Tuple Generating Dependencies

Tomasz Gogacz

Zapytania koniunkcyjne (CQ)

$$Q(\bar{x}) = \exists \bar{y}\ \Phi(\bar{x},\ \bar{y})$$

Przykład: Istnieje ścieżka długości 3 między x a y

$$P_3(x, y) = \exists z_1, z_2\ E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, y)$$

W formacie SQL

SELECT edge1.x, edge3.y
FROM edges edge1, edges edge2, edges edge3
WHERE edge1.y=edge2.x AND edge2.y=edge3.x;

$$\forall\ \bar{x}\bar{y}\ \Phi(\bar{x}, \bar{y}) \ \rightarrow\ \exists \bar{z}\ \Psi(\bar{x}, \bar{z})$$

$$\forall\ \bar{x}\bar{y}\ \Phi(\bar{x}, \bar{y}) \ \rightarrow\ \exists \bar{z}\ \Psi(\bar{x}, \bar{z})$$

Drzewo genealogiczne

$$D = \left\{ \begin{array}{c} mother(cersei, joffrey),\quad father(robert, joffrey) \\ father(tywin, cersei),\quad father(tywin, jaime) \\ mother(cersei, tommen),\quad father(jaime, myrcella) \end{array} \right\}$$

$$T = \left\{ \begin{array}{c} \forall\, X\ human(X) \rightarrow \exists Z\ mother(Z, X) \wedge human(Z) \\ \forall\, X\ human(X) \rightarrow \exists Z\ father(Z, X) \wedge human(Z) \end{array} \right\}$$

Drzewo genealogiczne

$$D = \left\{ \begin{array}{c} mother(cersei, joffrey), \quad father(robert, joffrey) \\ father(tywin, cersei), \quad father(tywin, jaime) \\ mother(cersei, tommen), \quad father(jaime, myrcella) \end{array} \right\}$$

$$T = \left\{ \begin{array}{c} human(X) \rightarrow \exists Z\ mother(Z, X) \wedge human(Z) \\ human(X) \rightarrow \exists Z\ father(Z, X) \wedge human(Z) \end{array} \right\}$$

Drzewo genealogiczne

$$D = \left\{ \begin{array}{c} mother(cersei, joffrey), \quad father(robert, joffrey) \\ father(tywin, cersei), \quad father(tywin, jaime) \\ mother(cersei, tommen), \quad father(jaime, myrcella) \end{array} \right\}$$

$$T = \left\{ \begin{array}{c} human(X) \to \exists Z\ mother(Z, X) \wedge human(Z) \\ human(X) \to \exists Z\ father(Z, X) \wedge human(Z) \end{array} \right\}$$

Drzewo genealogiczne

$$D = \left\{ \begin{array}{c} mother(cersei, joffrey), \quad father(robert, joffrey) \\ father(tywin, cersei), \quad father(tywin, jaime) \\ mother(cersei, tommen), \quad father(jaime, myrcella) \end{array} \right\}$$

Wnioskowanie z wszystkich dostępnych informacji

Zamiast sprawdzać czy $D \models Q$ sprawdzamy $D,\ T \models Q$

$$T = \left\{ \begin{array}{c} human(X) \rightarrow \exists Z\ mother(Z, X) \wedge human(Z) \\ human(X) \rightarrow \exists Z\ father(Z, X) \wedge human(Z) \end{array} \right\}$$

Drzewo genealogiczne

$$D = \left\{ \begin{array}{c} mother(cersei, joffrey), \quad father(robert, joffrey) \\ father(tywin, cersei), \quad father(tywin, jaime) \\ mother(cersei, tommen), \quad father(jaime, myrcella) \end{array} \right\}$$

Wnioskowanie z wszystkich dostępnych informacji

Zamiast sprawdzać czy $D \models Q$ sprawdzamy $D,\ T \models Q$

Przykład

$has\_grandpa(joffrey) = \exists X, Y\ father(X, Y) \wedge father(Y, joffrey)$

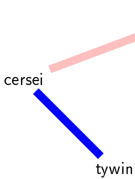
Istnienie uniwersalnej struktury

Dla dowolnego Q mamy: $D, T \models Q \Leftrightarrow chase(T, D) \models Q$.

Istnienie uniwersalnej struktury

Dla dowolnego Q mamy: $D, T \models Q \Leftrightarrow chase(T, D) \models Q$.

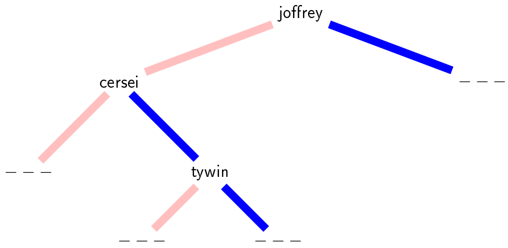
$$T = \left\{ \begin{array}{c} human(X) \rightarrow \exists Z \; mother(Z, X) \wedge human(Z) \\ human(X) \rightarrow \exists Z \; father(Z, X) \wedge human(Z) \end{array} \right\}$$



Istnienie uniwersalnej struktury

Dla dowolnego Q mamy: $D, T \models Q \Leftrightarrow chase(T, D) \models Q$.

