L3 Internship:
Implementing the Core Chase for **Horn-$\mathcal{ALCH}$**

Maël Abily
Computer Science Department
ENS de Lyon

Supervised in Montpellier from 31/05/2021 to 09/07/2021 by:
Jean-François Baget and David Carral
Permanent researchers at INRIA

Contents

1 Introduction **2**

2 Preliminaries **3**

3 The Chase **5**

4 The Merge Chase **8**

5 Conclusion **14**

1 Introduction

During my studies at ENS de Lyon, I did a research internship in computer science for six weeks. I did this internship at INRIA, in the GRAPHIK team (GRAPHIK means Graphs for Inferences on Knowledge), supervised by Jean-François Baget and David Carral. I thank them for the time they took with me for these six weeks to help me in my research and answer all the questions I had. I also thank the whole GRAPHIK team which welcomed and integrated me for this internship. GRAPHIK's research work focuses on databases thanks to knowledge representation and reasoning (knowledge means set of data). The subject of my internship, entitled "Implementing the Core Chase for the Description Logic ALC", looks for an algorithm that reasons on a specific type of data to answer an important problem in database theory.

Knowledge representation and reasoning is the field of artificial intelligence dedicated to representing information about the world in a form that a computer system can use to solve complex tasks such as diagnosing a medical condition or having a dialog in a natural language. The conjunctive query entailment is an important issue in knowledge representation and reasoning. This problem can be described in a first order logic background. We work on conjunctive formulas that are formulas constructed only with conjunction and existential quantification; and boolean conjunctive queries (BCQ) that are conjunctive formulas without free variables. The answer of a BCQ is either yes or no. For example, the formula $\exists y.\mathit{Human}(y) \wedge \mathit{IsTheBrotherOf}(\mathbf{Pierre}, y)$ is a BCQ asking if Pierre has a human brother. We also work on existential rules that are formulas of the form $\forall \vec{x}.\forall \vec{y}.(A(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.B(\vec{x}, \vec{z}))$ where $\vec{x}$, $\vec{y}$, and $\vec{z}$ represent tuples of variables. A knowledge base K will then be a set of existential rules and conjunctive formulas. Note that in practice, a lot of formulas representing data can be expressed via conjunctive formulas and existential rules. We can now define the problem of conjunctive query entailment: Given a knowledge base K and a BCQ Q determine if K entails the query Q. More precisely, the input is the knowledge base K and the BCQ Q, and the output is yes if and only if K entails Q under standard first order logic semantics. For example, the knowledge base $K = \{\mathit{Mother}(\mathbf{Marie}), \forall x.\mathit{Mother}(x) \rightarrow \mathit{Parent}(x), \forall x.(\mathit{Parent}(x) \rightarrow \exists y.\mathit{IsTheParentOf}(x, y))\}$ entails the query $Q = \exists x.\mathit{IsTheParentOf}(\mathbf{Marie}, x)$. We usually use reasoning algorithms to answer the conjunctive query entailment. Note that the problem of conjunctive query entailment is undecidable ([2], theorem 4).

A knowledge base K entails a BCQ Q if every model of K is a model of Q. However, we cannot compute all models because K can have an infinite number of them. To deal with this problem, we can compute an universal model of the knowledge base K. That is a model of K that is entailed by all the models of K. If a such model U exists, we just need to show that U entails Q to conclude that K entails Q. Hence, to solve query entailment, we just have to compute a finite universal model U for a given input K and check if U entails Q. To compute these models, we can use an algorithm called the chase. In this document, we present several variants of this algorithm: the oblivious chase, the restricted chase and then the core chase. The first two chase are defined in [3] and the last chase is defined in [4]. The chase can be seen as a two-step process. First, it repeatedly applies rules to the initial set of conjunctive formulas (and the core chase sometimes removes redundant conjunctive formulas). Then it looks for an answer to the BCQ in this saturated set of conjunctive formulas.

The third chase is the best in the sense that it is the only one that terminates if and only if there exists a finite universal model. To remove redundant conjunctive formulas, the core chase computes the core of a set of conjunctive formulas (that is the smallest subset logically stronger). Nevertheless, in order to compute

the core, we have to look for global endomorphisms in the set of conjunctive formulas produced by the core chase. The core chase is then more complicated to implement since global endomorphisms are hard to compute. We will then focus on a restricted type of knowledge base (that is called **Horn-$\mathcal{ALCH}$** knowledge base) where the initial set of conjunctive formulas is ground (that is there is no free variable in the formula) and where the rules are **Horn-$\mathcal{ALCH}$** axioms. **Horn-$\mathcal{ALCH}$** is defined in [5]. It is a part of description logics that is widely used in the context of knowledge reasoning. Description logics are parts of first-order logic with unary and binary predicates, and are the main tools for reasoning about knowledge bases. Note that description logics are undecidable whereas **Horn-$\mathcal{ALCH}$** is decidable.

We present a new variant of the chase for **Horn-$\mathcal{ALCH}$** knowledge bases which is also guaranteed to terminate if the input knowledge base admits a finite universal model and is much easier to implement. In the core chase, one has to look for global endomorphisms in the sets of facts produced by the chase whereas in the new variant, we only need to look for very local endomorphisms, which are much simpler to compute.

First of all, in Section 2 and 3, we will define the notions that will be useful for this paper and present some well known properties. Then, we will move on to our contributions in Section 4

2 Preliminaries

We consider a set of variables **Vars**, a set of constants **Csts**, and a set of predicates **Preds**, that are pairwise disjoint. A *term* is a variable or a constant. We call **Terms** the set of terms. If $t_1, \ldots, t_n$ are terms and P is a predicate of arity n, then $P(t_1, \ldots, t_n)$ is a *fact*. The fact $P(t_1, \ldots, t_n)$ is *ground* if $t_1, \ldots, t_n$ are constants. We can now define a factbase that is a notion omnipresent in all the paper.

Definition 2.1. A *factbase* F is an existentially closed conjunction of facts that is, a formula that does not contain occurrences of free variables and is of the form

$$\exists x_1, \ldots, x_n . P_1(t_1^1, \ldots, t_{k_1}^1) \wedge \ldots \wedge P_m(t_1^m, \ldots, t_{k_m}^m)$$

where t_i^j are terms and P_i are predicates. A factbase is *ground* if each of its facts is ground.

Note that we do consider factbases that are not ground, unlike other authors. Consequently, the notions of boolean conjunctive queries and factbases coincide. For convenience, we identify factbases as sets of facts, which allows us to use operators from set theory. For example, we identify the factbase $\exists x, y, z . P(x) \wedge Q(x, a) \wedge R(y, z, b)$ with the set of facts $\{P(x), Q(x, a), R(y, z, b)\}$ where P, Q, R are predicates, a, b are constants, and x, y, z are variables. For a formula A, let $\boldsymbol{Vars}(A)$, $\boldsymbol{Csts}(A)$, and $\boldsymbol{Terms}(A)$ be the sets of variables, constants, and terms that occur in A, respectively.

A *substitution* $\sigma : X \to \mathbf{Terms}$ is a function where X is a set of variables. For example $\{x \mapsto z, y \mapsto a\}$ is a substitution from $\{x, y\}$ to **Terms**. By extension:

- if $c \in \mathbf{Csts}$, then $\sigma(c) = c$,
- if $x \in \mathbf{Vars} \setminus X$, $\sigma(x) = x$,
- if $f = P(t_1, \ldots, t_n)$ is a fact, then $\sigma(f) = P(\sigma(t_1), \ldots, \sigma(t_n))$, and
- if $F = \{f_1, \ldots, f_n\}$ is a factbase, then $\sigma(F) = \{\sigma(f_1), \ldots, \sigma(f_n)\}$.

This definition leads us to the central notion of homomorphism.

Definition 2.2. For two factbases F and G, a *homomorphism* from F to G is a substitution $\sigma : \mathbf{Vars}(F) \to \mathbf{Terms}$ where $\sigma(F) \subseteq G$.

For a factbase F, let id_F be the substitution mapping each variable in $\mathbf{Vars}(F)$ to itself. An *isomorphism* from F to G is a homomorphism h such that there exists a homomorphism g from G to F verifying $h \circ g = id_G$ and $g \circ h = id_F$. For the remainder of this paper, we identify sets of facts that are unique up to isomorphism.

Definition 2.3. A factbase F *entails* another factbase G (often noted $F \vDash G$) if each interpretation satisfying F satisfies G. F is *equivalent* to G if $F \vDash G$ and $G \vDash F$.

In this document, it is not necessary to know exactly the definition of interpretation because in our fragment of first-order logic, we have a convenient characterization. For example, to know if $F \vDash G$, we use the following theorem.

Theorem 2.1 (Homomorphism Theorem)**.** A factbase F *entails* another factbase Q if and only if there exists a homomorphism from Q to F.

