Converging to the Chase and Some Finite Controllability Results

Tomasz Gogacz and Jerzy Marcinkowski

Institute of Computer Science, Wrocław University

May 2012

Disclaimer.

This talk may contain oversimplifications.

TGDs, Chase, Finite controllability and BDD..

- Tuple generating dependency:
  $FlightNumber(x) \wedge Departs(x, y) \Rightarrow \exists z\ Lands(x, z)$
- $Chase_0(D, \mathcal{T}) := D$
  $Chase_{i+1}(D, \mathcal{T}) := Chase_i(D, \mathcal{T}) \cup \{\text{a witness for each } TGD \in \mathcal{T}\}$
  $Chase := \bigcup_{i \in \mathbb{N}} Chase_{i+1}$
- Theorem: Chase is the universal structure. Let Φ be a UCQ. Then $D, \mathcal{T} \models \Phi$ iff $Chase(D, \mathcal{T}) \models \Phi$
- Definition (of Finite Controllability): $\mathcal{T}$ has FC if:
  $\forall \Phi, D\ \exists \text{ finite } M \quad Chase(D, \mathcal{T}) \not\models \Phi \ \Rightarrow\ M \models D, \mathcal{T}, \neg\Phi$
- Definition (of Bounded Derivation Depth property): $\mathcal{T}$ has BDD if:
  $\forall \Phi\ \exists n\ \forall D \quad Chase(D, \mathcal{T}) \models \Phi \ \Rightarrow\ Chase_n(D, \mathcal{T}) \models \Phi.$

Joinless Logic and Sticky Datalog$^{\exists}$.

Chase **is like the mankind:**

Some elements are born:
$P(\bar{x}) \wedge Q(\bar{y}) \Rightarrow \exists w\ R(w, \bar{x}, \bar{y})$

and some are projected out:
$P(\bar{x}, \bar{y}) \Rightarrow R(\bar{x})$

Joinless Logic and Sticky Datalog$^{\exists}$.

Chase **is like the mankind:**

Some elements are born:
$P(\bar{x}) \wedge Q(\bar{y}) \Rightarrow \exists w\ R(w, \bar{x}, \bar{y})$ parenthood rule

and some are projected out:
$P(\bar{x}, \bar{y}) \Rightarrow R(\bar{x})$ projection rule

Joinless Logic and Sticky Datalog$^{\exists}$.

Chase **is like the mankind:**

Some elements are born:
$P(\bar{x}) \wedge Q(\bar{y}) \Rightarrow \exists w\ R(w, \bar{x}, \bar{y})$ parenthood rule

and some are projected out:
$P(\bar{x}, \bar{y}) \Rightarrow R(\bar{x})$ projection rule

Joinless logic: variables in $\bar{x}, \bar{y}$ are pairwise distinct.

Sticky Datalog$^{\exists}$: if a variable occurs more than once in $\bar{x}, \bar{y}$ then it gets immortal – never can be projected out.

From Joinless Logic to Sticky Datalog$^{\exists}$. The BDD $\Rightarrow$ FC conjecture.

Theorem 1: Sticky Datalog$^{\exists}$ programs are FC.

Theorem 2: Joinless Logic programs are FC.

Remark: Both the logics are easily seen to be BDD.

From Joinless Logic to Sticky Datalog$^{\exists}$. The BDD $\Rightarrow$ FC conjecture.

Theorem 1: Sticky Datalog$^{\exists}$ programs are FC.

Theorem 2: Joinless Logic programs are FC.

Remark: Both the logics are easily seen to be BDD.

Proof of Theorem 1 (outline): Induction on the number of immortal variables. The base case: Theorem 2. This is the hard part.

From Joinless Logic to Sticky Datalog$^{\exists}$. The BDD $\Rightarrow$ FC conjecture.

Theorem 1: Sticky Datalog$^{\exists}$ programs are FC.