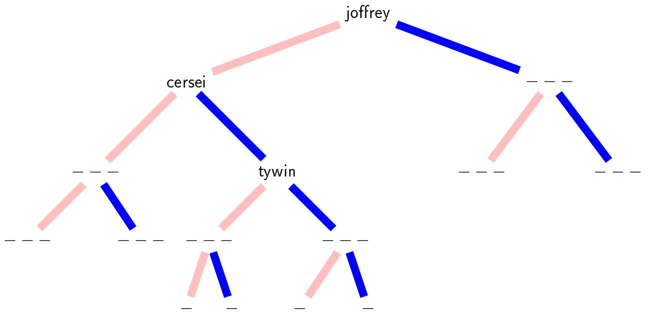
$$T = \left\{ \begin{array}{c} human(X) \rightarrow \exists Z\ mother(Z, X) \wedge human(Z) \\ human(X) \rightarrow \exists Z\ father(Z, X) \wedge human(Z) \end{array} \right\}$$



Istnienie uniwersalnej struktury

Dla dowolnego Q mamy: $D, T \models Q \Leftrightarrow chase(T, D) \models Q$.

$$T = \left\{ \begin{array}{c} human(X) \rightarrow \exists Z\ mother(Z, X) \wedge human(Z) \\ human(X) \rightarrow \exists Z\ father(Z, X) \wedge human(Z) \end{array} \right\}$$



Definicja Finite Controllability (FC)

T ma FC jeśli dla każdych D i Q zachodzi

$$D, T \models Q \qquad \Leftrightarrow \qquad D, T \models_{fin} Q$$

Definicja Finite Controllability (FC)

T ma FC jeśli dla każdych D i Q zachodzi

$$D, T \models Q \quad \Leftrightarrow \quad D, T \models_{fin} Q$$

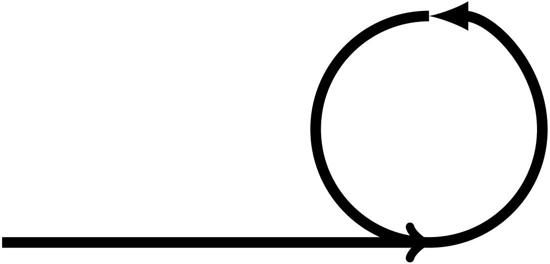
$$T = \left\{ \quad R(X, Y) \to \exists Z\ R(Y, Z) \quad \right\}$$

Definicja Finite Controllability (FC)

T ma FC jeśli dla każdych D i Q zachodzi

$$D, T \models Q \quad \Leftrightarrow \quad D, T \models_{fin} Q$$

$$T = \left\{ \quad R(X, Y) \rightarrow \exists Z \; R(Y, Z) \quad \right\}$$



Definicja Finite Controllability (FC)

T ma FC jeśli dla każdych D i Q zachodzi

$$D, T \models Q \qquad \Leftrightarrow \qquad D, T \models_{fin} Q$$

$$T = \left\{ \begin{array}{c} greater(X,Y) \rightarrow \exists Z\ greater(Y,Z) \\ greater(X,Y) \wedge greater(Y,Z) \rightarrow greater(X,Z) \end{array} \right\}$$

Definicja Finite Controllability (FC)

T ma FC jeśli dla każdych D i Q zachodzi

$$D, T \models Q \quad \Leftrightarrow \quad D, T \models_{fin} Q$$

$$T = \left\{ \begin{array}{c} greater(X, Y) \rightarrow \exists Z\ greater(Y, Z) \\ greater(X, Y) \wedge greater(Y, Z) \rightarrow greater(X, Z) \end{array} \right\}$$

$$Q = \exists X\ greater(X, X)$$


Is He Really The Best I Can Get?

