Normalization Procedures for Existential Rules that Preserve Chase Termination

Lucas Larroque

GraphIK team

Supervisors: David Carral and Marie-Laure Mugnier

Contents

Contents

A few preliminary notions

This part will focus on the notions of model and universal model.

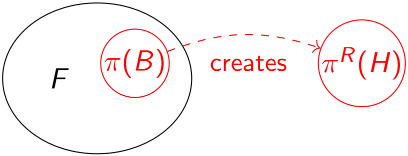
Definition

A **model** of a knowledge base $\mathcal{K} = \langle \mathcal{R}, F \rangle$ is a set of atoms $\mathcal{M}$ such that $F \subseteq \mathcal{M}$ and $\mathcal{M} \vDash \mathcal{R}$.

Definition

A **universal model** $\mathcal{U}$ of a knowledge base $\mathcal{K}$ is a model of $\mathcal{K}$ such that for every model $\mathcal{M}$ of $\mathcal{K}$, $\mathcal{M} \vDash \mathcal{U}$.

The chase aims at finding a universal model for the knowledge base by expanding the initial factbase, via rule application.



Why the equivalent chase?

In this part, we will focus on the **equivalent chase**, which stops whenever all rule applications yield an equivalent factbase.

Theorem (A. Deutsch, A. Nash, and J. B. Remmel., 2008)

The equivalent chase terminates on a knowledge base $\mathcal{K}$ if and only if $\mathcal{K}$ admits a finite universal model.

This theorem lets us use logics instead of algorithmics to prove termination.

In addition, the equivalent chase is the reason we have three normalization procedures and not two.

Contents

Single-piece translation - Piece graph

Definition

The **piece graph** of a rule $B \to H_1[\vec{y_1}] \wedge \ldots \wedge H_n[\vec{y_n}]$ is the following:

- Vertices: Atoms in the head $(H_1[\vec{y_1}], \ldots, H_n[\vec{y_n}])$.
- Edges: Edge between $H_i[\vec{y_i}]$ and $H_j[\vec{y_j}]$ if $\vec{y_i}$ and $\vec{y_j}$ share an existential variable.

A **rule piece** is a connected component of the piece graph.

Example:

$$R = P(x,x) \to \exists y, z, v, w.\ P(x,y) \wedge Q(x,y) \wedge Q(x,w) \wedge P(z,w) \wedge Q(z,v)$$

has the following piece graph:

$$P(x,y) \overset{y}{\text{———}} Q(x,y) \qquad Q(x,w) \overset{w}{\text{———}} P(z,w) \overset{z}{\text{———}} Q(z,v)$$

Single-piece translation - Piece graph

Definition

The **piece graph** of a rule $B \rightarrow H_1[\vec{y_1}] \wedge \ldots \wedge H_n[\vec{y_n}]$ is the following:

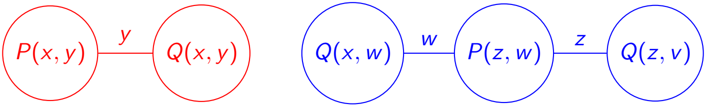
- Vertices: Atoms in the head $(H_1[\vec{y_1}], \ldots, H_n[\vec{y_n}])$.
- Edges: Edge between $H_i[\vec{y_i}]$ and $H_j[\vec{y_j}]$ if $\vec{y_i}$ and $\vec{y_j}$ share an existential variable.

A **rule piece** is a connected component of the piece graph.

Example:

$$R = P(x, x) \rightarrow \exists y, z, v, w.\ P(x, y) \wedge Q(x, y) \wedge Q(x, w) \wedge P(z, w) \wedge Q(z, v)$$

has the following piece graph:



Single-piece translation

Definition

The **single-piece translation** of a rule $R = B \to H$ is the set $sp(R)$, composed of the rules $B \to H_i$ for H_i rule pieces of R.

Example:

$R = P(x,x) \to \exists y, z, v, w.\ P(x,y) \wedge Q(x,y) \wedge Q(x,w) \wedge P(z,w) \wedge Q(z,v)$ gives the rules:

- $P(x,x) \to \exists y.\ P(x,y) \wedge Q(x,y)$
- $P(x,x) \to \exists z, v, w.\ Q(x,w) \wedge P(z,w) \wedge Q(z,v)$

The single-piece translation is an equivalent transformation: $\mathcal{R}$ and $sp(\mathcal{R})$ produce the same models, and in particular, the same universal models. As such, it preserves the termination of the equivalent chase.