The previous theorem has been proved in ([1], Theorem 6.2.3). Intuitively, if $F \vDash G$, then F is logically stronger than G and from F, we can deduce G. For example, the factbase $F = \{P(b, a), A(x)\}$ entails the factbase $Q = \{P(x, a), P(y, z)\}$ due to the homomorphism $\{x \mapsto b, y \mapsto b, z \mapsto a\}$. For a subsitution σ defined on a factbase G containing F, let σ_F be the maximal subset of σ defined only on $\mathbf{Vars}(F)$.

Definition 2.4. A factbase G is a *retract* of another factbase F if $G \subseteq F$ and $G \vDash F$. A *retractation* from F to G is a homomorphism σ from F to G such that $\sigma_G = id_G$. G is a *strict retract* of F if G is a retract of F and $G \neq F$.

If G is a strict rectract of F, then G is logically as strong as F and yet smaller. We impose that $\sigma_G = id_G$ in the definition of a retractation because it helps us in the proof of Proposition 4.6 thanks to the well known property:

Proposition 2.1. *The factbase G is a retract of the factbase F if and only if $G \subseteq F$ and there exists a retractation from F to G.*

A core of a factbase F is then one of the smallest retract of F:

Definition 2.5. A *core* is a factbase F that does not contain a strict retract. A *core* of a factbase F is a subset of F that is a core.

The cores of a finite factbase F are unique up to isomorphism. Hence, we speak of "the" core of a factbase. The core of F is one of the smallest set of facts that is logically as strong as F. For example, the factbase $G = \{B(x, y), R(y, z)\}$ is the core of $F = \{B(x, y), R(y, z),\ B(x, w), R(w, z)\}$ because:

- $G \subseteq F$,
- $\{x \mapsto x, y \mapsto y, z \mapsto z, w \mapsto y\}$ is a homomorphism from F to G, so G is a retract of F, and
- all strict subsets of G are not retracts of G.

Note that $\{B(x, w), R(w, z)\}$ is also the core of F and is indeed isomorphic to G due to the homomorphism $\{x \mapsto x, y \mapsto w, z \mapsto z\}$. For a tuple of variables $\vec{x}$ and a formula A, we write $A(\vec{x})$ to denote that $\vec{x}$ is the set of all free variables that occur in A.

Definition 2.6. Let $\vec{x}$, $\vec{y}$, and $\vec{z}$ be some tuples of variables that are pairwise disjoint. An *(existential) rule* α is a first-order formula of the form

$$\forall \vec{x}.\forall \vec{y}.(A(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.B(\vec{x}, \vec{z}))$$

where $A(\vec{x}, \vec{y})$ and $B(\vec{x}, \vec{z})$ are conjunctions of facts. We define $Body(\alpha) = A$ and $Head(\alpha) = B$.

We omit the universal quantifiers when representing existential rules. A knowledge base represents a set of rules and a set of ground facts:

Definition 2.7. A *knowledge base* K is a pair (R, F) where R is a set of existential rules and F is a ground factbase.

A factbase F *entails* a rule α if each interpretation satisfying F satisfies α. We will note $F \vDash R$ if F entails each rule of the rule set R. Again, we will not have to use the formal definition of interpretations because the following theorem is used in pratice to determine if $F \vDash \alpha$:

Theorem 2.2. A factbase F *entails* a rule $\alpha = A(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.B(\vec{x}, \vec{z})$ if and only if for every homomorphism σ from A to F, there exists an extension of σ that is a homomorphism from B to F.

For example, the factbase $F = \{A(c,d), B(c,e)\}$ entails the rule $\alpha = A(x,y) \rightarrow \exists z.B(x,z)$ because $\sigma = \{x \mapsto c, y \mapsto d\}$ is the only homomorphism from $A(x,y)$ to F, and $\hat{\sigma} = \sigma \cup \{z \mapsto e\}$ is an extension of σ that is a homomorphism from $B(x,z)$ to F.

Definition 2.8 (Model and universality)**.** A factbase M is a *model* of a knowledge base $K = (R, F)$ if $M \vDash F$ and $M \vDash R$. The factbase U is *universal* for K if for every model M of K, $M \vDash U$.

We often talk of *universal models* of a knowledge base K; that is, factbases that are both a model of K and universal for K.

Example 2.1. If we set $K = (\{A(x) \rightarrow \exists z.R(x,z) \wedge A(z)\}, \{A(b)\})$, then a universal model for K is

$$U = \{A(b), R(b,x_0)\} \cup \{A(x_i) \mid i \in \mathbb{N}\} \cup \{R(x_i, x_{i+1}) \mid i \in \mathbb{N}\}$$

Note that the knowledge base K does not admit finite universal models.

Definition 2.9 (Entailment)**.** A knowledge base K *entails* a factbase B (often noted $K \vDash B$) if for each model M of K, $M \vDash B$.

We can use universal models to solve the conjunctive query entailment:

Proposition 2.2. *A knowledge base K entails a factbase B if there exists a universal model U of K such that $U \vDash B$.*

An important problem that we want to solve is: Given a knowledge base $K = (R, F)$ and a factbase Q, does $K \vDash Q$? It is well-known that this problem is undecidable ([2], theorem 4).

3 The Chase

The process of applying rules on a factbase in order to infer more knowledge is called forward chaining. Forward chaining in existential rules is usually achieved via a family of algorithms called the chase. It can be seen as a two-step process. First, it repeatedly applies rules to the set of facts (and sometimes computes the core to remove redundant facts). Then it looks for an answer to the BCQ in this saturated set of facts. This saturated set of facts is a universal model of the knowledge base. The chase is sound and complete; so it must be non-terminating since the problem of entailment is undecidable [2]. To determine how we apply a rule to a set of facts, we introduce the notion of trigger:

Definition 3.1 (Trigger)**.** Let α be a rule, σ be a substitution, and F be a factbase. The tuple $t = (\alpha, \sigma)$ is an *oblivious trigger* for F if:

- the domain of σ is the set of all variables occurring in $Body(\alpha)$.
- σ is a homomorphism from $Body(\alpha)$ to F.

In this case, we say that t is *applicable* on F.

The tuple $t = (\alpha, \sigma)$ is a *restricted trigger* for F if t is an oblivious trigger for F and if for all $\hat{\sigma}$ that extend σ over $\mathbf{Vars}(Head(\alpha))$, $\hat{\sigma}(Head(\alpha)) \nsubseteq F$.

Notice that a restricted trigger is also an oblivious trigger. We will use the term *trigger* to talk both oblivious and restricted triggers. The oblivious trigger $t = (\alpha, \sigma)$ for a factbase F is a restricted trigger for F if and only if the rule α is not satisfied by F. The chase considers triggers to infer new knowledge from an initial factbase. We now explain how it would apply a trigger, giving rise to the notion of application.

Definition 3.2 (application)**.** Let F be a factbase and $t = (\alpha, \sigma)$ be a trigger where α is of the form $A(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.B(\vec{x}, \vec{z})$. We set σ^s the substitution that extends σ over $\mathbf{Vars}(Head(\alpha))$ such that for $z \in \vec{z}$, $\sigma^s(z) = f^z_\alpha(\sigma(\vec{x}))$ where $f^z_\alpha(\sigma(\vec{x}))$ is a fresh variable unique with respect to the trigger $t = (\alpha, \sigma)$ and the variable z.

The factbase $\mathbf{appl}(F, t) = F \cup \sigma^s(Head(\alpha))$ is called an *application* on the factbase F through the trigger $t = (\alpha, \sigma)$. We say that the trigger t is *useless* on F if $\mathbf{appl}(F, t) = F$.

We use a naming convention that resembles function symbols because this will be convenient when we will discuss about a new variant of the chase. For example, if we have the rule $\alpha = A(x, y) \rightarrow \exists z.B(x, z)$, the factbase $F = \{A(b, c)\}$, and the substitution $\sigma = \{x \mapsto b, y \mapsto c\}$, then (α, σ) is a restricted trigger for F. We then have

$$\mathbf{appl}(F, (\alpha, \sigma)) = \{A(b, c), B(b, f^z_\alpha(b, c)\}$$

It is well known that the application preserves the universality:

Proposition 3.1. *Let K be a knowledge base, M be a model of K, F be a factbase such that $M \vDash F$, and t be an oblivious trigger. We then have $M \vDash \boldsymbol{appl}(F, t)$.*

We can now define the oblivious and restricted chase that are also defined in [3]. A sequence of rule applications is called a derivation. In this paper, we differentiate between oblivious derivations and restricted derivations:

Definition 3.3 (Derivation)**.** An *oblivious derivation* (respectively a *restricted derivation*) for a knowledge base $K = (F, R)$ is a (possibly infinite) sequence $D = F_0, F_1, F_2, \ldots$ where

- $F_0, F_1, \ldots$ are factbases such that $F_i \subsetneq F_{i+1}$.
- $F_0 = F$.
- For all $i > 0$, $F_i = \mathbf{appl}(F_{i-1}, t_i)$ where t_i is an oblivious trigger (resp. a restricted trigger).