Sticky Logic: definicja na przykładzie

$$r(Y, X) \wedge r(X, Y) \to \exists A, B\ s(X, A, B)$$

Sticky Logic: definicja na przykładzie

$$r(Y, X) \wedge r(X, Y) \rightarrow \exists A, B\ s(X, A, B)$$
$$s(X, Y; Z) \rightarrow \exists W\ s(X, Z, W)$$

Sticky Logic: definicja na przykładzie

$$r(Y, X) \wedge r(X, Y) \rightarrow \exists A, B\ s(X, A, B)$$
$$s(X, Y; Z) \rightarrow \exists W\ s(Y, X, W)$$

Sticky Logic: definicja na przykładzie

$$r(Y, {\color{blue}X}) \wedge r({\color{blue}X}, Y) \to \exists A, B\ s({\color{red}X}, A, B)$$
$$s({\color{red}X}, {\color{red}Y}; Z) \to \exists W\ s({\color{red}Y}, {\color{red}X}, W)$$

Twierdzenie 1

Sticky Logic ma własność Finite Controllability

Definicja własności Bounded Derivation Depth (BDD)

T ma własność BDD jeśli: $\forall Q \;\; \exists n_Q \in \mathbb{N} \;\; \forall D$

$$Chase(T, D) \models Q \;\;\Leftrightarrow\;\; Chase(T, D)^{n_Q} \models Q$$

Definicja własności Bounded Derivation Depth (BDD)

T ma własność BDD jeśli: $\forall Q \ \ \exists n_Q \in \mathbb{N} \ \ \forall D$

$$Chase(T, D) \models Q \ \ \Leftrightarrow \ \ Chase(T, D)^{n_Q} \models Q$$

$$T = \{ \quad R(X, Y) \to \exists Z\ R(Y, Z) \quad \}$$

Definicja własności Bounded Derivation Depth (BDD)

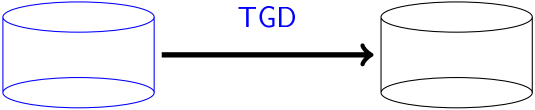
T ma własność BDD jeśli: $\forall Q \;\; \exists n_Q \in \mathbb{N} \;\; \forall D$

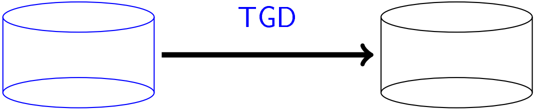
$$Chase(T, D) \models Q \;\; \Leftrightarrow \;\; Chase(T, D)^{n_Q} \models Q$$

$$T = \left\{ greater(X, Y) \land greater(Y, Z) \rightarrow greater(X, Z) \right\}$$

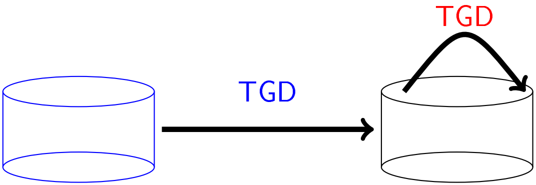
Twierdzenie 2

Nad binarną sygnaturą BDD implikuje FC


TGD

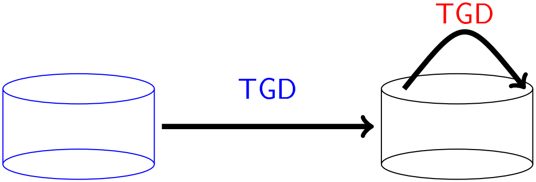


$$T = \{ \quad r(X,Y) \wedge r(Y,Z) \rightarrow \exists A\, s(Y,A) \quad \}$$



$$T = \{ \quad r(X,Y) \wedge r(Y,Z) \to \exists A\, s(Y,A) \quad \}$$

$$T = \{ \quad s(X,Y) \to \exists Z\, s(Y,Z) \quad \}$$



$$T = \{ \quad r(X,Y) \wedge r(Y,Z) \rightarrow \exists A\, s(Y,A) \quad \}$$

$$T = \{ \quad s(X,Y) \rightarrow \exists Z\, s(Y,Z) \quad \}$$

Problem terminacji Chase

- **Input:** T - Zbiór TGD
- **Output:** Czy dla każdej struktury D proces $Chase(T,D)$ kiedyś się kończy?

Problem terminacji Chase'a

- **Input:** T - Zbiór TGD
- **Output:** Czy dla każdej struktury D proces $Chase(T, D)$ kiedyś się kończy?

Twierdzenie 3

Problem terminacji Chase'a jest nierozstrzygalny.

Definicja determinacji (nieformalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli zapytania $Q_1, \ldots, Q_n$ zawsze dostarczają dość informacji żeby odpowiedzieć na pytanie Ψ

Definicja determinacji (nieformalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli zapytania $Q_1, \ldots, Q_n$ zawsze dostarczają dość informacji żeby odpowiedzieć na pytanie Ψ

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\Rightarrow\; \Psi(D_1) = \Psi(D_2)$$

Definicja determinacji (nieformalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli zapytania $Q_1, \ldots, Q_n$ zawsze dostarczają dość informacji żeby odpowiedzieć na pytanie Ψ

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\; \Rightarrow \;\; \Psi(D_1) = \Psi(D_2)$$

$\{P_3, P_4\} \twoheadrightarrow P_7 \qquad P_7(x, y) = \exists z \; P_3(x, z) \wedge P_4(z, y)$

Definicja determinacji (nieformalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli zapytania $Q_1, \ldots, Q_n$ zawsze dostarczają dość informacji żeby odpowiedzieć na pytanie Ψ