Contents

Conservative extension

We now consider normalization procedures that produce rule sets that only have one atom in the head.

It cannot be done without introducing new predicates. Thus, we are looking for conservative extensions.

Definition

$\mathcal{R}'$ is a **conservative extension** of $\mathcal{R}$ if for any factbase F:

- The restriction of any model of $\langle \mathcal{R}', F \rangle$ to the predicates in $\mathcal{R}$ is a model of $\langle \mathcal{R}, F \rangle$.
- Any model $\mathcal{M}$ of $\langle \mathcal{R}, F \rangle$ can be extended into a model of $\langle \mathcal{R}', F \rangle$ that agrees with $\mathcal{M}$ on the predicates in $\mathcal{R}$.

A conservative extension preserves logical interpretation, which lets us use it to solve query entailment for the initial knowledge base.

One-way Atomic Decomposition

Let us have a look at the standard procedure:

Definition

For $R = B \rightarrow (H_1 \wedge \ldots \wedge H_n)[\vec{y}]$ a rule, $\mathit{1ad}(R)$ is composed of:

- $B \rightarrow X_R[\vec{y}]$
- for each $i \leq n$, $X_R[\vec{y}] \rightarrow H_i$

Example:

$$R = \mathit{Manager}(x) \rightarrow \exists y.\ \exists z.\ \mathit{ReportsTo}(x, y) \wedge \mathit{ReportsTo}(z, x)$$

gives the rules:

- $\mathit{Manager}(x) \rightarrow \exists y.\ \exists z.\ X_R(x, y, z)$
- $X_R(x, y, z) \rightarrow \mathit{ReportsTo}(x, y)$
- $X_R(x, y, z) \rightarrow \mathit{ReportsTo}(z, x)$

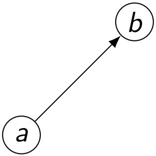
Result regarding the One-way Atomic Decomposition

Theorem

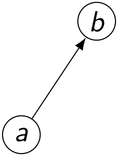
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$$



$$P(x, y) \to \exists z.\ X(y, z)$$
$$X(y, z) \to P(y, z)$$
$$X(y, z) \to P(z, y)$$



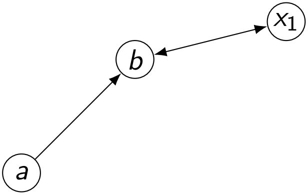
Result regarding the One-way Atomic Decomposition

Theorem

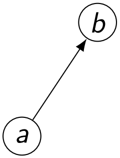
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \land P(z, y)$$



$$P(x, y) \to \exists z.\ X(y, z)$$
$$X(y, z) \to P(y, z)$$
$$X(y, z) \to P(z, y)$$



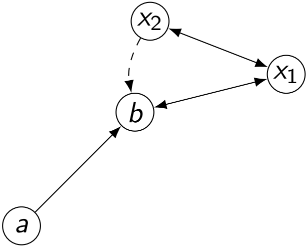
Result regarding the One-way Atomic Decomposition

Theorem

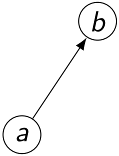
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \land P(z, y)$$



$$P(x, y) \to \exists z.\ X(y, z)$$
$$X(y, z) \to P(y, z)$$
$$X(y, z) \to P(z, y)$$



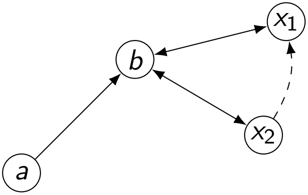
Result regarding the One-way Atomic Decomposition

Theorem

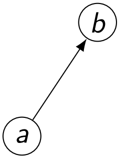
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \rightarrow \exists z.\ P(y, z) \wedge P(z, y)$$



$$P(x, y) \rightarrow \exists z.\ X(y, z)$$
$$X(y, z) \rightarrow P(y, z)$$
$$X(y, z) \rightarrow P(z, y)$$



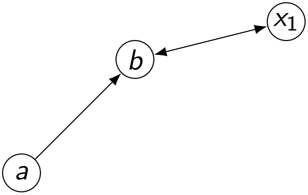
Result regarding the One-way Atomic Decomposition

