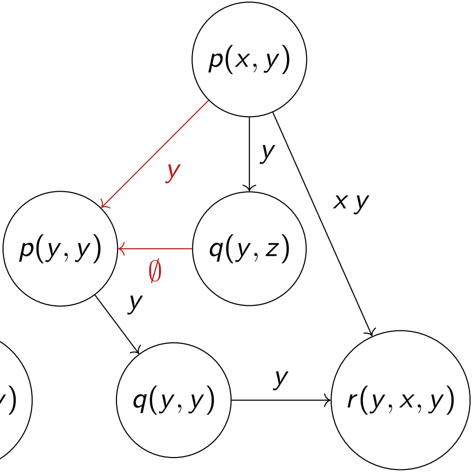
Idea (Baget et al. 2011): For a rule set $\mathcal{R}$, if derivation graph $\rightsquigarrow$ tree for any fact, then $\mathcal{R}$ is BTS.



Reduction Operations

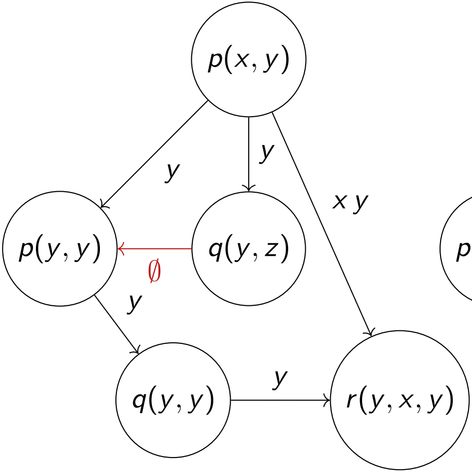
Term Removal: If a term labels two converging arcs, it may be removed from one.

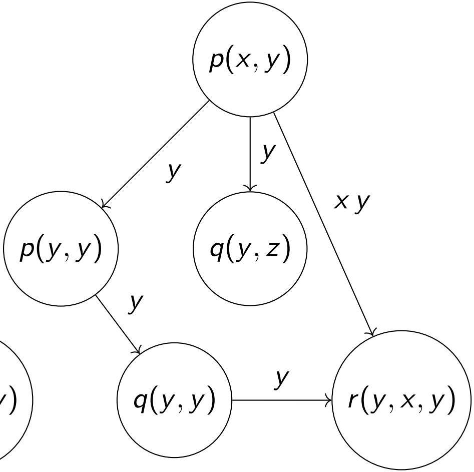




Reduction Operations

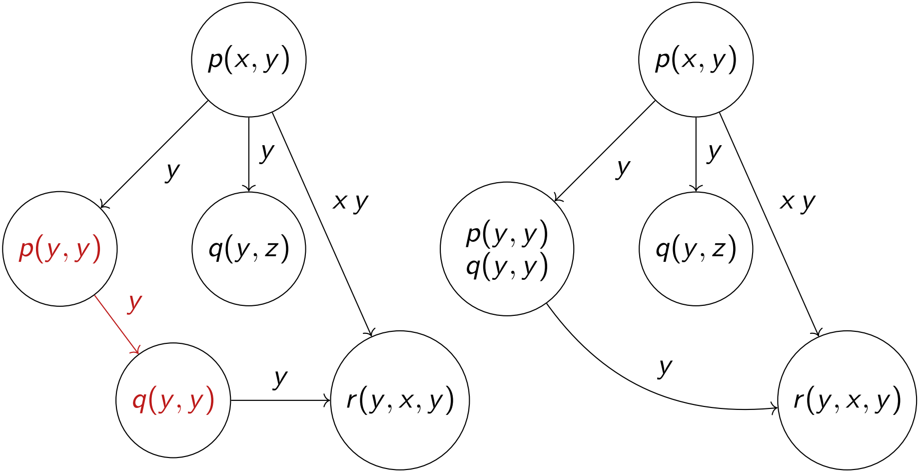
Arc Removal: If an arc is labelled with $\emptyset$, then it may be deleted.





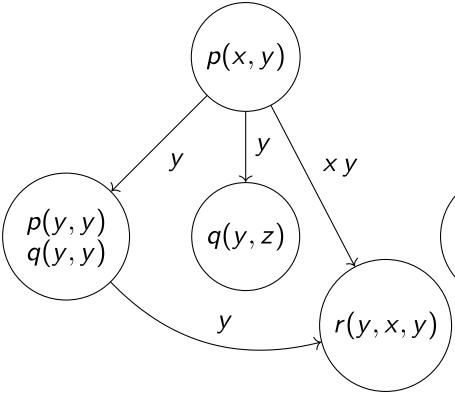
Reduction Operations

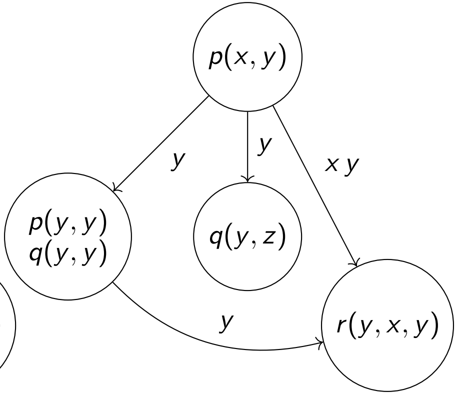
Arc Contraction: If a node X sees a node Y and $terms(Y) \subseteq terms(X)$, then they can be merged.



Reduction Operations

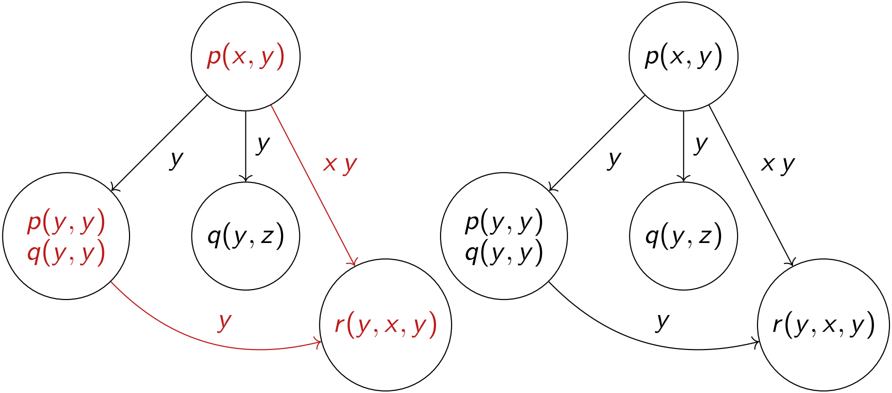
Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.





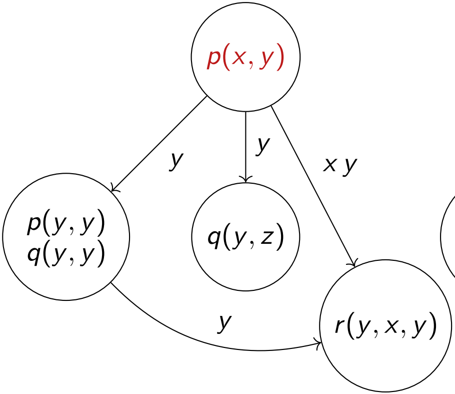
Reduction Operations

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq \mathit{terms}(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = \mathit{Lab}(X, Z) \cup L(Y, Z)$.