Theorem 2: Joinless Logic programs are FC.

Remark: Both the logics are easily seen to be BDD.

Proof of Theorem 1 (outline): Induction on the number of immortal variables. The base case: Theorem 2. This is the hard part.

The assumptions about Sticky Datalog$^{\exists}$ are used 3 times. In this number 2 times we only use the fact that Sticky Datalog$^{\exists}$ is BDD.
This motivates:

Conjecture: For each $\mathcal{T}$ if $\mathcal{T}$ is BDD then $\mathcal{T}$ is FC.

From Joinless Logic to Sticky Datalog$^{\exists}$. The BDD $\Rightarrow$ FC conjecture.

Theorem 1: Sticky Datalog$^{\exists}$ programs are FC.

Theorem 2: Joinless Logic programs are FC.

Remark: Both the logics are easily seen to be BDD.

Proof of Theorem 1 (outline): Induction on the number of immortal variables. The base case: Theorem 2. This is the hard part.

The assumptions about Sticky Datalog$^{\exists}$ are used 3 times. In this number 2 times we only use the fact that Sticky Datalog$^{\exists}$ is BDD.
This motivates:

Conjecture: For each $\mathcal{T}$ if $\mathcal{T}$ is BDD then $\mathcal{T}$ is FC.

The rest of the talk is about the proof of Theorem 2

Joinless Logic is FC. The structure of our proof.

- We assume that $\mathcal{T}$ respects family patterns. Observation: This can be done for free.

Joinless Logic is FC. The structure of our proof.

- We assume that $\mathcal{T}$ respects family patterns. Observation: This can be done for free.
- We construct a sequence M_n of finite structures, such that for each UCQ Φ if $M_n \not\models \Phi$ then $M_{n+1} \not\models \Phi$.
- Definition: Φ is M-true if $M_n \models \Phi$ for all n. We want to show that M_n ,,converges" to Chase – **if Φ is M-true then** $Chase \models \Phi$ **(*)**
- 1st Little Trick: If some UCQ is M-true then one of its disjuncts is M-true. So it is enough to talk about CQs.

Joinless Logic is FC. The structure of our proof.

- We assume that $\mathcal{T}$ respects family patterns. Observation: This can be done for free.
- We construct a sequence M_n of finite structures, such that for each UCQ Φ if $M_n \not\models \Phi$ then $M_{n+1} \not\models \Phi$.
- Definition: Φ is M-true if $M_n \models \Phi$ for all n. We want to show that M_n ,,converges" to Chase – **if Φ is M-true then** $Chase \models \Phi$ **(*)**
- 1st Little Trick: If some UCQ is M-true then one of its disjuncts is M-true. So it is enough to talk about CQs.
- 2nd Little Trick: If a CQ Φ is M-true and if condition(Φ) then there exists σ such that $\sigma(\Phi)$ is also M-true. So it is enough to prove **(*)** for $\sigma(\Phi)$.

Joinless Logic is FC. The structure of our proof.

- We assume that $\mathcal{T}$ respects family patterns. Observation: This can be done for free.
- We construct a sequence M_n of finite structures, such that for each UCQ Φ if $M_n \not\models \Phi$ then $M_{n+1} \not\models \Phi$.
- Definition: Φ is M-true if $M_n \models \Phi$ for all n. We want to show that M_n ,,converges" to Chase – **if Φ is M-true then** *Chase* $\models \Phi$ **(*)**
- 1st Little Trick: If some UCQ is M-true then one of its disjuncts is M-true. So it is enough to talk about CQs.
- 2nd Little Trick: If a CQ Φ is M-true and if condition(Φ) then there exists σ such that $\sigma(\Phi)$ is also M-true. So it is enough to prove **(*)** for $\sigma(\Phi)$.
- Lemma: If $\neg$condition(Φ) (we say that such Φ is in ,,normal form") and if $M_1 \models \Phi$ **then** *Chase* $\models \Phi$**.**

Family patterns – introduction

- As we said, *Chase* is like the mankind.
- We think that atoms are families. We want to know more about the structure of these families.