Theorem

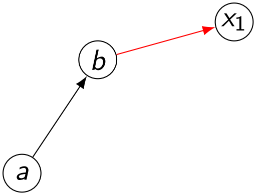
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$$



$$\boxed{P(x, y) \to \exists z.\ X(y, z)}$$
$$X(y, z) \to P(y, z)$$
$$X(y, z) \to P(z, y)$$



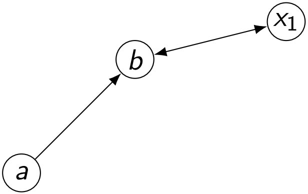
Result regarding the One-way Atomic Decomposition

Theorem

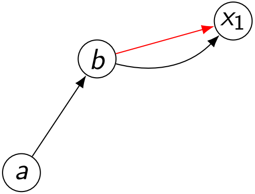
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$$



$$P(x, y) \to \exists z.\ X(y, z)$$
$$\boxed{X(y, z) \to P(y, z)}$$
$$X(y, z) \to P(z, y)$$



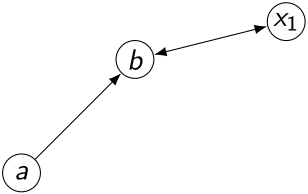
Result regarding the One-way Atomic Decomposition

Theorem

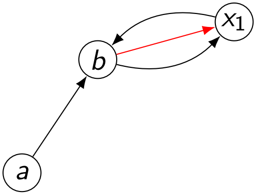
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x,y) \rightarrow \exists z.\ P(y,z) \wedge P(z,y)$$



$$P(x,y) \rightarrow \exists z.\ X(y,z)$$
$$X(y,z) \rightarrow P(y,z)$$
$$\boxed{X(y,z) \rightarrow P(z,y)}$$



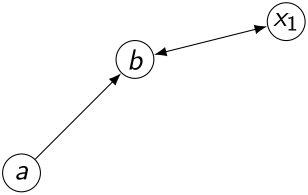
Result regarding the One-way Atomic Decomposition

Theorem

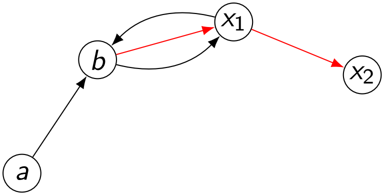
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x,y) \to \exists z.\ P(y,z) \land P(z,y)$$



$$\boxed{P(x,y) \to \exists z.\ X(y,z)}$$
$$X(y,z) \to P(y,z)$$
$$X(y,z) \to P(z,y)$$



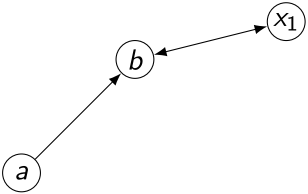
Result regarding the One-way Atomic Decomposition

Theorem

The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

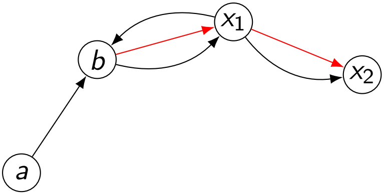
$$P(x,y) \to \exists z.\ P(y,z) \wedge P(z,y)$$



$$P(x,y) \to \exists z.\ X(y,z)$$

$$\boxed{X(y,z) \to P(y,z)}$$

$$X(y,z) \to P(z,y)$$



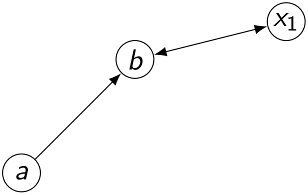
Result regarding the One-way Atomic Decomposition

Theorem

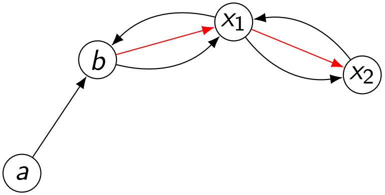
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$$



$$P(x, y) \to \exists z.\ X(y, z)$$
$$X(y, z) \to P(y, z)$$
$$\boxed{X(y, z) \to P(z, y)}$$



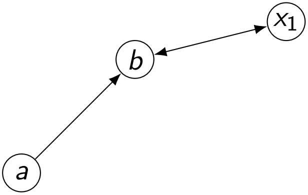
Result regarding the One-way Atomic Decomposition

Theorem

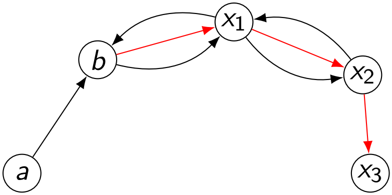
The one-way atomic decomposition does not preserve the termination of the equivalent chase.