The oblivious (resp. restricted) derivation $D = F_0, F_1, F_2, \ldots$ is *fair* if for every i and every oblivious (resp. restricted) trigger t for F_i, there exists $k \geq i$ such that t is useless on F_k (resp. t is not anymore a restricted trigger for F_k). An *oblivious chase* (resp. a restricted chase) for a knowledge base $K = (F, R)$ is a fair oblivious (resp. restricted) derivation $D = F_0, F_1, F_2, \ldots$

The use of fairness yields a semi-decision procedure since a fair derivation guarantees that we consider every possible application.

For every oblivious chase derivation $D = F_0, F_1, F_2, F_3, \ldots$ for K, we have $F_0 \subseteq F_1 \subseteq F_2 \subseteq \ldots$ The set $\cup_i F_i$ does not depend on the derivation D since for each F_i and for each trigger t for F_i, there exists $k > i$ such that t is useless on F_k. We can then set $\mathit{Obl(K)} = \cup_{i \in \mathbb{N}} F_i$ and we say that the oblivious chase *terminates* if *Obl(K)* is finite. It is well known that:

Theorem 3.1. For a knowledge base K, $\mathit{Obl}(K)$ is a universal model of K.

Consequently, the oblivious chase can be used to solve query entailment. It is more difficult to define the result of the restricted chase because the result depends on the order of the application of the rules. We define

$$\mathit{Res(K)} = \{\cup_{i \in \mathbb{N}} F_i \mid D = F_0, F_1, F_2, \ldots \text{is a retricted chase for } K\}$$

We say that the restricted chase *terminates* if there exists an element in *Res(K)* that is finite. It is well known that:

Theorem 3.2. For a knowledge base K and for every $U \in \mathit{Res}(K)$, U is a universal model of K.

Note that if the oblivious chase termines for a knowledge base K, then the restricted chase terminates for K. We present an example to showcase that the restricted chase may terminate even when the oblivivious chase does not:

Example 3.1. Suppose that we have the knowledge base $K = (\{\alpha = A(x, y) \rightarrow \exists z.A(y, z) \wedge A(z, y)\}, F)$ where $F = \{A(a, b)\}$. An oblivious chase derivation is $F_0, F_1, F_2, ...$ where

- $F_0 = F$,
- $t_1 = (\alpha, \{x \mapsto a, y \mapsto b\})$ is an oblivious trigger for F_0,
- $F_1 = \mathbf{appl}(F_0, t_1) = \{A(a, b), A(b, z_1), A(z_1, b)\}$ where $z_1 = f^z_\alpha(a, b)$,

- $t_2 = (\alpha, \{x \mapsto b, y \mapsto z_1\})$ is an oblivious trigger for F_1,
- $F_2 = \mathbf{appl}(F_1, t_2) = F_1 \cup \{A(z_1, z_2), A(z_2, z_1)\}$ where $z_2 = f^z_\alpha(b, z_1)$,
- ...

It will never terminate because each new fact brings new rule applications. So the oblivious chase does not terminate on K whereas the restricted chase does. A restricted chase derivation is F_0, F_1 where $F_0 = F$, $t_1 = (\alpha, \{x \mapsto\ a,\ y \mapsto b\})$, and $F_1 = \mathbf{appl}(F_0, t_1) = \{A(a, b), A(b, f^z_\alpha(a, b)),\ A(f^z_\alpha(a, b), b)\}$. This derivation is fair because there is not anymore any restricted trigger for F_1, so we have $F_1 \in \mathit{Res(K)}$.

This theorem follows from results in [3]:

Theorem 3.3. A knowledge base K entails a factbase B if $\mathit{Obl}(K) \vDash B$ or if there exists $U \in \mathit{Res}(K)$ such that $U \vDash B$. Conversely, if a knowledge base K entails a factbase B, then $\mathit{Obl}(K) \vDash B$ and for every $U \in \mathit{Res}(K)$, $U \vDash B$

The core chase terminates more often than the oblivious and restricted chase since the core chase terminates when the knowledge base admits a finite universal model. It was first defined in [4].

Definition 3.4 (Core derivation and fairness)**.** A *core derivation* for a knowledge base $K = (R, F)$ is a (possibly infinite) sequence $D = F_0, F_1, F_2, \ldots$ where $F_0 = F$, and for $i > 0$, either $F_i = \mathbf{appl}(F_{i-1}, t_i)$ is obtained by an application with t_i an oblivious trigger, or F_i is the core of F_{i-1}.

The core derivation D is *fair* if:

- For every i, for every oblivious trigger t for F_i, there exists k such that t is useless on F_k; or there exists $k > i$ such that t is not anymore an oblivious trigger for F_k.
- For every i, there exists $k \geq i$ such that F_k is a core.

Note that in the first condition, we can have $k < i$ because in a chase derivation a fact can be removed. The second condition is essential to guarantee the terminaison of a fair core derivation when there exists a finite universal model.

Definition 3.5. A *core chase* for a knowledge base $K = (R, F)$ is a fair core derivation $D = F_0, F_1, F_2, \ldots$ The core chase *terminates* on K if it is a finite core derivation.

It was proven in [4] that the result of the core chase for a knowledge base is unique up to isomorphism. Therefore, we can define the result of the core chase:

Definition 3.6. If the core chase terminates on K due to the finite core derivation $D = F_0, F_1, F_2, \ldots, F_i$, then we set $\mathit{Core}(K) = F_i$. Otherwise, if the core chase does not terminate, $\mathit{Core}(K)$ is undefined.

The following theorem highlights why the core chase has been introduced. It has been proven in ([4], Theorem 7).

Theorem 3.4. The knowledge base $K = (R, F)$ admits a finite universal model if and only if the core chase algorithm terminates on K.

We present an example to showcase that the core chase may terminate even when the restricted chase does not:

Example 3.2. Suppose that we have the knowledge base $K = (\{\alpha = A(x, y) \rightarrow \exists z.(A(x, x) \wedge A(y, z))\}, \{A(a, b)\})$. A restricted chase derivation is $F_0, F_1, F_2, ...$ where

- $F_0 = F$,
- $t_1 = (\alpha, \{x \mapsto a, y \mapsto b\})$ is a restricted trigger for F_0,
- $F_1 = \mathbf{appl}(F_0, t_1) = F_0 \cup \{A(a, a), A(b, z_1)\}$ where $z_1 = f^z_\alpha(a, b)$,

- $t_2 = (\alpha, \{x \mapsto b, y \mapsto z_1\})$ is a restricted trigger for F_1,
- $F_2 = \mathbf{appl}(F_1, t_2) = F_1 \cup \{A(b,b), A(z_1, z_2)\}$ where $z_2 = f^z_\alpha(b, z_1)$,
- $t_3 = (\alpha, \{x \mapsto z_1, y \mapsto z_2\})$ is a restricted trigger for F_2,
- $F_3 = \mathbf{appl}(F_2, t_3) = F_2 \cup \{A(z_1, z_1), A(z_2, z_3)\}$ where $z_3 = f^z_\alpha(z_1, z_2)$,
- ...

It will never terminate because each new fact brings new restricted triggers. The core chase terminates on K: a core chase derivation is $F_0, F_1, F_2, F_3, F_4, F_5$ where

- $F_0 = F$,
- $t_1 = (\alpha, \{x \mapsto a, y \mapsto b\})$ is an oblivious trigger for F_0,
- $F_1 = \mathbf{appl}(F_0, t_1) = F_0 \cup \{A(a,a), A(b, z_1)\}$ where $z_1 = f^z_\alpha(a, b)$,
- $t_2 = (\alpha, \{x \mapsto b, y \mapsto z_1\})$ is an oblivious trigger for F_1,
- $F_2 = \mathbf{appl}(F_1, t_2) = F_1 \cup \{A(b,b), A(z_1, z_2)\}$ where $z_2 = f^z_\alpha(b, z_1)$,
- $t_3 = \{x \mapsto a, y \mapsto a\}$ is an oblivious trigger for F_2, $F_3 = \mathbf{appl}(F_2, t_3)$, $t_4 = \{x \mapsto b, y \mapsto b\}$ is an oblivious trigger for F_3, and $F_4 = \mathbf{appl}(F_3, t_4) = F_2 \cup \{A(a, f^z_\alpha(a,a)), A(b, f^z_\alpha(b,b))\}$
- $F_5 = \mathit{Core}(F_4) = \{A(a,a), A(a,b), A(b,b)\}$.

All the oblivious triggers for F_5 have been applied so the derivation is fair. Consequently the core chase terminates on K.