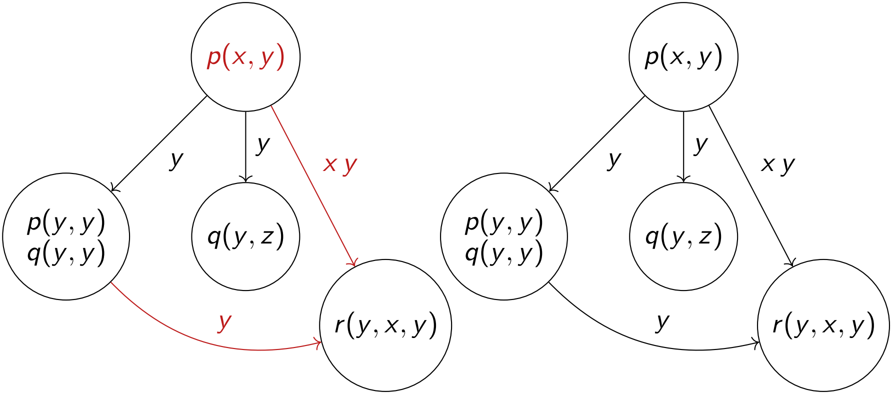
Reduction Operations

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



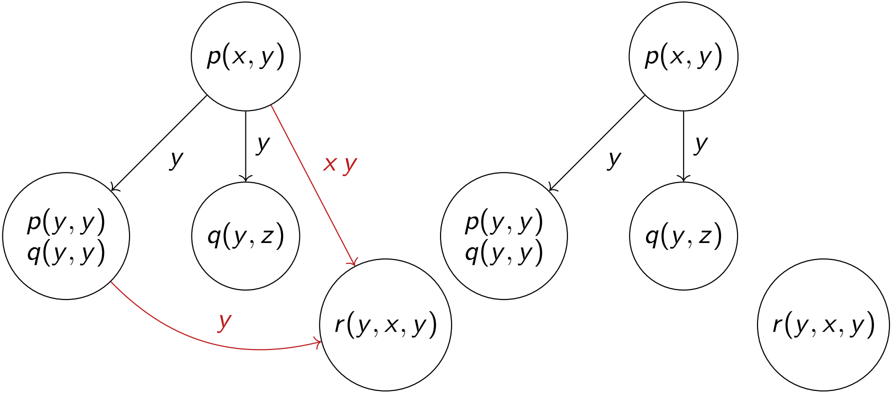
Reduction Operations

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq \mathit{terms}(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = \mathit{Lab}(X, Z) \cup L(Y, Z)$.



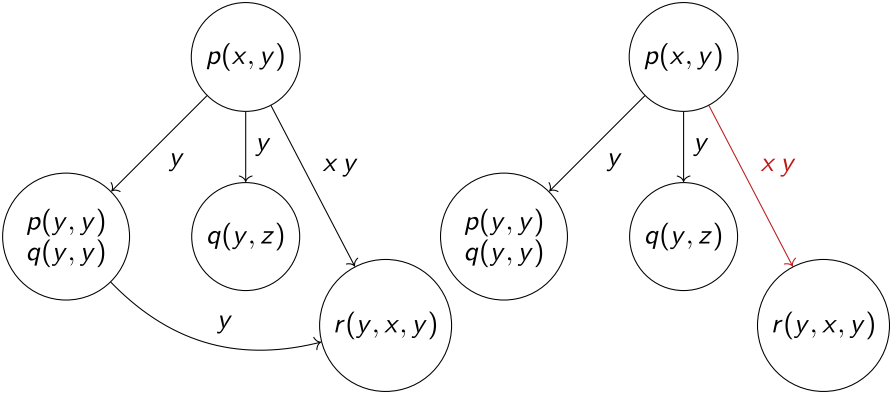
Reduction Operations

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



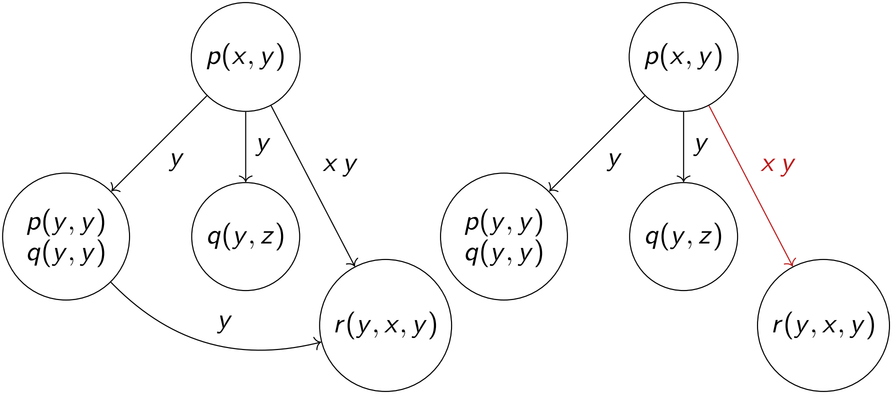
Reduction Operations

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



Reduction Operations

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq \mathit{terms}(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = \mathit{Lab}(X, Z) \cup L(Y, Z)$.



Properties of Reduced Derivation Graphs

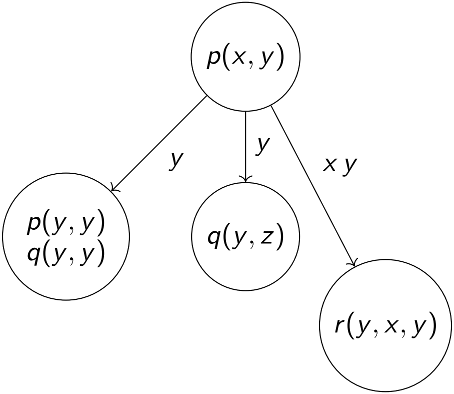
Theorem (Baget et al. 2011)

1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R\in\mathcal{R}}\ \}$.

$F = \{p(x, y)\}$
$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(y, z)$,
2. $p(x, y) \wedge q(y, z) \rightarrow p(y, y)$,
3. $p(y, y) \rightarrow q(y, y)$,
4. $q(x, x) \wedge p(y, z) \rightarrow r(x, y, z)$

$\}$



Properties of Reduced Derivation Graphs

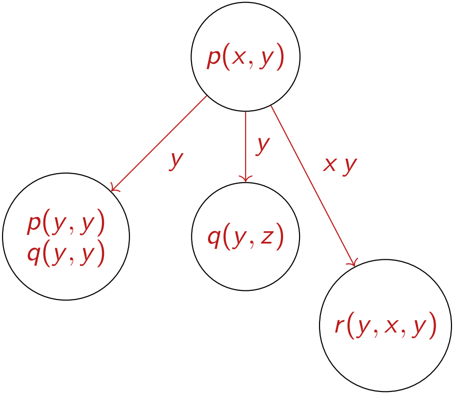
Theorem (Baget et al. 2011)

1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R \in \mathcal{R}}\ \}$.