Family patterns – introduction

- As we said, *Chase* is like the mankind.
- We think that atoms are families. We want to know more about the structure of these families.
- $\mathcal{T}$ *respects family patterns* if each predicate Q has, as a part of its name, two parameters, F and γ, where:
  - $F \subseteq \{1, 2, \ldots \mathit{arity}(Q)\}^2$ is a tree (or a forest)
  - γ is a partial function: $F \to \{1, 2, \ldots, \mathit{maxarity}\}$

  and some soundness conditions are satisfied.

  (do not give up yet! it is going to be quite intuitive).

Family patterns – the meaning of F and γ

The idea is that $F \subseteq \{1, 2, \ldots \mathit{arity}(Q)\}^2$ is the ancestor relation.

If *alice* lives (in one atom) with her parents then:
$\{\mathit{alice}\ F\ \mathit{father},\ \mathit{alice}\ F\ \mathit{mother}\}$.

Family patterns – the meaning of F and γ

The idea is that $F \subseteq \{1, 2, \ldots arity(Q)\}^2$ is the ancestor relation.

If *alice* lives (in one atom) with her parents then:
$\{alice\ F\ father,\ alice\ F\ mother\}$.

But the ancestor relation is not all we need to know. The families:


$\{alice\ F\ mother\}$


$\{alice\ F\ grandmother\}$

look the same from the ancestor relation point of view,
but they represent different family patterns.

Family patterns – the meaning of F and γ

The idea is that $F \subseteq \{1, 2, \ldots arity(Q)\}^2$ is the ancestor relation.

If *alice* lives (in one atom) with her parents then:
$\{alice\ F\ father,\ alice\ F\ mother\}$.

But the ancestor relation is not all we need to know. The families:


$\{alice\ F\ mother\}$


$\{alice\ F\ grandmother\}$

look the same from the ancestor relation point of view,
but they represent different family patterns.

Here is where γ is comming:

γ is a partial function: $F \to \{1, 2, \ldots, maxarity\}$

If aFb then $\gamma(a, b)$ tells us how a addresses b.

Family patterns – introduction

- As we said, *Chase* is like the mankind.
- We think that atoms are families. We want to know more about the structure of these families.
- $\mathcal{T}$ *respects family patterns* if each predicate Q has, as a part of its name, two parameters, F and γ, where:
  - $F \subseteq \{1, 2, \ldots \mathit{arity}(Q)\}^2$ is a tree (or a forest)
  - γ is a partial function: $F \to \{1, 2, \ldots, \mathit{maxarity}\}$

  and some soundness conditions are satisfied.

  (do not give up yet! it is going to be quite intuitive).

Family patterns – the soundness conditions on $\mathcal{T}$

- The case of a projection rule in $\mathcal{T}$:

$$R_{F,\delta}(\bar{x}) \Rightarrow P_{G,\gamma}(\bar{y}) \qquad (\bar{y} \subset \bar{x})$$

Then $G = F \cap \bar{y}^2$ and $\gamma = \delta \restriction G$.

The fact that someone is projected out
neither changes the ancestor relation on the surviving elements,
nor the way they address each other.

Family patterns – the soundness conditions on $\mathcal{T}$

- The case of a parenthood rule in $\mathcal{T}$:

  $R_{F,\delta}(\bar{x}) \wedge R_{F',\delta'}(\bar{y}) \Rightarrow \exists w\ P_{G,\gamma}(\bar{x}, \bar{y}, w) \qquad |\bar{x}| = k$

  Then: $G = F \cup F' \cup \{\langle w, u\rangle :\ u \in \bar{x} \cup \bar{y}\}$ and $\gamma = \delta \cup \delta' \cup \gamma_0$

  where: $\gamma_0(w, x_i) = i$ and $\gamma_0(w, y_i) = i + k$

  A newborn element learns to address its ancestors by their position in the family (at the moment of its birth).