The three chases presented in Section 3 are the most famous chases. The goal of my internship is to introduce a new chase that work in the particular case of **Horn-$\mathcal{ALCH}$** and prove that this new algorithm is as powerful as the core chase. The new chase is called the merge chase and is presented in the next section.

4 The Merge Chase

The core chase always terminates when there exists a finite universal model but this chase is very expensive timewise because computing the core of a factbase is hard. Therefore we present a more efficient way to solve this problem in the particular case of **Horn-$\mathcal{ALCH}$**. The new procedure, called the merge chase, considers a special relation between the variables. In the merge chase, two variables in relation are merged thanks to a local homomorphism.

For a factbase F and some $t, u \in \mathbf{Terms}(F)$, let $\mathbf{Preds}^1_F(t) = \{P \mid P(t) \in F\}$ and $\mathbf{Preds}^2_F(t,u) = \{P \mid P(t,u) \in F\}$. In this section, we only consider factbases with predicates of arity one or two. Hence, we will represent a factbase F by a labelled graph $G = (V, E)$ where $V = \{t \mid t \in \mathbf{Terms}\}$ and $E = \{(t,u) \mid t, u \in \mathbf{Terms} \wedge \mathbf{Preds}^2_F(t,u) \neq \emptyset\}$. We label each vertex $v \in V$ with the predicates in $\mathbf{Preds}^1_F(v)$ and each edge (t,u) with the predicates in $\mathbf{Preds}^2_F(t,u)$. For example, we represent $F = \{A(a), B(a), R(a,b), T(a,b), C(b), R(b,z)\}$ by the Figure 1.

Definition 4.1 (**Horn-$\mathcal{ALCH}$** axioms)**.** A ***Horn-$\mathcal{ALCH}$*** *axiom* is an existential rule of one of the following forms:

$$A(x) \wedge B(x) \to C(x) \quad (C_\wedge) \qquad R(x,y) \wedge B(y) \to A(x) \quad (\exists_-)$$
$$A(x) \wedge R(x,y) \to B(y) \quad (\forall_+) \qquad R(x,y) \wedge S(x,y) \to V(x,y) \quad (R_\wedge)$$
$$A(x) \to \exists y. R(x,y) \wedge B(y) \quad (\exists_+)$$

As a remainder, we omit the universal quantifiers when representing rules. A ***Horn-$\mathcal{ALCH}$*** *knowledge base* $K = (R, F)$ is a knowledge base where R is a set of **Horn-$\mathcal{ALCH}$** axioms and F only contains predicates of arity one or two.

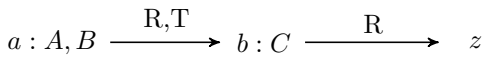

Figure 1: Factbase Representation

Horn-$\mathcal{ALCH}$ was introduced in [5]. Note that the type of **Horn-$\mathcal{ALCH}$** rule $(\exists_+)$ is the only one that features existentially quantified variables. To introduce the merge chase, we first define the notion of mergeable variable, that allows us to consider a special relation between variables. We then describe the atomic merging operation, that merges a mergeable variable on a term. This atomic operation is used to define the merging that does atomic merging until it is not possible anymore. The merge chase is then a core chase where we replace the core operation by the merging. We want to show that the merge chase for a **Horn-$\mathcal{ALCH}$** knowledge base K computes a finite universal model when it terminates and that it terminates if and only if there exists a finite universal model of K.

For a term t and a variable x, we say that $t \prec x$ if x is of the form $f^y_\alpha(t)$. We write $\prec^+$ to denote the transitive closure of $\prec$ and we set $\boldsymbol{Desc}(t)= \{y \in \mathbf{Vars} \mid t \prec^+ y\} \cup \{t\}$. The type of **Horn-$\mathcal{ALCH}$** rule $(\exists_+)$ is the only one that leads to the introduction of an existentially quantified variable when we apply it. Therefore:

Proposition 4.1. *Consider a variable x that occurs in an oblivious chase derivation D of some **Horn-$\mathcal{ALCH}$** knowledge base K. Then, x is of the form $f^y_\alpha(u)$ where u is a term, α is a rule of the form $(\exists_+)$ in K, and y is the only existentially quantified variable occurring in α. Moreover, u is the only term in D such that $u \prec x$.*

Proposition 4.2. *For a factbase F that occurs in an oblivious chase derivation of some **Horn-$\mathcal{ALCH}$** knowledge base, $\prec^+$ is a strict partial order over the set of terms of F.*

The proof of the last proposition is given in the section A. We can deduce that the graph induced by $\prec$ does not contain any cycle. We obtain the following result from Propositions 4.1 and 4.2:

Proposition 4.3. *For a factbase F that occurs in an oblivious chase derivation of some **Horn-$\mathcal{ALCH}$** knowledge base, the graph induced by $\prec$ is a forest of trees and the root of each tree is a constant.*

We define the notion of tree of a term t, that is all the facts that only contain variables in $\mathbf{Desc}(t)$:

Definition 4.2 (Tree)**.** For a term t occurring in an oblivious chase derivation of the **Horn-$\mathcal{ALCH}$** knowledge base K, we set:

$$\begin{aligned} \mathit{Tree}_F(t) = &\{A(u) \mid A(u) \in F \wedge u \in \mathbf{Desc}(t)\} \cup \\ &\{R(u,x) \mid R(u,x) \in F \wedge u, x \in \mathbf{Desc}(t)\} \end{aligned}$$

Definition 4.3 (Mergeable variable)**.** For two terms u and v appearing in a factbase F, u is *mergeable* on v in F if:

- $u \neq v$ and u is a variable,
- $\mathbf{Preds}^1_F(u) \subseteq \mathbf{Preds}^1_F(v)$, and
- there exists a term t such that $\mathbf{Preds}^2_F(t,u) \neq \emptyset$ and $\mathbf{Preds}^2_F(t,u) \subseteq \mathbf{Preds}^2_F(t,v)$.

In this case, we say that u is a *mergeable variable*.

When x is mergeable on t in F, all the facts in $\mathit{Tree}_F(x)$ "are in" $\mathit{Tree}_{\mathbf{Fut}(F,t,x)}(t)$ where $\mathbf{Fut}(F,t,x)$ is a factbase obtained by applying some oblivious triggers on F. To be more precise, the words "are in" mean that there exists an injective homomorphism from the first set to the second. Note that this result is one of the most importants to prove that the merge chase computes a universal factbase when it terminates, which is Proposition 4.4. From this result, we can conclude that all the facts in $\mathit{Tree}_F(x)$ are redundant or can become redundant by applying some triggers. Therefore, we would like to remove the facts already redundant and "recycle" the other facts. This intuition leads us to define atomic merging:

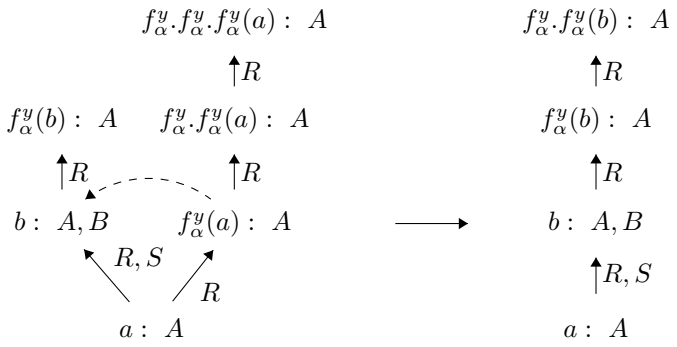

Figure 2: Atomic Merging Example

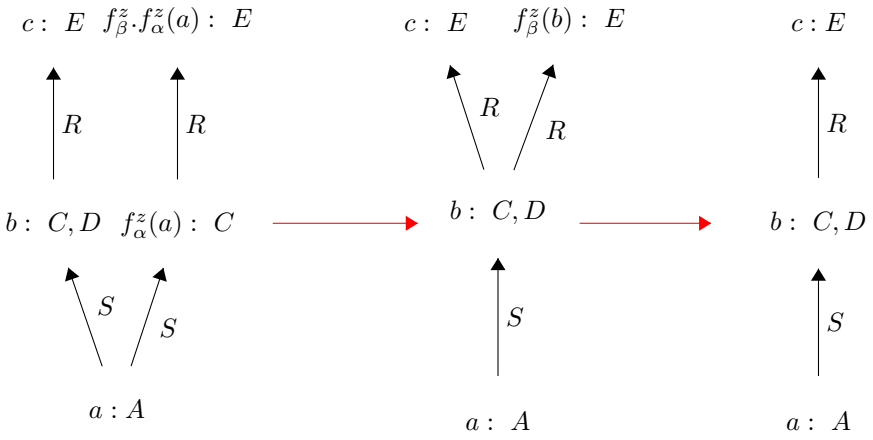

Figure 3: Merging Example

Definition 4.4 (atomic merging)**.** If a variable x is mergeable on a term t in a factbase F, then we define $h_{x/t}$ to be the substitution defined on $\mathbf{Vars}(F)$ such that

- for every variable y in $\mathbf{Desc}(x)$, $h_{x/t}(y)$ is the variable y where the occurence of x is replaced by t, and
- for every variable y not in $\mathbf{Desc}(x)$, $h(y) = y$.