$F = \{p(x, y)\}$
$\mathcal{R} = \{$

1. $p(x, y) \to q(y, z)$,
2. $p(x, y) \wedge q(y, z) \to p(y, y)$,
3. $p(y, y) \to q(y, y)$,
4. $q(x, x) \wedge p(y, z) \to r(x, y, z)$

$\}$



Properties of Reduced Derivation Graphs

Theorem (Baget et al. 2011)

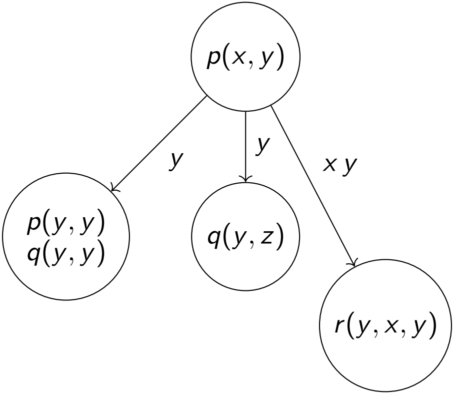
1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R\in\mathcal{R}}\ \}$.

$F = \{p(x, y)\}$

$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(y, z)$,
2. $p(x, y) \wedge q(y, z) \rightarrow p(y, y)$,
3. $p(y, y) \rightarrow q(y, y)$,
4. $q(x, x) \wedge p(y, z) \rightarrow r(x, y, z)$

$\}$



Properties of Reduced Derivation Graphs

Theorem (Baget et al. 2011)

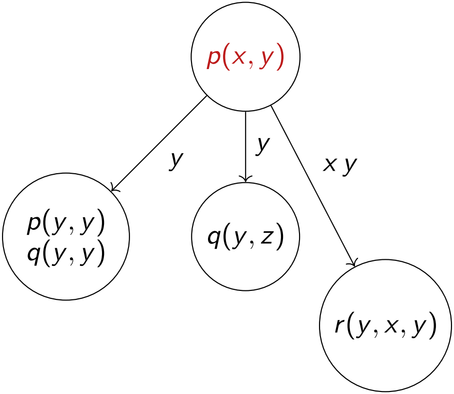
1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R \in \mathcal{R}}\ \}$.

$F = \{p(x,y)\}$

$\mathcal{R} = \{$

1. $p(x,y) \to q(y,z)$,
2. $p(x,y) \wedge q(y,z) \to p(y,y)$,
3. $p(y,y) \to q(y,y)$,
4. $q(x,x) \wedge p(y,z) \to r(x,y,z)$

$\}$



Properties of Reduced Derivation Graphs

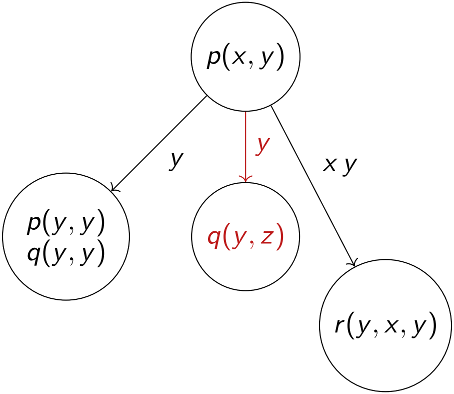
Theorem (Baget et al. 2011)

1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R \in \mathcal{R}}\ \}$.

$F = \{p(x,y)\}$
$\mathcal{R} = \{$

1. $p(x,y) \rightarrow q(y,z)$,
2. $p(x,y) \wedge q(y,z) \rightarrow p(y,y)$,
3. $p(y,y) \rightarrow q(y,y)$,
4. $q(x,x) \wedge p(y,z) \rightarrow r(x,y,z)$

$\}$



Properties of Reduced Derivation Graphs

Theorem (Baget et al. 2011)

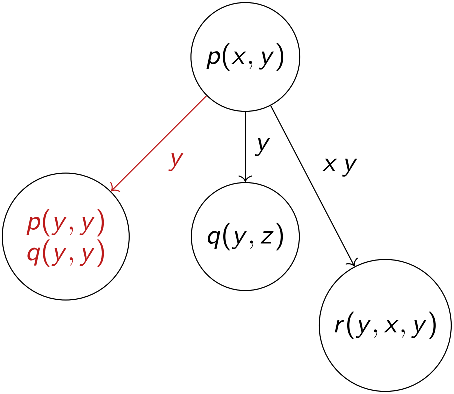
1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R\in\mathcal{R}}\ \}$.

$F = \{p(x,y)\}$

$\mathcal{R} = \{$

1. $p(x,y) \to q(y,z)$,
2. $p(x,y) \wedge q(y,z) \to p(y,y)$,
3. $p(y,y) \to q(y,y)$,
4. $q(x,x) \wedge p(y,z) \to r(x,y,z)$

$\}$



Properties of Reduced Derivation Graphs

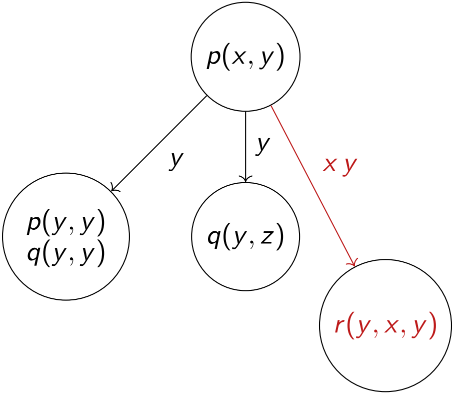
Theorem (Baget et al. 2011)

1. *A reduced derivation graph is a tree decomposition.*
2. *Width is bounded by* $\max\{\ |terms(F)|\ ,\ |terms(head(R))|_{R\in\mathcal{R}}\ \}$.

$F = \{p(x,y)\}$
$\mathcal{R} = \{$

1. $p(x,y) \rightarrow q(y,z)$,
2. $p(x,y) \wedge q(y,z) \rightarrow p(y,y)$,
3. $p(y,y) \rightarrow q(y,y)$,
4. $q(x,x) \wedge p(y,z) \rightarrow r(x,y,z)$

$\}$



Derivation Graph Sets

Acyclic Derivation Graph Set (adgs): A rule set $\mathcal{R}$ is **adgs** iff for any F and any derivable F', the derivation graph of every derivation of F' from F is reducible to a tree.

Weakly Acyclic Derivation Graph Set (wadgs): A rule set $\mathcal{R}$ is **wadgs** iff for any F and any derivable F', the derivation graph of some derivation of F' from F is reducible to a tree.

Theorem

- **adgs** $\subset$ **wadgs** $\subseteq$ **adbts** $\subset$ **bts**

Derivation Graph Sets

Acyclic Derivation Graph Set (adgs): A rule set $\mathcal{R}$ is **adgs** iff for any F and any derivable F', the derivation graph of every derivation of F' from F is reducible to a tree.

Weakly Acyclic Derivation Graph Set (wadgs): A rule set $\mathcal{R}$ is **wadgs** iff for any F and any derivable F', the derivation graph of some derivation of F' from F is reducible to a tree.

Theorem

- **adgs** $\subset$ **wadgs** $\subseteq$ **adbts** $\subset$ **bts**

Proof: Let $\mathcal{R}$ be **wadgs** and F be a fact.