  The birth of a new element
  neither changes the ancestor relation on its ancestors,
  nor the way they address each other.

Joinless Logic is FC. The structure of our proof.

- We assume that $\mathcal{T}$ respects family patterns. Observation: This can be done for free.
- We construct a sequence M_n of finite structures, such that for each UCQ Φ if $M_n \not\models \Phi$ then $M_{n+1} \not\models \Phi$.
- Definition: Φ is M-true if $M_n \models \Phi$ for all n. We want to show that M_n ,,converges" to Chase – **if Φ is M-true then** $Chase \models \Phi$ **(*)**
- 1st Little Trick: If some UCQ is M-true then one of its disjuncts is M-true. So it is enough to talk about CQs.
- 2nd Little Trick: If a CQ Φ is M-true and if condition(Φ) then there exists σ such that $\sigma(\Phi)$ is also M-true. So it is enough to prove **(*)** for $\sigma(\Phi)$.
- Lemma: If $\neg$condition(Φ) (we say that such Φ is in ,,normal form") and if $M_1 \models \Phi$ **then** $Chase \models \Phi$**.**

The canonical finite models M_n.

- Let $a, b \in \mathit{Chase}$. Then $a \equiv_1 b$ if a and b were born with the same rule.
- $a \equiv_{n+1} b$ if $a \equiv_1 b$ and $c \equiv_n d$ for each pair c, d of respective parents of a and b.
- $M_n = \mathit{Chase}/\equiv_n$. The relations are defined, as minimal relations such that the quotient mapping is a homomorphism.

The canonical finite models M_n.

- Let $a, b \in Chase$. Then $a \equiv_1 b$ if a and b were born with the same rule.
- $a \equiv_{n+1} b$ if $a \equiv_1 b$ and $c \equiv_n d$ for each pair c, d of respective parents of a and b.
- $M_n = Chase/\equiv_n$. The relations are defined, as minimal relations such that the quotient mapping is a homomorphism.

Remark. M_n is not always a model of $\mathcal{T}$. But for Joinless Logic and for Guarded TGDs it is. For Sticky Datalog$^{\exists}$ it is not.

The canonical finite models M_n.

- Let $a, b \in \mathit{Chase}$. Then $a \equiv_1 b$ if a and b were born with the same rule.
- $a \equiv_{n+1} b$ if $a \equiv_1 b$ and $c \equiv_n d$ for each pair c, d of respective parents of a and b.
- $M_n = \mathit{Chase}/\equiv_n$. The relations are defined, as minimal relations such that the quotient mapping is a homomorphism.

Remark. M_n is not always a model of $\mathcal{T}$. But for Joinless Logic and for Guarded TGDs it is. For Sticky Datalog$^{\exists}$ it is not.

Remark. As promised, the sequence M_n is ,,monotonic'':
for each UCQ Φ if $M_n \not\models \Phi$ then $M_{n+1} \not\models \Phi$.

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

Observation: Let $\sigma = [w/y]$. If $\mathit{Chase} \models \Phi$ then $\mathit{Chase} \models \sigma(\Phi)$.

Problem: It is not always true in finite structures.

Parents are unique in Chase, but not in finite structures.

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

Observation: Let $\sigma = [w/y]$. If $\mathit{Chase} \models \Phi$ then $\mathit{Chase} \models \sigma(\Phi)$.

Problem: It is not always true in finite structures.

Parents are unique in Chase, but not in finite structures.

2nd Little Trick: if Φ is M-true then $\sigma(\Phi)$ is M-true.

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

Observation: Let $\sigma = [w/y]$. If $\mathit{Chase} \models \Phi$ then $\mathit{Chase} \models \sigma(\Phi)$.