Counter-example:

$$P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$$



$$\boxed{P(x, y) \to \exists z.\ X(y, z)}$$
$$X(y, z) \to P(y, z)$$
$$X(y, z) \to P(z, y)$$



Universal-conservative extension

Obviously, conservative extensions are not enough. We introduce universal-conservative extensions:

Definition

$\mathcal{R}'$ is a **universal-conservative extension** of $\mathcal{R}$ if for any factbase F:

- The restriction of a UM of $\langle \mathcal{R}', F \rangle$ to the predicates in $\mathcal{R}$ is a UM of $\langle \mathcal{R}, F \rangle$.
- Any UM $\mathcal{M}$ of $\langle \mathcal{R}, F \rangle$ can be extended into a UM of $\langle \mathcal{R}', F \rangle$ that agrees with $\mathcal{M}$ on the predicates in $\mathcal{R}$.

A normalization procedure that produces universal-conservative extensions will preserve the termination of the equivalent chase, using the following theorem we mentioned previously:

Theorem (A. Deutsch, A. Nash, and J. B. Remmel., 2008)

The equivalent chase terminates on a knowledge base $\mathcal{K}$ if and only if $\mathcal{K}$ admits a finite universal model.

Contents

Two-way atomic decomposition

Definition

For $R = B \to (H_1 \wedge \ldots \wedge H_n)[\vec{y}]$ a rule, $\mathit{2ad}(R)$ is composed of:

- $B \to X_R[\vec{y}]$
- for each $i \leq n$, $X_R[\vec{y}] \to H_i$
- $H_1 \wedge \ldots \wedge H_n \to X_R[\vec{y}]$

Example:

$$R = \mathit{Manager}(x) \to \exists y.\, \exists z.\, \mathit{ReportsTo}(x, y) \wedge \mathit{ReportsTo}(z, x)$$

gives the rules:

- $\mathit{Manager}(x) \to \exists y.\, \exists z.\, X_R(x, y, z)$
- $X_R(x, y, z) \to \mathit{ReportsTo}(x, y)$
- $X_R(x, y, z) \to \mathit{ReportsTo}(z, x)$
- $\mathit{ReportsTo}(x, y) \wedge \mathit{ReportsTo}(z, x) \to X_R(x, y, z)$

Result, extension

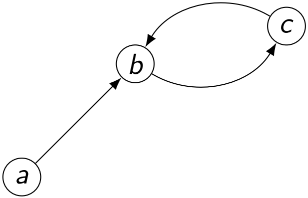
Theorem

The two-way atomic decomposition produces universal-conservative extensions.

Proof. First, we need to define our extension. We will extend our models by satisfying the rules of the form $\bigwedge_i H_i \to X_R$ whenever it is possible.

Example: $P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$

2ad($\mathcal{R}$):
$P(x, y) \to \exists z.\ X(y, z)$
$X(y, z) \to P(y, z)$
$X(y, z) \to P(z, y)$
$P(y, z) \wedge P(z, y) \to X(y, z)$



Result, extension

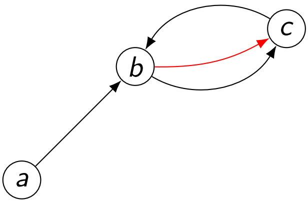
Theorem

The two-way atomic decomposition produces universal-conservative extensions.

Proof. First, we need to define our extension. We will extend our models by satisfying the rules of the form $\bigwedge_i H_i \to X_R$ whenever it is possible.

Example: $P(x,y) \to \exists z.\ P(y,z) \wedge P(z,y)$

$\mathit{2ad}(\mathcal{R})$:
$P(x,y) \to \exists z.\ X(y,z)$
$X(y,z) \to P(y,z)$
$X(y,z) \to P(z,y)$
$\boxed{P(y,z) \wedge P(z,y) \to X(y,z)}$



Result, extension

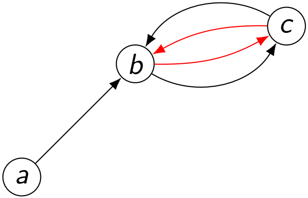
Theorem

The two-way atomic decomposition produces universal-conservative extensions.