The *atomic merging* of x on t in F produces $h_{x/t}(F)$

For example, in the Figure 2, $f^y_\alpha(a)$ is mergeable on b on the left factbase. The atomic merging of $f^y_\alpha(a)$ on b in this factbase produces the factbase on the right where all the occurences of $f^y_\alpha(a)$ are replaced by b. The following definition describes the new operation that will replace the computation of the core in the core chase.

Definition 4.5. For a factbase F, a *merge* sequence of F is a sequence $F_0, \ldots, F_n$ of factbases such that:

- $F_0 = F$.
- For $i \in \{1, \ldots, n\}$, the factbase F_i is the atomic merging of a variable x on a term t in F_{i-1}.
- The factbase F_n does not contain a mergeable variable.

Then, $\mathit{Merge}(F) = F_n$.

The result of the *Merge* operation is unique up to isomorphism. For an oblivious chase derivation $D = F_0, ..., F_k$ of K, if we apply our new operation on F_k, then it computes the core of a factbase that could have been produced by continuing the derivation D.

Example 4.1. In the Figure 3, we merge the factbase on the left. The variable $f^z_\alpha(a)$ is mergeable on b, an atomic merging of $f^z_\alpha(a)$ on b yields the factbase on the middle. Now, $f^z_\beta(b)$ is mergeable on c, an atomic merging of $f^z_\beta(b)$ on c produces the factbase on the right. The last factbase is the result of the merge since it does not contain any mergeable variable.

We can now consider a new chase:

Definition 4.6 (derivation)**.** A *merge derivation* for a **Horn-$\mathcal{ALCH}$** knowledge base $K = (R, F)$ is a (possibly infinite) sequence $D = F_0, F_1, F_2, \dots$ where $F_0 = F$, and for $i > 0$, either $F_i = \mathbf{appl}(F_{i-1}, t_i)$ is obtained by an application with t_i an oblivious trigger, or F_i is obtained by the merging of F_{i-1}.

D is *fair* if:

- For every i, for every oblivious trigger t applicable on F_i, there exists $k \in \mathbb{N}$ such that t is useless on F_k; or there exists $k > i$ such that t is not anymore an oblivious trigger for F_k.
- For every i, there exists $k \geq i$ such that F_k is a core.

A *merge chase* for a knowledge base $K = (R, F)$ is a fair merge derivation $D = F_0, F_1, F_2, \dots$

We also introduce the atomic merge chase in order to simplify a future proof, and to show that we can be more flexible on the use of the merge chase. The *atomic merge chase* is a merge chase where we use the atomic merge operation instead of the merge operation.

Definition 4.7. The merge chase *terminates* on K if it is a finite derivation $D = F_0, F_1, F_2, \dots, F_k$. We set, in this case, $\mathbf{MC}(K) = F_k$ the result of the merge chase.

The result of the merge chase is unique up to isomorphism. The next two propositions will help us show that the atomic merge chase only computes universal models.

Proposition 4.4. *Let $D = F_0, F_1, \dots, F_m$ be an atomic merge chase derivation of a knowledge base K and let x be a variable mergeable on a term t in F_m. Then, there exists an atomic merge chase derivation D' of the form D, E (where E is a sequence of factbases) such that:*

1. *If we consider some factbase $F \in E$ and its predecessor G in D', then there is an oblivious trigger $t = (\alpha, \sigma)$ for G such that $F = \boldsymbol{appl}(G, t)$.*
2. *The last element of E is the factbase $F_m \cup h_{x/t}(\mathit{Tree}_{F_m}(x))$.*

We set $\boldsymbol{Fut}(F_m, t, x) = F_m \cup h_{x/t}(\mathit{Tree}_{F_m}(x))$.

Graphically, in the Figure 4, if we set $A = \mathit{Tree}_{F_m}(t)$ and $B = \mathit{Tree}_{F_m}(x)$, then we want to do an atomic merge derivation from the left factbase to the factbase on the right. In the figure, the sets A and $h_{x/t}(B)$ can have facts in common.

To prove Proposition 4.4, we will look at all the triggers dealing with the variables in $\mathbf{Desc}(x)$ and apply them on the variables in $\mathbf{Desc}(t)$ if they have not already been applied. You can find the proof in the section B.

Proposition 4.5. *Let $D = F_0, F_1, \dots, F_m$ be an atomic merge chase derivation for the knowledge base K. If x is mergeable on t in F_m, then the atomic merging of x on t in F_m is a retract of $\boldsymbol{Fut}(F_m, t, x)$.*

The proof is in section C. The following theorem is a corollary of Propositions 4.4 and 4.5:

Theorem 4.1. Let $D = F_0, F_1, \dots$ be an atomic merge chase derivation for the knowledge base K. Then, the factbases $F_0, F_1, \dots$ are universal for K.

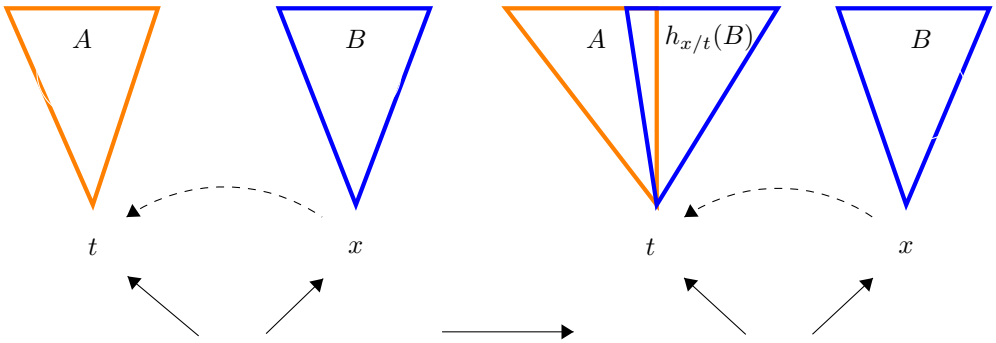

Figure 4: Illustration of the proof

Proof. Let M be a model for K. We prove by induction on $n \geq 0$ that $M \vDash F_n$. By definition of a model, $M \vDash F$ so the initial step is true since $F_0 = F$. Suppose that $M \vDash F_{n-1}$, where $n > 0$, and that F_{n-1} is not the last factbase of the derivation D.

- If $F_n = \mathbf{appl}(F_{n-1}, t)$, where $t = (\alpha, \sigma)$ is an oblivious trigger, by Proposition 3.1, $M \vDash F_n$.
- If F_n is the atomic merging of a variable x on a term t in F_{n-1}, then according to Proposition 4.4, there exist factbases $G_0, \ldots, G_k$ such that
  - $G_0 = F_{n-1}$,
  - $G_k = \mathbf{Fut}(F_{n-1}, t, x)$, and
  - for $0 \leq i \leq k-1$, $G_{i+1} = \mathbf{appl}(G_i, t)$ where t is an oblivious trigger for G_i.

  We then have by induction on $0 \leq i \leq k$ and by Proposition 3.1, $M \vDash G_i$. Therefore $M \vDash \mathbf{Fut}(F_{n-1}, t, x)$, so M is a model of the atomic merging of x on t in F_{n-1} (that is F_n) since, by Proposition 4.5, F_n is a subset of $\mathbf{Fut}(F_{n-1}, t, x)$.

The heredity is proved. □

The main reason to prefer the merge chase to the atomic merge chase is that unless the atomic merging operations are applied exhaustively, the merge chase may not terminate on inputs that admit finite universal models:

Theorem 4.2. There exists a **Horn-$\mathcal{ALCH}$** knowledge base K such that K admits a finite universal model and the atomic merge chase derivation for K does not terminate.