Derivation Graph Sets

Acyclic Derivation Graph Set (adgs): A rule set $\mathcal{R}$ is **adgs** iff for any F and any derivable F', the derivation graph of every derivation of F' from F is reducible to a tree.

Weakly Acyclic Derivation Graph Set (wadgs): A rule set $\mathcal{R}$ is **wadgs** iff for any F and any derivable F', the derivation graph of some derivation of F' from F is reducible to a tree.

Theorem

- **adgs** $\subset$ **wadgs** $\subseteq$ **adbts** $\subset$ **bts**

Proof: Let $\mathcal{R}$ be **wadgs** and F be a fact.

1. Assume F' is $\mathcal{R}$-derivable from F with $\mathcal{D}$

Derivation Graph Sets

Acyclic Derivation Graph Set (adgs): A rule set $\mathcal{R}$ is **adgs** iff for any F and any derivable F', the derivation graph of every derivation of F' from F is reducible to a tree.

Weakly Acyclic Derivation Graph Set (wadgs): A rule set $\mathcal{R}$ is **wadgs** iff for any F and any derivable F', the derivation graph of some derivation of F' from F is reducible to a tree.

Theorem

- **adgs** $\subset$ **wadgs** $\subseteq$ **adbts** $\subset$ **bts**

Proof: Let $\mathcal{R}$ be **wadgs** and F be a fact.

1. Assume F' is $\mathcal{R}$-derivable from F with $\mathcal{D}$
2. $DG(\mathcal{D})$ reducible to tree decomposition $T(F')$

Derivation Graph Sets

Acyclic Derivation Graph Set (adgs): A rule set $\mathcal{R}$ is **adgs** iff for any F and any derivable F', the derivation graph of every derivation of F' from F is reducible to a tree.

Weakly Acyclic Derivation Graph Set (wadgs): A rule set $\mathcal{R}$ is **wadgs** iff for any F and any derivable F', the derivation graph of some derivation of F' from F is reducible to a tree.

Theorem

- **adgs** $\subset$ **wadgs** $\subseteq$ **adbts** $\subset$ **bts**

Proof: Let $\mathcal{R}$ be **wadgs** and F be a fact.

1. Assume F' is $\mathcal{R}$-derivable from F with $\mathcal{D}$
2. $DG(\mathcal{D})$ reducible to tree decomposition $T(F')$
3. $tw(F') \leq w(T(F')) \leq N = \max\{|terms(F)|, |terms(head(R))|_{R\in\mathcal{R}}\}$

Derivation Graph Sets

Acyclic Derivation Graph Set (adgs): A rule set $\mathcal{R}$ is **adgs** iff for any F and any derivable F', the derivation graph of every derivation of F' from F is reducible to a tree.

Weakly Acyclic Derivation Graph Set (wadgs): A rule set $\mathcal{R}$ is **wadgs** iff for any F and any derivable F', the derivation graph of some derivation of F' from F is reducible to a tree.

Theorem

- **adgs** $\subset$ **wadgs** $\subseteq$ **adbts** $\subset$ **bts**
- *If $\mathcal{R}$ is* **adgs** *or* **wadgs***, then query entailment is decidable.*

Proof: Let $\mathcal{R}$ be **wadgs** and F be a fact.

1. Assume F' is $\mathcal{R}$-derivable from F with $\mathcal{D}$
2. $DG(\mathcal{D})$ reducible to tree decomposition $T(F')$
3. $tw(F') \leq w(T(F')) \leq N = \max\{|terms(F)|, |terms(head(R))|_{R \in \mathcal{R}}\}$

Greedy Derivations

Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Greedy Derivations

Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Greedy Derivations

Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Greedy Derivations

Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Greedy Derivations

Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Example:

$F = \{p(x, y)\}$
$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(y, z)$,
2. $p(x, y) \wedge q(y, z) \rightarrow q(y, x)$,
3. $q(y, x) \wedge q(y, z) \rightarrow r(x, y)$.

$\}$

Greedy Derivations

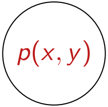
Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Example:

$F = \{p(x, y)\}$
$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(y, z)$,
2. $p(x, y) \wedge q(y, z) \rightarrow q(y, x)$,
3. $q(y, x) \wedge q(y, z) \rightarrow r(x, y)$.

$\}$



Greedy Derivations

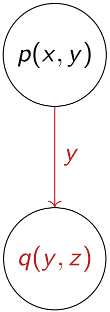
Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Example:

$F = \{p(x, y)\}$
$\mathcal{R} = \{$

1. $p(x, y) \to q(y, z)$,
2. $p(x, y) \land q(y, z) \to q(y, x)$,
3. $q(y, x) \land q(y, z) \to r(x, y)$.

$\}$



Greedy Derivations

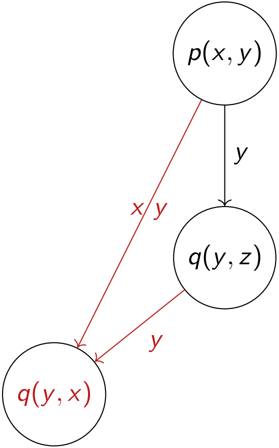
Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Example:

$F = \{p(x, y)\}$
$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(y, z)$,
2. $p(x, y) \wedge q(y, z) \rightarrow q(y, x)$,
3. $q(y, x) \wedge q(y, z) \rightarrow r(x, y)$.

$\}$



Greedy Derivations

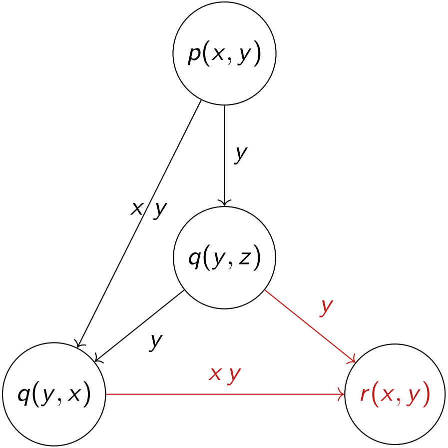
Greedy Derivation (Thomazo et al. 2012) Frontier of any rule application is (1) from the KB, or (2) from a single previous rule application.

Example:

$F = \{p(x,y)\}$
$\mathcal{R} = \{$

1. $p(x,y) \to q(y,z)$,
2. $p(x,y) \wedge q(y,z) \to q(y,x)$,
3. $q(y,x) \wedge q(y,z) \to r(x,y)$.

$\}$



Greedy Derivations and Decidability

Greedy Bounded Treewidth Set (Thomazo et al. 2012): $\mathcal{R}$ is **gbts** iff for all facts F and F', if F' is derivable from F, then every derivation of F' from F is greedy.

Weakly Greedy Bounded Treewidth Set: $\mathcal{R}$ is **wgbts** iff for all facts F and F', if F' is derivable from F, then some derivation of F' from F is greedy.

Theorem

- **gbts** = **adgs**
- **gbts** $\subset$ **wgbts** = **wadgs** $\subseteq$ **bts**.
- *If $\mathcal{R}$ is* **gbts** *or* **wgbts**, *then query entailment is decidable.*

Greedy Derivations and Decidability

Greedy Bounded Treewidth Set (Thomazo et al. 2012): $\mathcal{R}$ is **gbts** iff for all facts F and F', if F' is derivable from F, then every derivation of F' from F is greedy.