Proof. First, we need to define our extension. We will extend our models by satisfying the rules of the form $\bigwedge_i H_i \to X_R$ whenever it is possible.

Example: $P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)$

$2ad(\mathcal{R})$:
$P(x, y) \to \exists z.\ X(y, z)$
$X(y, z) \to P(y, z)$
$X(y, z) \to P(z, y)$
$\boxed{P(y, z) \wedge P(z, y) \to X(y, z)}$



Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part		
	$\mathcal{K}_{2ad}$	$\mathcal{K}$
UM		
M		

Second part		
	$\mathcal{K}_{2ad}$	$\mathcal{K}$
UM		
M		

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

	$\mathcal{K}_{2ad}$	$\mathcal{K}$
UM	$\mathcal{V}$	
M		

Second part

	$\mathcal{K}_{2ad}$	$\mathcal{K}$
UM		
M		

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$\begin{tikzcd} & \mathcal{K}_{2ad} & \mathcal{K} \\ \text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] & \mathcal{U} \\ \text{M} & & \end{tikzcd}$$

Second part

$$\begin{tikzcd} & \mathcal{K}_{2ad} & \mathcal{K} \\ \text{UM} & & \\ \text{M} & & \end{tikzcd}$$

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$\begin{tikzcd} & \mathcal{K}_{2ad} & \mathcal{K} \\ \text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] & \mathcal{U} \\ \text{M} & & \mathcal{M} \end{tikzcd}$$

Second part

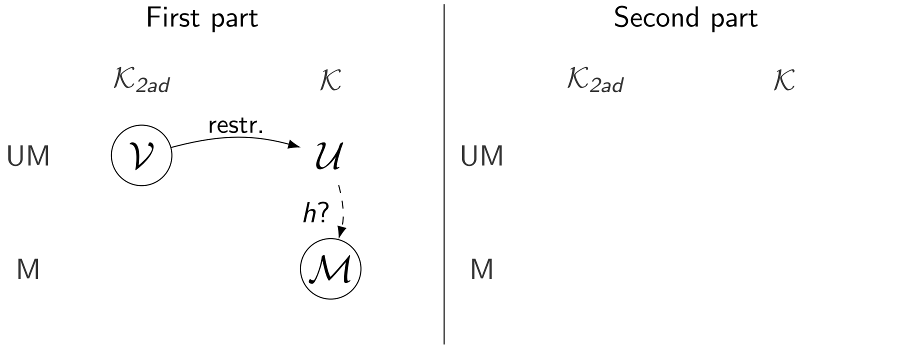
$$\begin{array}{ccc} & \mathcal{K}_{2ad} & \mathcal{K} \\ \text{UM} & & \\ \text{M} & & \end{array}$$

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

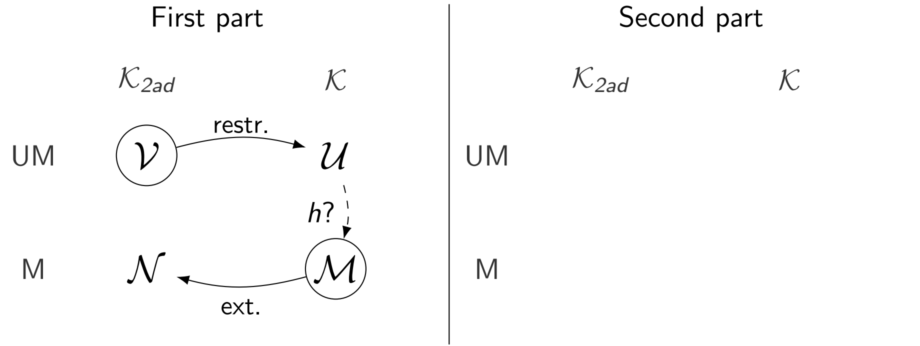


Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.



Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] \arrow[d, "h"] & \mathcal{U} \arrow[d, dashed, "h?"'] \\
\text{M} & \mathcal{N} & \mathcal{M} \arrow[l, bend left, "\text{ext.}"]
\end{tikzcd}$$

Second part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & & \\
\text{M} & &
\end{tikzcd}$$

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$
\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] \arrow[d, "h"', ""{name=H}] & \mathcal{U} \arrow[d, dashed, "h?"', ""{name=HQ}] \\
\text{M} & \mathcal{N} & \mathcal{M} \arrow[l, bend left, "\text{ext.}"]
\arrow[from=H, to=HQ, dashed, "\text{restr.}"]
\end{tikzcd}
$$

Second part

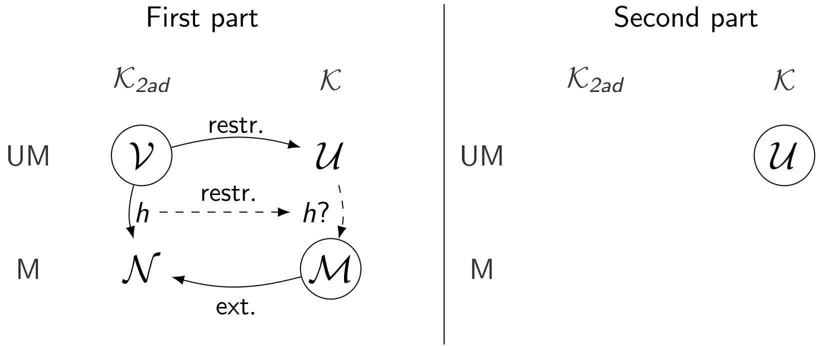
$$
\begin{array}{ccc}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & & \\
\text{M} & &
\end{array}
$$

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.



Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] \arrow[d, "h"'] & \mathcal{U} \arrow[d, dashed, "h?"] \\
\text{M} & \mathcal{N} & \mathcal{M} \arrow[l, bend left, "\text{ext.}"]
\end{tikzcd}$$

Second part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} & \mathcal{U} \arrow[l, bend right, "\text{ext.}"'] \\
\text{M} & &
\end{tikzcd}$$

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] \arrow[d, "h"'] \arrow[dr, phantom, "\xrightarrow{\text{restr.}}"] & \mathcal{U} \arrow[d, dashed, "h?"'] \\
\text{M} & \mathcal{N} & \mathcal{M} \arrow[l, bend left, "\text{ext.}"]
\end{tikzcd}$$

Second part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} & \mathcal{U} \arrow[l, bend right, "\text{ext.}"'] \\
\text{M} & \mathcal{N} &
\end{tikzcd}$$

Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$
\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[r, "\text{restr.}"] \arrow[d, "h"', ""{name=A}] & \mathcal{U} \arrow[d, dashed, "h?", ""'{name=B}] \\
\text{M} & \mathcal{N} & \mathcal{M} \arrow[l, "\text{ext.}"] \arrow[from=A, to=B, dashed, "\text{restr.}"]
\end{tikzcd}
$$

Second part

$$
\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[d, dashed, "h?"] & \mathcal{U} \arrow[l, "\text{ext.}"'] \\
\text{M} & \mathcal{N} &
\end{tikzcd}
$$

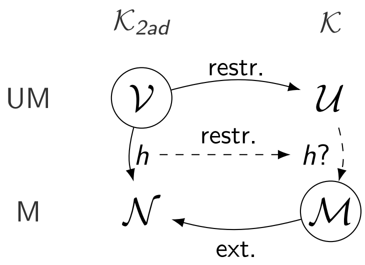
Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

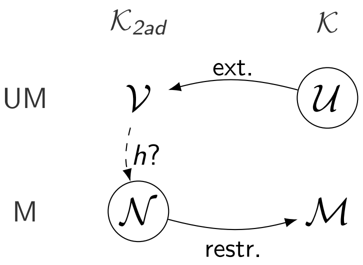
What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part



Second part



Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[r, bend left, "\text{restr.}"] \arrow[d, bend right, "h"'] \arrow[dr, phantom, "\overset{\text{restr.}}{\dashrightarrow}"] & \mathcal{U} \arrow[d, dashed, bend left, "h?"] \\
\text{M} & \mathcal{N} & \mathcal{M} \arrow[l, bend left, "\text{ext.}"]
\end{tikzcd}$$

Second part

$$\begin{tikzcd}
 & \mathcal{K}_{2ad} & \mathcal{K} \\
\text{UM} & \mathcal{V} \arrow[d, dashed, bend right, "h?"] & \mathcal{U} \arrow[l, bend right, "\text{ext.}"'] \arrow[d, bend left, "h"] \\
\text{M} & \mathcal{N} \arrow[r, bend right, "\text{restr.}"'] & \mathcal{M}
\end{tikzcd}$$

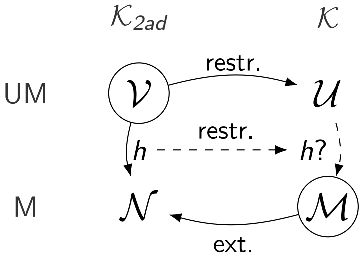
Proof sketch

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$.