Proof. Let $F = \{R(a,b), R(b,a), A(a), B(b)\}$ and $R = \{\alpha = A(x) \to \exists y.R(x,y) \wedge A(y), \beta = B(x) \to A(x)\}$. We consider for the knowledge base $K = (R, F)$, the atomic merge chase derivation $F_0 = F, F_1, F_2, \ldots$ constructed as follow:

- We apply the oblivious trigger $t_1 = (\alpha, \{x \mapsto a\})$ to F_0;
- we then apply the oblivious trigger $t_2 = (\alpha, \{x \mapsto f^y_\alpha(a)\})$ to F_1 to obtain the factbase F_2 and we apply the oblivious trigger $t_3 = (\beta, \{x \mapsto b\})$ to F_2 giving rise to the factbase F_3 that is the first factbase of Figure 5;
- At this moment, $f^y_\alpha(a)$ is mergeable on b so we do an atomic merging of $f^y_\alpha(a)$ on b to get F_4 that is the second factbase of the figure;
- F_5 is obtained by the application of the oblivious trigger $t_4 = (\alpha, \{x \mapsto f^y_\alpha(b)\})$ and F_6 by the oblivious trigger $t_5 = (\alpha, \{x \mapsto f^y_\alpha(f^y_\alpha(b))\})$, F_6 is the third factbase of the figure;

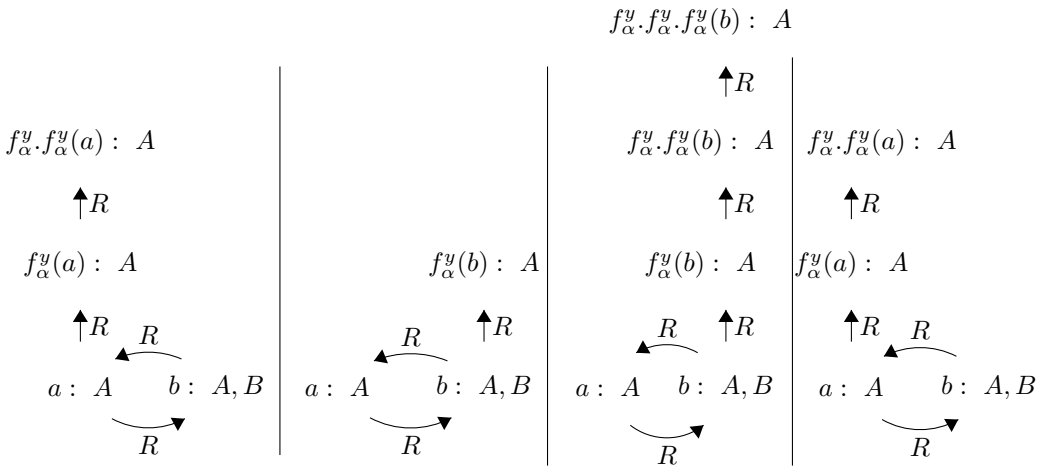

Figure 5: Example where the atomic merge chase is not efficient

- At this moment, $f^y_\alpha(b)$ is mergeable on a so we do an atomic merging of $f^y_\alpha(b)$ on a to get F_7 that is the last factbase of the figure;
- We can repeat this infinitely.

But K admits a finite universal model: $U = F \cup \{A(b)\}$.

□

We now want to prove that a merging computes a core:

Proposition 4.6. *For a factbase G that occurs in any merge chase derivation of some **Horn-$\mathcal{ALCH}$** knowledge base, Merge(G) is a core.*

We prove it in section D. Note that *Merge*(G) is a core but not necessarily the core of G. We need the following lemma to show that the merge chase for a **Horn-$\mathcal{ALCH}$** knowledge base K computes a model of K. It is proved in section E.

Proposition 4.7. *Let $D = F_0, \ldots, F_k, \ldots$ be an atomic merge chase derivation of a knowledge base K, α be a rule, and σ be a substitution such that $\sigma(Body(\alpha)) \subseteq F_k$. Then, either there exists $n > k$ such that $t = (\alpha, \sigma)$ is not an oblivious trigger for $F_n, F_{n+1}, \ldots$ or there exists $n \geq 0$ such that for all $j \geq n$, there exists a substitution σ_j prolonging σ and verifying $\sigma_j(Head(\alpha)) \subseteq F_j$.*

Proposition 4.8. *The merge chase computes a finite universal model of K when it terminates.*

Proof. Let $D = F_0, \ldots, F_n$ be a merge chase for $K = (R, G)$.

- The merge chase never removes ground facts and $G = F_0$ so $G \subseteq F_n$. We then have $F_n \vDash G$. Suppose that there exists a rule α in R and a substitution σ such that $F_n \vDash \sigma(Body(\alpha))$. Then by Proposition 4.7, there exists a substitution $\hat{\sigma}$ prolonging σ such that $\hat{\sigma}(Head(\alpha)) \subseteq F_n$. We can conclude that $F_n \vDash R$, so F_n is a model of K.
- According to Theorem 4.1, F_n is universal for K.
- F_n is finite.

□

Proposition 4.9. *If there exists a finite universal model for a* ***Horn-$\mathcal{ALCH}$*** *knowledge base $K = (R, F)$, then the merge chase terminates.*

The last proposition is proven in section F. The following theorem is then a direct consequence of Propositions 4.8 and 4.9.

Theorem 4.3. The merge chase computes a universal model if and only if there exists a universal model.

We will now describe a deterministic algorithm to merge a factbase:

Definition 4.8 (Merging)**.** Let F be a factbase that occurs in a merge chase derivation of the knowledge base K.

Algorithm 1: M(F):

```
1 Let Vars(F) = {x_1, ..., x_n} be such that (x_i ≺⁺ x_j) ⇒ i < j ;
2 for i = 1 to n do
3 |  if x_i is still a variable in F then
4 |  |  for all term t such that x_i is mergeable on t do
5 |  |  |  F ← the atomic merging of x_i on t in F.
6 |  |  end
7 |  end
8 end
9 return F
```

On line 1, we can sort terms that way because, by proposition 4.2, $\prec^+$ is a strict partial order over the set of variables of F.

Note that the application of an atomic merge may result in new mergeable variables. Therefore, we have to be careful about the order of the variables in the Algorithm 1. In the Example 4.1, if we treat $f^z_\beta(f^z_\alpha(a))$ before $f^z_\alpha(a)$, then at the moment when we treat $f^z_\beta(f^z_\alpha(a))$, it is not mergeable on any term yet. In the end, we would get the factbase on the middle and we would not have merged every possibly mergeable variable. Example 4.2 shows the importance of doing a total merging. We have to prove that our merging algorithm does a total merging:

Proposition 4.10. *Let G be a factbase that occurs in a chase derivation of a knowledge base K. Then, there does not exist a term t and a variable x such that x is mergeable on t in $M(G)$.*

The proof is in section G

5 Conclusion

During this internship, our research work will have given progress around the problem of conjunctive query entailment in the case of **Horn-$\mathcal{ALCH}$** by introducing the merge chase. The most important result is the Theorem 4.3 stating that there exists a finite universal model of K if and only if the merge chase terminates for a knowledge base K and its result is a universal model of K. We think that we can extend our work to **Horn-$\mathcal{ALCHI}$** rules. A **Horn-$\mathcal{ALCHI}$** rule is either a **Horn-$\mathcal{ALCH}$** axiom or an existential rule of the form $R_1(x, y) \wedge R_2(x, y) \rightarrow S(y, x)$.

This internship was a good introduction to the research's world and confirm my career orientation choices towards research. I particularly enjoyed the interaction with the entire GRAPHIK team during informal discussions around research.

References

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases: The Logical Level.* Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1995.

[2] Catriel Beeri and Moshe Y. Vardi. The implication problem for data dependencies. In *Proceedings of the 8th Colloquium on Automata, Languages and Programming*, page 73–85, Berlin, Heidelberg, 1981. Springer-Verlag.

[3] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *Journal of Artificial Intelligence Research*, 48:115–174, Oct 2013.

[4] Alin Deutsch, Alan Nash, and Jeffrey B. Remmel. The chase revisited. In Maurizio Lenzerini and Domenico Lembo, editors, *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, pages 149–158. ACM, 2008.

[5] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Complexities of horn description logics. *ACM Trans. Comput. Log.*, 14(1):2:1–2:36, 2013.

Appendix

Contents

A Proof of Proposition 4.2 **15**

B Proof of Proposition 4.4 **15**

C Proof of Proposition 4.5 **16**

D Proof of Proposition 4.6 **16**

E Proof of Proposition 4.7 **17**

F Proof of Proposition 4.9 **17**

G Proof of Proposition 4.10 **18**

H Institutional and social context of the internship **19**

A Proof of Proposition 4.2

Suppose for a contradiction that there exists a term t such that $t \prec^+ t$. Then, there exists $n \in \mathbb{N}^*$ and terms $t_0, \ldots, t_n$ such that $t = t_0 \prec t_1 \prec \cdots \prec t_n = t$. By Proposition 4.1, there exists rules $\alpha_1, \ldots, \alpha_n$ and variables $v_1, \ldots, v_n$ such that $t_n = f^{v_n}_{\alpha_n}(f^{v_{n-1}}_{\alpha_{n-1}}(\cdots f^{v_1}_{\alpha_1}(t_0)\cdots))$. It is a contradiction since $t_0 = t$ and $t_n = t$. Therefore $\prec^+$ is irreflexive. By construction, $\prec^+$ is transitive; So $\prec^+$ is a strict partial order

B Proof of Proposition 4.4

In order to define an atomic merge chase sequence such as D', we first introduce some sequences of oblivious triggers:

- Let $t_1 = (\alpha_1, \sigma_1), \ldots, t_n = (\alpha_n, \sigma_n)$ be the maximal sequence of oblivious triggers such that:
- (1) for each $1 \leq i \leq n$, there is some $1 \leq j \leq m$ such that $\mathbf{appl}(F_j, t_i) = F_{j+1}$;

(2) if $F_k = \mathbf{appl}(F_{k-1}, t_i)$ and $F_l = \mathbf{appl}(F_{l-1}, t_j)$ for some $i, j \in \{1, \ldots, n\}$ and some $1 \leq k < l \leq m$, then $i < j$; and

(3) the range of σ_i is a subset of $\mathbf{Desc}(x)$ for each $1 \leq i \leq n$.