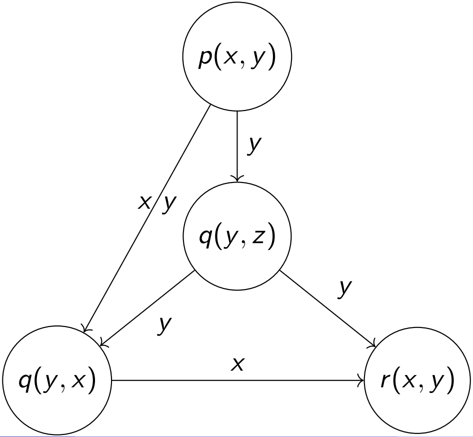
Weakly Greedy Bounded Treewidth Set: $\mathcal{R}$ is **wgbts** iff for all facts F and F', if F' is derivable from F, then some derivation of F' from F is greedy.

Theorem

- **gbts** $=$ **adgs**
- **gbts** $\subset$ **wgbts** $=$ **wadgs** $\subseteq$ **bts**.
- *If $\mathcal{R}$ is* **gbts** *or* **wgbts**, *then query entailment is decidable.*

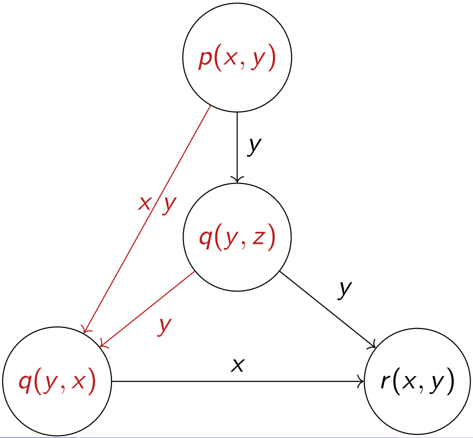
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq \mathit{terms}(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = \mathit{Lab}(X, Z) \cup L(Y, Z)$.



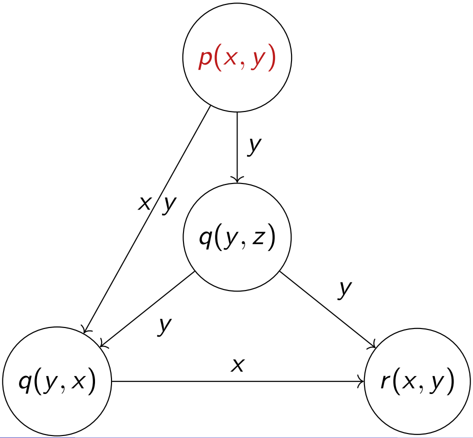
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



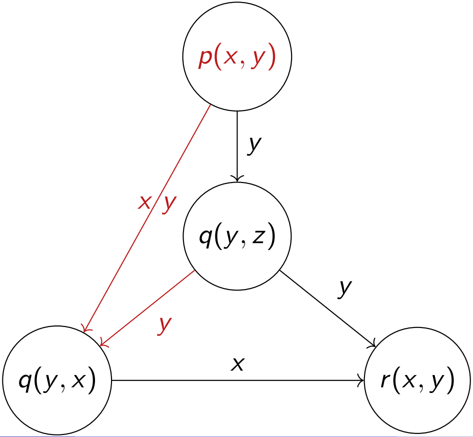
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



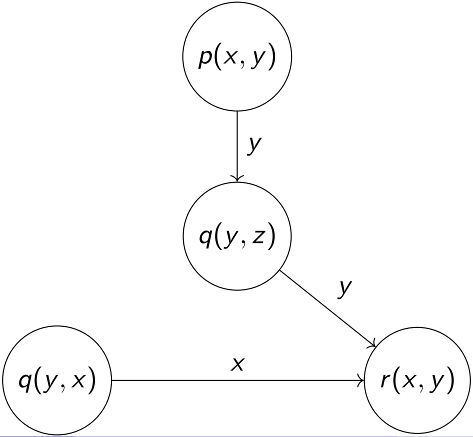
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



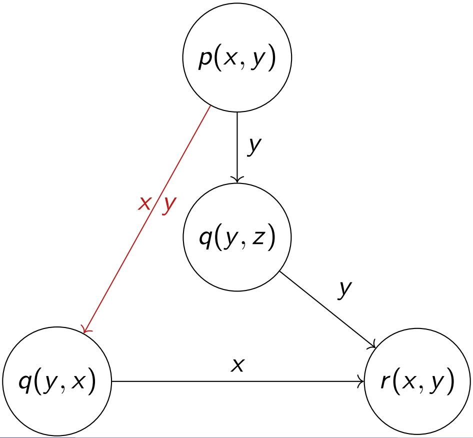
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



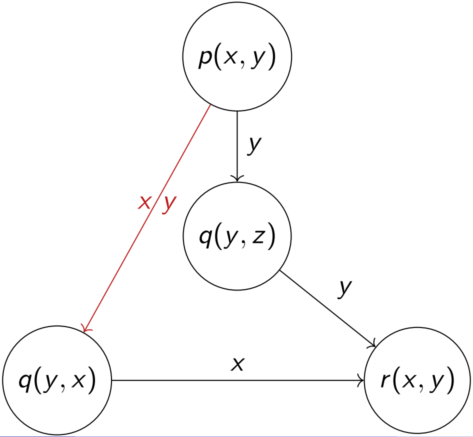
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an “ancestor” W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



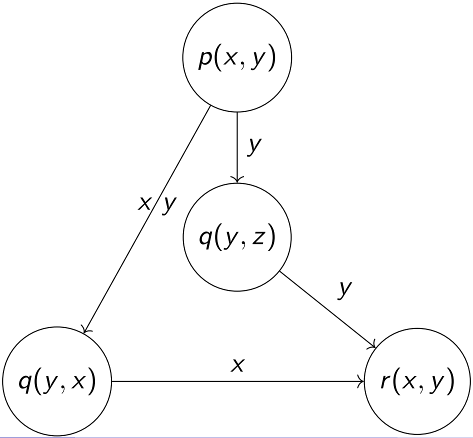
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq \mathit{terms}(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = \mathit{Lab}(X, Z) \cup L(Y, Z)$.