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \ \ Q_i(D_1) = Q_i(D_2)) \ \Rightarrow \ \Psi(D_1) = \Psi(D_2)$$

$\{P_3, P_4\} \twoheadrightarrow P_7$ $\qquad P_7(x, y) = \exists z \; P_3(x, z) \wedge P_4(z, y)$

$\{P_3, P_4\} \not\twoheadrightarrow P_2$ $\qquad D_1 = \emptyset \quad D_2 = \{E(a, b), E(b, c)\}$

Definicja determinacji (nieformalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli zapytania $Q_1, \ldots, Q_n$ zawsze dostarczają dość informacji żeby odpowiedzieć na pytanie Ψ

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\; \Rightarrow \;\; \Psi(D_1) = \Psi(D_2)$$

$\{P_3, P_4\} \twoheadrightarrow P_7$ $\qquad P_7(x, y) = \exists z \; P_3(x, z) \wedge P_4(z, y)$

$\{P_3, P_4\} \not\twoheadrightarrow P_2$ $\qquad D_1 = \emptyset \quad D_2 = \{E(a, b), E(b, c)\}$

$\{P_3, P_4\} \overset{?}{\twoheadrightarrow} P_5$

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\; \Rightarrow \;\; \Psi(D_1) = \Psi(D_2)$$

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\; \Rightarrow \;\; \Psi(D_1) = \Psi(D_2)$$

Aksjomatyzacja braku kontrprzykładu na determinację

- zamiast dwóch struktur D_1, D_2 mamy jedną $\color{red}{D} \cup \color{blue}{D}$

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\; \Rightarrow \;\; \Psi(D_1) = \Psi(D_2)$$

Aksjomatyzacja braku kontrprzykładu na determinację

- zamiast dwóch struktur D_1, D_2 mamy jedną $\color{red}{D} \cup \color{blue}{D}$
- $\color{red}{Q_i}(\bar{X}, \bar{Y}) \;\Rightarrow\; \exists \bar{Z}\, \color{blue}{Q_i}(\bar{X}, \bar{Z})$
  $\color{blue}{Q_i}(\bar{X}, \bar{Y}) \;\Rightarrow\; \exists \bar{Z}\, \color{red}{Q_i}(\bar{X}, \bar{Z})$

Definicja determinacji (formalna)

Mówimy, że zbiór zapytań $Q_1, \ldots, Q_n$ determinuje zapytanie Ψ jeśli, dla każdej pary baz danych D_1, D_2 mamy:

$$(\forall i \;\; Q_i(D_1) = Q_i(D_2)) \;\; \Rightarrow \;\; \Psi(D_1) = \Psi(D_2)$$

Aksjomatyzacja braku kontrprzykładu na determinację

- zamiast dwóch struktur D_1, D_2 mamy jedną $\color{red}{D} \cup \color{blue}{D}$
- $\color{red}{Q_i}(\bar{X}, \bar{Y}) \;\Rightarrow\; \exists \bar{Z} \color{blue}{Q_i}(\bar{X}, \bar{Z})$
  $\color{blue}{Q_i}(\bar{X}, \bar{Y}) \;\Rightarrow\; \exists \bar{Z} \color{red}{Q_i}(\bar{X}, \bar{Z})$
- $Q_1, \ldots, Q_n \twoheadrightarrow \Psi \;\;\Leftrightarrow\;\; \color{blue}{\Psi}, \mathcal{T} \models \color{red}{\Psi}$

Problem determinacji dla zapytań koniunkcyjnych

- **Input:** $Q_1, \ldots, Q_n$, oraz Ψ
- **Output:** Czy $Q_1, \ldots, Q_n \twoheadrightarrow \Psi$?

Twierdzenie 4

Problem determinacji dla zapytań koniunkcyjnych jest nierozstrzygalny

Twierdzenie 1

Sticky Logic ma własność FC

Tomasz Gogacz, Jerzy Marcinkowski:
Converging to the Chase - A Tool for Finite Controllability. LICS 2013: 540-549

Twierdzenie 2

Nad binarną sygnaturą BDD implikuje FC

Tomasz Gogacz, Jerzy Marcinkowski:
On the BDD/FC conjecture. PODS 2013: 127-138

Twierdzenie 3

Problem terminacji Chase'a jest nierozstrzygalny.

Tomasz Gogacz, Jerzy Marcinkowski:
All-Instances Termination of Chase is Undecidable. ICALP (2) 2014: 293-304

Twierdzenie 4

Problem determinacji dla zapytań koniunkcyjnych jest nierozstrzygalny

Tomasz Gogacz, Jerzy Marcinkowski:
The Hunt for a Red Spider: Conjunctive Query Determinacy Is Undecidable. LICS 2015: 281-292