- Let $t'_1, \ldots, t'_o$ be the maximal subsequence of $(\alpha_1, h_{x/t} \circ \sigma_1), \ldots, (\alpha_n, h_{x/t} \circ \sigma_n)$ that are not useless triggers for F_m

Intuitively, $t_1, \ldots, t_n$ are the triggers that have generated $Tree_{F_m}(x)$ and are ordered by application time. And $t'_1, \ldots t'_o$ are the triggers that we have to apply to produce the factbase $F_m \cup h_{x/t}(Tree_{F_m}(x))$.

Let $F_{m+i} = \mathbf{appl}(F_{m+i-1}, t'_i)$ for each $1 \leq i \leq o$. We show via induction that the sequence $D' = D, F_{m+1}, \ldots, F_{m+o}$ satisfies conditions 1. and 2. stated in the proposition.

Suppose for $1 \leq i \leq n$ that $D, F_{m+1}, \ldots, F_{m+i-1}$ is an atomic merge derivation such that the triggers $t'_1, \ldots, t'_{i-1}$ are useless for F_{m+i-1}. There exists $1 \leq j \leq n$ such that the trigger t'_i is equal to $h_{x/t}(t_j)$. The oblivious trigger $h_{x/t}(t_j)$ is not useless for F_{m+i-1}. We want to show that $t'_i = h_{x/t}(t_j)$ is an oblivious trigger for F_{m+i-1}. By definition of an **Horn-$\mathcal{ALCH}$** rule, the only facts that can appear in $Body(\alpha_j)$ are of the form $A(u)$ or $R(u, v)$ where u, v are variables and A, R are predicates.

- Suppose that $A(u) \in Body(\alpha_j)$ where u is a variable and A is an unary predicate. As t_j is an oblivious trigger for F_m, $A(\sigma_j(u)) \in F_m$. We have $\sigma_j(u) \in \mathbf{Desc}(x)$ so either $\sigma_j(u) = x$ or $x \prec^+ \sigma_j(u)$.
  - If $\sigma_j(u) = x$, then $A(x) \in F_m$. As x is mergeable on t in F_m, $A(t) \in F_m$. So $A(h_{x/t}(\sigma_j(u))) \in F_{m+i-1}$ since $h_{x/t}(\sigma_j(u)) = t$ and $F_m \subseteq F_{m+i-1}$.
  - If $x \prec^+ \sigma_j(u)$, then the fact $A(\sigma_j(u))$ is in $\sigma^s_k(Head(\alpha_k))$ where $k < j$. By induction hypothesis, $h_{x/t}(t_j)$ is useless for F_{m+i-1} so $A(h_{x/t}(\sigma_j(u))) \in F_{m+i-1}$.
- Suppose that $R(u, v) \in Body(\alpha_j)$ where u and v are variables and R is a binary predicate. As t_j is an oblivious trigger for F_m, $R(\sigma_j(u), \sigma_j(v)) \in F_m$. We have $\sigma_j(u), \sigma_j(v) \in \mathbf{Desc}(x)$ and $\sigma_j(u) \prec \sigma_j(v)$ so the fact $R(\sigma_j(u), \sigma_j(v))$ is in $\sigma^s_k(Head(\alpha_k))$ where $k < j$. By induction hypothesis, $h_{x/t}(t_j)$ is useless for F_{m+i-1} so $R(h_{x/t}(\sigma_j(u)), h_{x/t}(\sigma_j(v))) \in F_{m+i-1}$.

Therefore $h_{x/t}(t_j)$ is an oblivious trigger for F_{m+i-1}. Consequently, D_i is an atomic merge derivation. By definition of an application, we have $h_{x/t} \circ \sigma^s_j(Head(\alpha_j)) \subseteq F_{m+i}$ so $H(i)$ is true.

We have proved the heredity. So, D_n is the oblivious derivation that we were looking for. We have $F_{m+k} = F_m \cup h_{x/t}(Tree_{F_m}(x))$.

C Proof of Proposition 4.5

- The atomic merging of x on t in F_m is $h_{x/t}(F_m)$.
- By Proposition 4.4, we have that $\mathbf{Fut}(F_m, t, x) = F_m \cup\ h_{x/t}(Tree_{F_m}(x))$.
  So $\mathbf{Fut}(F_m, t, x) = Tree_{F_m}(x) \cup\ h_{x/t}(F_m)$ and we then have $h_{x/t}(F_m) \subseteq \mathbf{Fut}(F_m, t, x)$.
- Finally, as $h_{x/t}(\mathbf{Fut}(F_m, t, x)) = h_{x/t}(F_m)$, we have that $h_{x/t}(F_m) \vDash \mathbf{Fut}(F_m, t, x)$.

D Proof of Proposition 4.6

Suppose for a contradiction that $Merge(G)$ is not a core. Then, there exists a factbase $G' \subsetneq Merge(G)$ such that G' is a retract of $Merge(G)$. Then, by Proposition 2.1, there exists a retractation h from $Merge(G)$ to G'. We have $var(Merge(G)) \setminus var(G') \neq \emptyset$. Let x be a $\prec$-minimal variable of this set. The term x is a variable, so has been introduced by the chase due to a rule of the form $(\exists_+)$, so there exists a term t such that $t \prec x$.

- We have $\mathbf{Preds}^2_{Merge(G)}(t, x) \neq \emptyset$. By $\prec$-minimality of x, $t \in \mathbf{Vars}(G')$. So, as h is a retractation: $h(t) = t$, so for $R \in \mathbf{Preds}^2_{Merge(G)}(t, x)$, $h(R(t, x)) = R(t, h(x)) \in Merge(G)$ and so $R \in \mathbf{Preds}^2_{Merge(G)}(t, h(x))$. Thus $\mathbf{Preds}^2_{Merge(G)}(t, x) \subseteq \mathbf{Preds}^2_{Merge(G)}(t, h(x))$.

- As $\mathbf{Preds}^2_{Merge(G)}(t,x) \subseteq \mathbf{Preds}^2_{Merge(G)}(t,h(x))$ and $\mathbf{Preds}^2_{Merge(G)}(t,x) \neq \emptyset$, we have $t \prec h(x)$.
- $x \notin G'$ and $h(x) \in G'$ so $h(x) \neq x$.
- Let $A \in \mathbf{Preds}^1_{Merge(G)}(x)$. $h(A(x)) \in \mathit{Merge}(G)$ so $A(h(x)) \in \mathit{Merge}(G)$ so $\mathbf{Preds}^1_{Merge(G)}(x) \subseteq \mathbf{Preds}^1_{Merge(G)}(h(x))$.

Consequently, x is mergeable on $h(x)$ in $\mathit{Merge}(G)$ which results on a contradiction. Hence, $\mathit{Merge}(G)$ is a core.

E Proof of Proposition 4.7

As the derivation D is fair, there exists $n \geq 0$ such that t is useless on F_n; or there exists $n > i$ such that t is not an oblivious trigger for F_n.

- If the trigger t is not an oblivious trigger for F_n, then a variable in $\mathbf{Vars}(\sigma(\mathit{Body}(\alpha))$ is not in $\mathbf{Vars}(F_n)$ and cannot reappear. So, t is not a trigger for $F_n, F_{n+1}, \ldots$
- Otherwise, there exists $n \geq 0$ such that $\mathbf{appl}(F_n, t) = F_n$ and for every $i \geq n$, $\sigma(\mathit{Body}(\alpha)) \subseteq F_i$. We show by induction that for $j \geq n$, there exists a substitution $\hat{\sigma}$ prolonging σ such that $\hat{\sigma}(\mathit{Head}(\alpha)) \subseteq F_j$. The initialisation is true with $\hat{\sigma} = \sigma^s$ since $\sigma^s(\mathit{Head}(\alpha)) \subseteq F_n$. Assume that there exists a substitution $\hat{\sigma}$ prolonging σ such that $\hat{\sigma}(\mathit{Head}(\alpha)) \subseteq F_{j-1}$ where $j > n$. If $F_j = \mathbf{appl}(F_{j-1}, t)$ where t is an oblivious trigger, then $\hat{\sigma}(\mathit{Head}(\alpha)) \subseteq F_j$ since $F_{j-1} \subseteq F_j$. If F_j is the atomic merging of the variable x on the term t in F_{j-1}, then $F_j = h_{x/t}(F_{j-1})$.