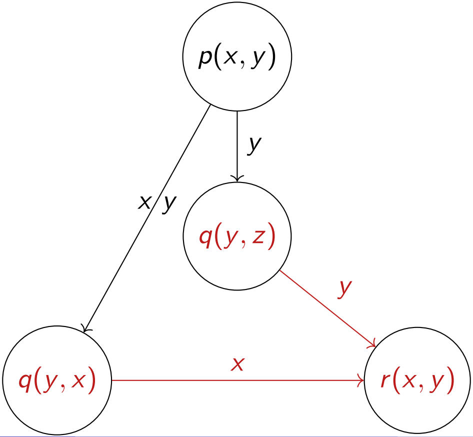
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



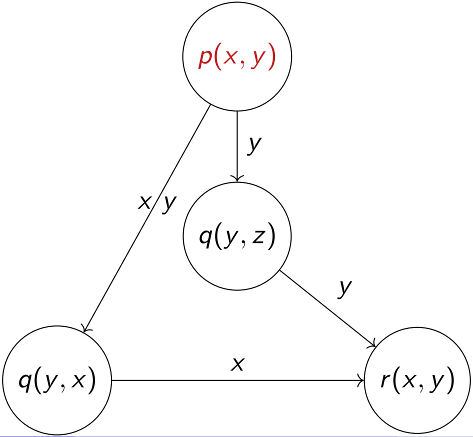
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



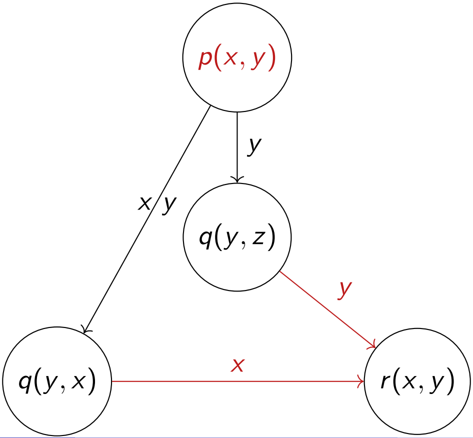
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an “ancestor” W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



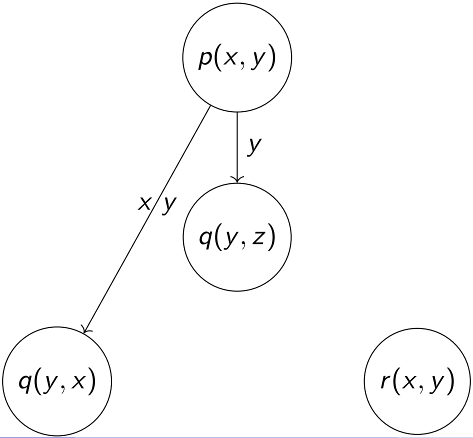
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



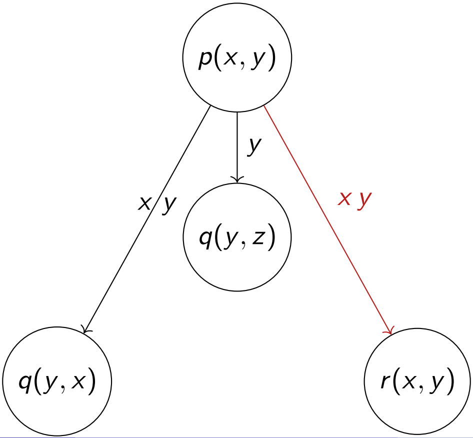
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



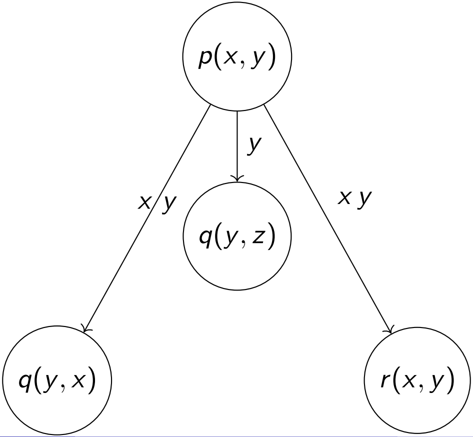
Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



Greedy Derivations and Derivation Graphs

Cycle Removal: If there are arcs (X, Z) and (Y, Z) with an "ancestor" W such that $L(X, Z) \cup L(Y, Z) \subseteq terms(W)$, then (1) delete (X, Z), (Y, Z), (2) add (W, Z), and (3) let $L(W, Z) = Lab(X, Z) \cup L(Y, Z)$.



Greedy Derivations and Decidability

Greedy Bounded Treewidth Set (Thomazo et al. 2012): $\mathcal{R}$ is **gbts** iff for all facts F and F', if F' is derivable from F, then every derivation of F' from F is greedy.

Weakly Greedy Bounded Treewidth Set: $\mathcal{R}$ is **wgbts** iff for all facts F and F', if F' is derivable from F, then some derivation of F' from F is greedy.

Theorem

- **gbts** $=$ **adgs**
- **gbts** $\subset$ **wgbts** $=$ **wadgs** $\subseteq$ **bts**.
- *If $\mathcal{R}$ is* **gbts** *or* **wgbts***, then query entailment is decidable.*

Greedy Derivations and Decidability

Greedy Bounded Treewidth Set (Thomazo et al. 2012): $\mathcal{R}$ is **gbts** iff for all facts F and F', if F' is derivable from F, then every derivation of F' from F is greedy.

Weakly Greedy Bounded Treewidth Set: $\mathcal{R}$ is **wgbts** iff for all facts F and F', if F' is derivable from F, then some derivation of F' from F is greedy.

Theorem

- **gbts** = **adgs**
- **gbts** $\subset$ **wgbts** = **wadgs** $\subseteq$ **bts**.
- *If $\mathcal{R}$ is* **gbts** *or* **wgbts**, *then query entailment is decidable.*

Rule Dependencies

Dependency (Baget 2004): A rule ρ' depends on a rule ρ, if there exists a fact F such that applying ρ to F gives a fact F' that ρ' can be applied to (and was not previously applicable to).

Example: Let $F = \{p(a, b)\}$

$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(x, y)$
2. $q(x, y) \rightarrow r(x, y)$

$\}$

Rule Dependencies

Dependency (Baget 2004): A rule ρ' depends on a rule ρ, if there exists a fact F such that applying ρ to F gives a fact F' that ρ' can be applied to (and was not previously applicable to).

Example: Let $F = \{p(a, b)\}$

$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(x, y)$
2. $q(x, y) \rightarrow r(x, y)$

$\}$

Rule Dependencies

Dependency (Baget 2004): A rule ρ' depends on a rule ρ, if there exists a fact F such that applying ρ to F gives a fact F' that ρ' can be applied to (and was not previously applicable to).

Example: Let $F = \{p(a, b)\}$

$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(x, y)$
2. $q(x, y) \rightarrow r(x, y)$

$\}$

1. $\{p(a, b)\}$

Rule Dependencies

Dependency (Baget 2004): A rule ρ' depends on a rule ρ, if there exists a fact F such that applying ρ to F gives a fact F' that ρ' can be applied to (and was not previously applicable to).

Example: Let $F = \{p(a, b)\}$

$\mathcal{R} = \{$

1. $p(x, y) \to q(x, y)$
2. $q(x, y) \to r(x, y)$

$\}$

1. $\{p(a, b)\}$
2. $\{p(a, b), q(a, b)\}$

Rule Dependencies

Dependency (Baget 2004): A rule ρ' depends on a rule ρ, if there exists a fact F such that applying ρ to F gives a fact F' that ρ' can be applied to (and was not previously applicable to).

Example: Let $F = \{p(a, b)\}$

$\mathcal{R} = \{$

1. $p(x, y) \rightarrow q(x, y)$
2. $q(x, y) \rightarrow r(x, y)$

$\}$

1. $\{p(a, b)\}$
2. $\{p(a, b), q(a, b)\}$
3. $\{p(a, b), q(a, b), r(a, b)\}$

Rule Dependencies

Dependency (Baget 2004): A rule ρ' depends on a rule ρ, if there exists a fact F such that applying ρ to F gives a fact F' that ρ' can be applied to (and was not previously applicable to).