Problem: It is not always true in finite structures.

Parents are unique in Chase, but not in finite structures.

2nd Little Trick: if Φ is M-true then $\sigma(\Phi)$ is M-true.

Proof: We need to show that $M_n \models \sigma(\Phi)$. Suppose Φ is M-true. So $M_{n+1} \models \Phi$,

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

Observation: Let $\sigma = [w/y]$. If $\mathit{Chase} \models \Phi$ then $\mathit{Chase} \models \sigma(\Phi)$.

Problem: It is not always true in finite structures.

Parents are unique in Chase, but not in finite structures.

2nd Little Trick: if Φ is M-true then $\sigma(\Phi)$ is M-true.

Proof: We need to show that $M_n \models \sigma(\Phi)$. Suppose Φ is M-true. So $M_{n+1} \models \Phi$, by valuation v. The elements $v(y)$ and $v(w)$ come from elements of *Chase* that are respective parents of $v(x)$.

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

Observation: Let $\sigma = [w/y]$. If $\mathit{Chase} \models \Phi$ then $\mathit{Chase} \models \sigma(\Phi)$.

Problem: It is not always true in finite structures.

Parents are unique in Chase, but not in finite structures.

2nd Little Trick: if Φ is M-true then $\sigma(\Phi)$ is M-true.

Proof: We need to show that $M_n \models \sigma(\Phi)$. Suppose Φ is M-true. So $M_{n+1} \models \Phi$, by valuation v. The elements $v(y)$ and $v(w)$ come from elements of *Chase* that are respective parents of $v(x)$. So maybe it is not true that $v(w) \equiv_{n+1} v(y)$ but we can be sure that $v(w) \equiv_n v(y)$.

2nd Little Trick – normalization of queries.

Suppose $\mathcal{T}$ satisfies family patterns. Consider a query:

$\Phi = P(\ldots x, \ldots y \ldots) \wedge Q(\ldots x, \ldots w \ldots) \wedge \phi$

such that $\gamma_P(x, y) = \gamma_Q(x, w)$

(y is the same parent of x in P as w in Q)

Observation: Let $\sigma = [w/y]$. If $\mathit{Chase} \models \Phi$ then $\mathit{Chase} \models \sigma(\Phi)$.

Problem: It is not always true in finite structures.

Parents are unique in Chase, but not in finite structures.

2nd Little Trick: if Φ is M-true then $\sigma(\Phi)$ is M-true.

Proof: We need to show that $M_n \models \sigma(\Phi)$. Suppose Φ is M-true. So $M_{n+1} \models \Phi$, by valuation v. The elements $v(y)$ and $v(w)$ come from elements of *Chase* that are respective parents of $v(x)$. So maybe it is not true that $v(w) \equiv_{n+1} v(y)$ but we can be sure that $v(w) \equiv_n v(y)$. Magic: this means that $M_n \models \sigma(\Phi)$!

Joinless Logic is FC. The structure of our proof.

- We assume that $\mathcal{T}$ respects family patterns. Observation: This can be done for free.
- We construct a sequence M_n of finite structures, such that for each UCQ Φ if $M_n \not\models \Phi$ then $M_{n+1} \not\models \Phi$.
- Definition: Φ is M-true if $M_n \models \Phi$ for all n. We want to show that M_n ,,converges" to Chase – **if Φ is M-true then** $Chase \models \Phi$ **(*)**
- 1st Little Trick: If some UCQ is M-true then one of its disjuncts is M-true. So it is enough to talk about CQs.
- 2nd Little Trick: If a CQ Φ is M-true and if condition(Φ) then there exists σ such that $\sigma(\Phi)$ is also M-true. So it is enough to prove **(*)** for $\sigma(\Phi)$.
- Lemma: If $\neg$condition(Φ) (we say that such Φ is in ,,normal form") and if $M_1 \models \Phi$ **then** $Chase \models \Phi$**.**