  If α is not of the form $(\exists_+)$, then $\hat{\sigma} = \sigma$ and $h_{x/t} \circ \sigma = \sigma$ since the variables in the range of σ are in F_j. Therefore, $\hat{\sigma}(\mathit{Head}(\alpha)) \subseteq F_j$

  Otherwise, α is of the form $A(y) \to \exists z.R(y,z) \wedge B(z)$. we do not have $x \prec \hat{\sigma}(z)$ in F_j because the variables in the range of σ are in F_j. So, if $x \neq \hat{\sigma}(z)$, then $h_{x/t}(\hat{\sigma}(z)) = \hat{\sigma}(z)$ and so $h_{x/t} \circ \hat{\sigma} = \hat{\sigma}$. So $\hat{\sigma}(\mathit{Head}(\alpha)) \subseteq F_j$. Otherwise, we have $x = \hat{\sigma}(z)$, so $h_{x/t}(\hat{\sigma}(z)) = t$. We then have $h_{x/t}(\hat{\sigma}(\mathit{Head}(\alpha))) = \{B(t), R(\sigma(y), t)\}$. The variable x is mergeable on t and $B(x), R(\sigma(y), x) \in F_{j-1}$. So by definition of a mergeable variable $B(t), R(\sigma(y), t)$ are in F_{j-1} and so $h_{x/t}(\hat{\sigma}(\mathit{Head}(\alpha))) \subseteq F_j$.

  We can conclude that $h_{x/t} \circ \hat{\sigma}$ is the substitution that we are looking for. The heredity is proved.

F Proof of Proposition 4.9

- Suppose for a contradiction that K admits a finite universal model U of K and does not admit a finite merge chase sequence. So, there exists an infinite merge chase sequence $D = F_0, F_1, \ldots$
- A term t is *redundant* with respect to this sequence if $t \in \mathbf{Terms}(F_i)$ and $t \notin \mathbf{Terms}(F_j)$ for some $i < j$.
- For each i, let G_i be the maximal subset of F_i that does not contain redundant terms.
- We set $M = \cup_i G_i$. For a model N of K and for $i \in \mathbb{N}$, there exists a homomorphism h_i from F_i to N since F_i is universal for K by Proposition 4.1. As $G_i \subseteq F_i$, h_i is a homomorphism from G_i to N. Finally, $\cup_i h_i$ is a homomorphism from M to N so M is universal for K.
- Constants cannot be redundant, so $F \subseteq M$ since $G_0 = F$, we then have $M \vDash F$. Suppose that there exists a rule α in R and a substitution σ such that $M \vDash \sigma(\mathit{Body}(\alpha))$. Then, there exists $n \geq 0$ such that $\sigma(\mathit{Body}(\alpha)) \subseteq G_n$, so $\sigma(\mathit{Body}(\alpha)) \subseteq F_n$ since $G_n \subseteq F_n$. The trigger (α, σ) is an oblivious trigger for $F_n, F_{n+1}, \ldots$ because (α, σ) is an oblivious trigger for G_n and no variable in G_n is redundant. Then by Proposition 4.7, for $k \geq n$, there exists a substitution σ_k prolonging σ such that $\sigma_k(\mathit{Head}(\alpha)) \subseteq F_k$. There is a problem here because if the sequence $\sigma_1, \ldots$ is not constant starting from a certain rank, then it means that for every $k \geq 0$, there exist variables in the range of σ_k that are redundant. We show that it cannot be the case.

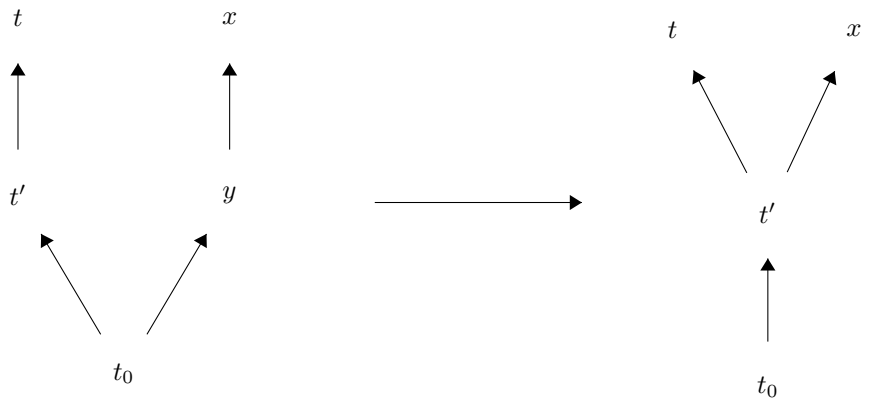

Figure 6: Illustration of the proof

If α is not of the form $(\exists_+)$, then $\sigma_k = \sigma$. As all the variables in the range of σ are in $\mathbf{Vars}(M)$, they are not redundant and so $\sigma(\mathit{Head}(\alpha)) \subseteq G_n \subseteq M$. Otherwise α is of the form $A(x) \to \exists y.R(x,y) \wedge B(y)$. We then have $\sigma_k = \sigma \cup \{y \mapsto f^z_\beta(\sigma(x))\}$ with β a rule in R and z the existential variable of the rule β. There is a finite number of rules in R and once a variable is merged, it does not reappear, therefore there exists $k \geq n$ such that for all $i \geq k$, $\sigma_i = \sigma_k$.Then $\sigma_k(y)$ is not redundant and as all the variables in the range of σ are in $\mathbf{Vars}(M)$, they are not redundant, so $\sigma_k(\mathit{Head}(\alpha)) \subseteq G_k \subseteq M$. We can conclude that $M \vDash R$, so M is a universal model of K.

- As M is a universal model of K, $M \vDash U$. Then, there exists a homomorphism h from U to M. As U is finite, the factbase $h(U) \subseteq M$ is also finite. The sequence $G_0, G_1, \ldots$ is monotonic (that is $G_0 \subseteq G_1 \subseteq \cdots$) so there exist i such that $h(U) \subseteq G_i$. We then have $F_i \vDash h(U)$ since $G_i \subseteq F_i$. We have $h(U) \vDash U$, so $F_i \vDash U$.
- As D is fair, there exists $j \geq i$ such that F_j is a core.
- By proposition 4.1, F_j is universal for K so $U \vDash F_j$. We have $F_j \vDash U$ since $F_j \vDash F_i$ and $F_i \vDash U$. So F_j is isomorphic to the core of U. Therefore F_j is a universal model of K.
- There exists a finite number of triggers for F_j and D is fair. So, there exists $k \geq j$ such that all the triggers for F_j have been applied in the derivation $F_0, F_1, \ldots, F_k$. Note that this step is necessary to guarantee the first condition in the Definition 4.6 of fairness.
- As D is fair, there exists $l \geq k$ such that F_l is a core. As F_j is a model of K, $F_l = F_j$.
- The derivation $F_0, \ldots, F_l$ is then fair so it is a finite merge chase sequence that leads us to a contradiction.

G Proof of Proposition 4.10

Suppose for a contradiction that there exists a term t and a variable x such that x is mergeable on t in $M(G)$, that is, there exists a term t' such that $\mathbf{Preds}^2_{M(G)}(t', x) \neq \emptyset$ and $\mathbf{Preds}^2_{M(G)}(t', x) \subseteq \mathbf{Preds}^2_{M(G)}(t', t)$. This case can only happen if x became mergeable after having been traited by the merging algorithm. Thus, during the merging, there has been an atomic merging on t'. Let y be the variable merged on t' such that $y \prec x$. We call G^1 the factbase just before the atomic merging of y on t' and we call G^2 the factbase just after the atomic merging. There exists a term t_0 such that $\mathbf{Preds}^2_{G^1}(t_0, y) \neq \emptyset$ and $\mathbf{Preds}^2_{G^1}(t_0, y) \subseteq \mathbf{Preds}^2_{G^1}(t_0, t')$. The factbase G^1 is on the left of the Figure 6 and the factbase G^2 is on the right (we do not represent the entire graph):

We have $t_0 \prec t' \prec t$ and $t_0 \prec y \prec x$ so x should have been treated by the algorithm after the merging of t' and y so the algorithm will merge x on t. It is a contradiction.

H Institutional and social context of the internship

I did my internship in the GRAPHIK team at the INRIA of Montpellier. GRAPHIK means Graphs for Inferences on Knowledge. GRAPHIK's research work focuses on databases thanks to knowledge representation and reasoning. I was supervised by Jean-François Baget and David Carral, both researchers at INRIA. We did a meeting every week face-to-face at Montpellier and I was in an office with another intern.