Example: Let $F = \{p(a,b)\}$

$\mathcal{R} = \{$

1. $p(x,y) \rightarrow q(x,y)$
2. $q(x,y) \rightarrow r(x,y)$

$\}$

1. $\{p(a,b)\}$
2. $\{p(a,b), q(a,b)\}$
3. $\{p(a,b), q(a,b), r(a,b)\}$

$q(x,y) \rightarrow r(x,y)$ depends on $p(x,y) \rightarrow q(x,y)$

Graph of Rule Dependencies

Graph of Rule Dependencies (Baget 2004): A directed graph showing all dependencies between rules.

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \rightarrow q(x, y)$
2. $\rho_2 = r(x) \rightarrow s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$

Graph of Rule Dependencies

Graph of Rule Dependencies (Baget 2004): A directed graph showing all dependencies between rules.

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \rightarrow q(x, y)$
2. $\rho_2 = r(x) \rightarrow s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$

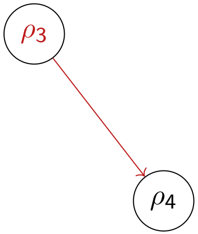
Graph of Rule Dependencies

Graph of Rule Dependencies (Baget 2004): A directed graph showing all dependencies between rules.

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \rightarrow q(x, y)$
2. $\rho_2 = r(x) \rightarrow s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$



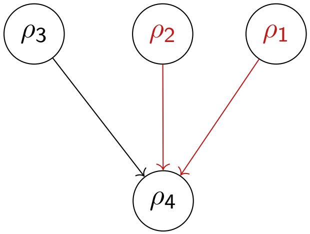
Graph of Rule Dependencies

Graph of Rule Dependencies (Baget 2004): A directed graph showing all dependencies between rules.

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \rightarrow q(x, y)$
2. $\rho_2 = r(x) \rightarrow s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$



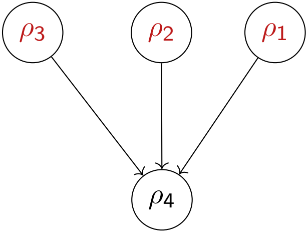
Graph of Rule Dependencies

Graph of Rule Dependencies (Baget 2004): A directed graph showing all dependencies between rules.

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \to q(x, y)$
2. $\rho_2 = r(x) \to s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

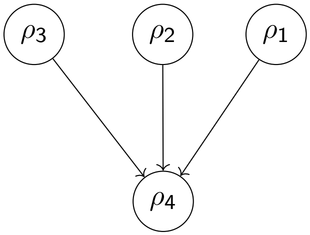


An Observation

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \to q(x, y)$
2. $\rho_2 = r(x) \to s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$



A Derivation: Let $\rho \in \{\rho_1, \rho_2, \rho_3\}$.

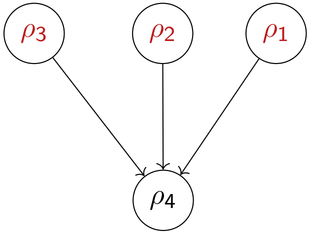
$$F_0, \dots, (\rho_i, h_i, F_i), (\rho, h_{i+1}, F_{i+1}), \dots, (\rho_n, h_n, F_n)$$

An Observation

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \to q(x, y)$
2. $\rho_2 = r(x) \to s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$



A Derivation: Let $\rho \in \{\rho_1, \rho_2, \rho_3\}$.

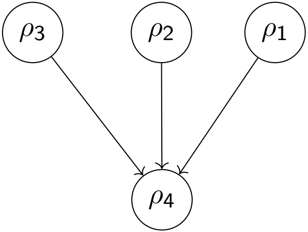
$$F_0, \dots, (\rho_i, h_i, F_i), (\rho, h_{i+1}, F_{i+1}), \dots, (\rho_n, h_n, F_n)$$

An Observation

$\mathcal{R} = \{$

1. $\rho_1 = p(x) \rightarrow q(x, y)$
2. $\rho_2 = r(x) \rightarrow s(x, y, z)$
3. $\rho_3 = p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $\rho_4 = q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$



A Derivation: Let $\rho \in \{\rho_1, \rho_2, \rho_3\}$.

$$F_0, \ldots, (\rho_i, h_i, F_i), (\rho, h_{i+1}, F_{i+1}), \ldots, (\rho_n, h_n, F_n)$$

The Observation: We can permute rule applications!

$$F_0, \ldots, (\rho, h_{i+1}, F_{i-1} \cup (F_{i+1} \setminus F_i)), (\rho_i, h_i, F_{i+1}), \ldots, (\rho_n, h_n, F_n)$$

Re-writing Derivations via Permutations

Lemma (Permutation Lemma)

Suppose we have a derivation of the following form:

$$\mathcal{D} := F_0, \ldots, (\rho_i, h_i, F_{i+1}), (\rho_{i+1}, h_{i+1}, F_{i+2}), \ldots, (\rho_{n-1}, h_{n-1}, F_n)$$

If ρ_{i+1} does not depend on ρ_i, then the following is a derivation as well:

$$\mathcal{D}' := F_0, \ldots, (\rho_{i+1}, h_{i+1}, F_i'), (\rho_i, h_i, F_{i+2}), \ldots, (\rho_{n-1}, h_{n-1}, F_n)$$

where $F_i' = F_i \cup (F_{i+2} - F_{i+1})$.

A Useful Rule Set for Non-Greediness

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \land r(y) \to q(x, z) \land s(y, u, v)$
4. $q(x, y) \land s(z, u, v) \to t(x, z, u, w)$

$\}$

A Useful Rule Set for Non-Greediness

$p(x)$
$r(y)$

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

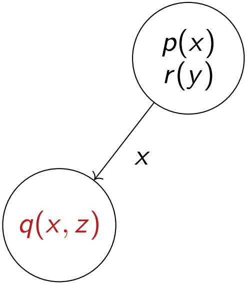
$\}$

A Useful Rule Set for Non-Greediness

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

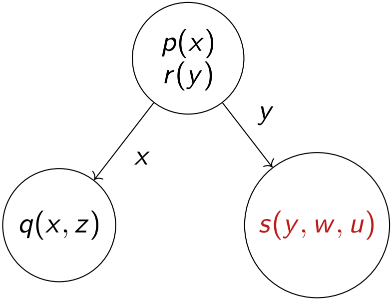


A Useful Rule Set for Non-Greediness

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

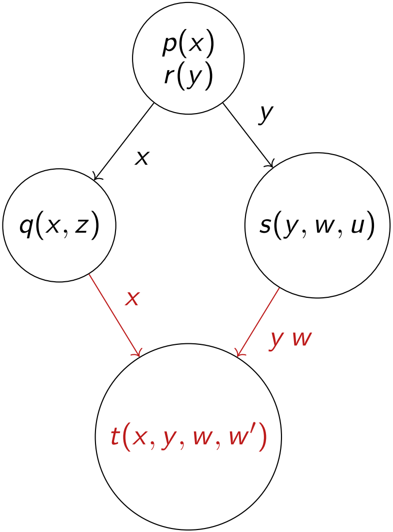


A Useful Rule Set for Non-Greediness

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

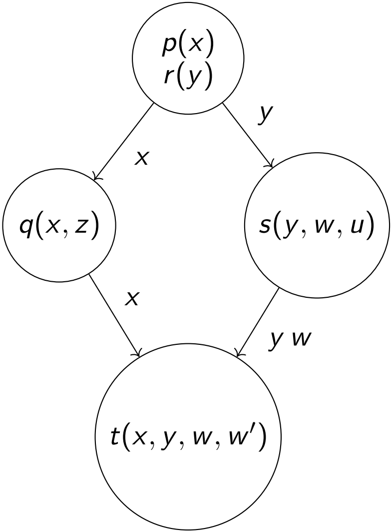


A Useful Rule Set for Non-Greediness

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$



$\mathcal{R}$ is not greedy!

But is $\mathcal{R}$ Weakly Greedy?!

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \land r(y) \to q(x, z) \land s(y, u, v)$
4. $q(x, y) \land s(z, u, v) \to t(x, z, u, w)$

$\}$

But is $\mathcal{R}$ Weakly Greedy?!

$p(x)$
$r(y)$

$\mathcal{R} = \{$

1. $p(x) \rightarrow q(x, y)$
2. $r(x) \rightarrow s(x, y, z)$
3. $p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

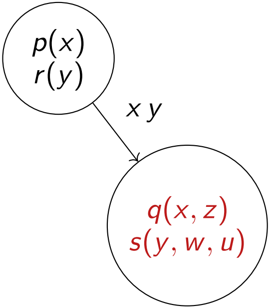
$\}$

But is $\mathcal{R}$ Weakly Greedy?!

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

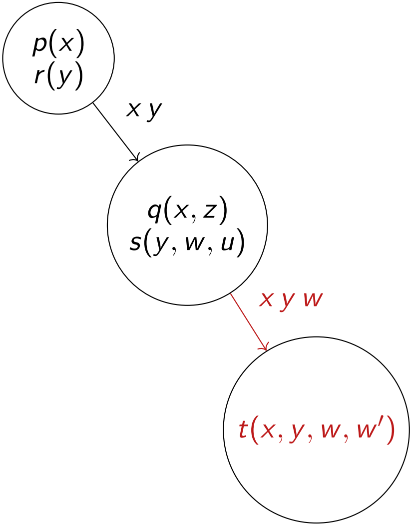


But is $\mathcal{R}$ Weakly Greedy?!

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

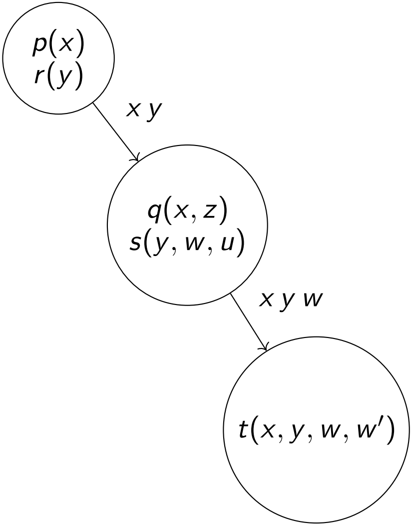


But is $\mathcal{R}$ Weakly Greedy?!

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(x) \to s(x, y, z)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$



Perhaps $\mathcal{R}$ is weakly greedy!

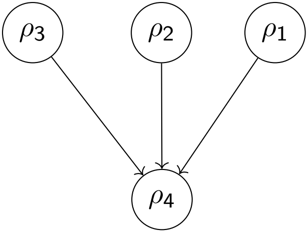
Showing $\mathcal{R}$ is weakly greedy

Idea 1: Use 3 to simulate 1 and 2.

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(y) \to s(y, u, v)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$



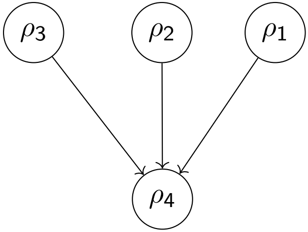
Showing $\mathcal{R}$ is weakly greedy

Idea 1: Use 3 to simulate 1 and 2.

$\mathcal{R} = \{$

1. $p(x) \rightarrow q(x, y)$
2. $r(y) \rightarrow s(y, u, v)$
3. $p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$



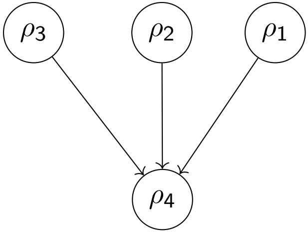
Showing $\mathcal{R}$ is weakly greedy

Idea 1: Use 3 to simulate 1 and 2.

$\mathcal{R} = \{$

1. $p(x) \rightarrow q(x, y)$
2. $r(y) \rightarrow s(y, u, v)$
3. $p(x) \wedge r(y) \rightarrow q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \rightarrow t(x, z, u, w)$

$\}$



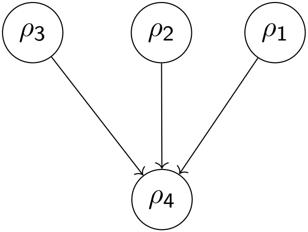
Showing $\mathcal{R}$ is weakly greedy

Idea 1: Use 3 to simulate 1 and 2.

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(y) \to s(y, u, v)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$



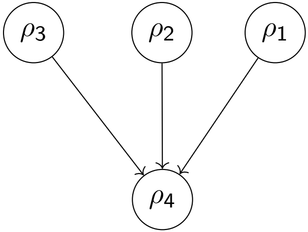
Showing $\mathcal{R}$ is weakly greedy

Idea 1: Use 3 to simulate 1 and 2.

$\mathcal{R} = \{$

1. $p(x) \to q(x, y)$
2. $r(y) \to s(y, u, v)$
3. $p(x) \wedge r(y) \to q(x, z) \wedge s(y, u, v)$
4. $q(x, y) \wedge s(z, u, v) \to t(x, z, u, w)$

$\}$

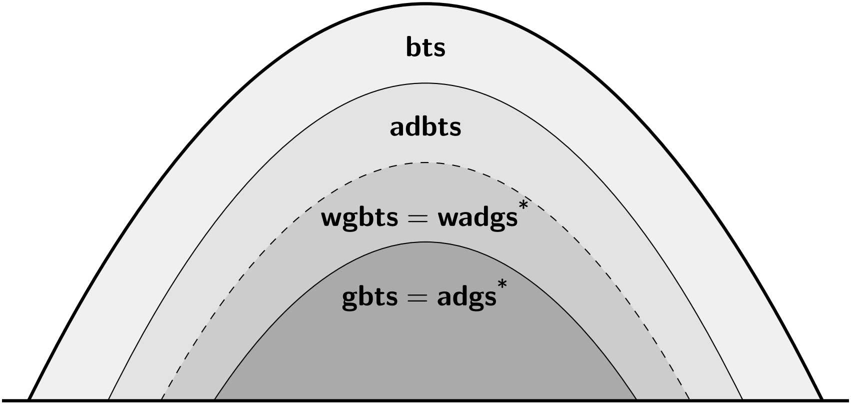


Idea 2: Permutation Lemma $\rightsquigarrow$

(i) Permute instances of 1 or 2 backward to instances of 2 or 1

(ii) Replace 1,2 or 2,1 instances with 3.

Conclusion



*Note: Proof must be checked in detail.