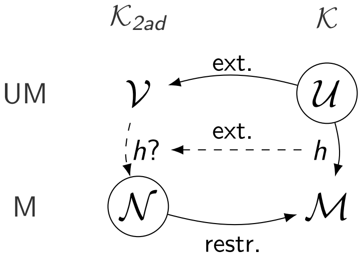
What we want to prove

1. The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.
2. Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

First part



Second part



Contents

The oblivious and semi-oblivious chase

Those are the simplest of the chase variants considered:

- Oblivious chase: One can apply a trigger if it has not been applied yet.
- Semi-oblivious chase: One can apply a trigger if it has not been applied **with the same frontier**.

For those two chase variants, every transformation preserve the termination. The proof techniques are also the same for both variants.

	O	**SO**
Single-piece translation	✓	✓
One-way atomic decomposition	✓	✓
Two-way atomic decomposition	✓	✓

Proof sketch - Single-piece translation

The oblivious chase (resp. semi-oblivious chase) produces the same set of atom after running on a knowledge base regardless of the derivation.

Definition

For $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, and $\mathcal{K}$ a knowledge base, consider $\mathcal{D}$ an arbitrary fair derivation from $\mathcal{K}$. We define $Ch_{\mathbf{X}}(\mathcal{K}) = res(\mathcal{D})$.

Thus, we only have to prove the following:

What we want to prove

For $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, F a factbase and $\mathcal{R}$ a ruleset, $Ch_{\mathbf{X}}(\langle sp(\mathcal{R}), F\rangle)$ is injectively embedded in $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$.

We show this by induction on an arbitrary derivation from $\langle sp(\mathcal{R}), F\rangle$. With this result, if the chase variant terminates on the initial ruleset, $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$ is finite, and so is $Ch_{\mathbf{X}}(\langle sp(\mathcal{R}), F\rangle)$.

Proof sketch - Atomic decompositions

We use the same idea, but we must remove the new predicates introduced by the decomposition.

Definition

For $\mathcal{R}$ a ruleset and A a set of atoms, we define $\gamma_{\mathcal{R}}(A)$ as the maximal subset of A such that every atom uses only predicates in $\mathcal{R}$.

Thus we want:

What we want to prove

For $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, $f \in \{\mathit{1ad}, \mathit{2ad}\}$, F a factbase and $\mathcal{R}$ a ruleset, $\gamma_R(Ch_{\mathbf{X}}(\langle f(\mathcal{R}), F\rangle))$ is injectively embedded in $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$.

This result is proven the same way previous result is.

Contents

The restricted chase

In the restricted chase, we cannot apply a trigger if its output can be retracted in the factbase.

No normalization procedure preserves the termination of the restricted chase.

	R
Single-piece translation	✗
One-way atomic decomposition	✗
Two-way atomic decomposition	✗

The proofs are just counter examples. Note that the counter example we saw previously, $P(x, y) \rightarrow \exists z.\ P(y, z) \wedge P(z, y)$, works for both atomic decompositions here.

The Datalog-first restricted chase

Recall the rules in the two-way atomic decomposition of $B \rightarrow \bigwedge_i H_i$:

$$\left.\begin{array}{l} \bullet\ B \rightarrow X_R \\ \bullet\ \forall i,\ X_R \rightarrow H_i \\ \bullet\ H_1 \wedge \ldots \wedge H_n \rightarrow X_R \end{array}\right\} \underline{\text{Datalog}}$$

We thus consider the **Datalog-first restricted chase**, in which we can apply an existential trigger only if there is no Datalog rule applicable.

	R	Df-**R**
Single-piece translation	✗	✗
One-way atomic decomposition	✗	✗
Two-way atomic decomposition	✗	✓

Df-**R**-chase and Two-way atomic decomposition

Theorem

The two-way atomic decomposition preserves the termination of the Datalog-first restricted chase.

Proof. The proof relies on two concepts.

First, the proof will be done by **contrapositive**: we want to show that if there is a non-terminating derivation from $\langle 2ad(\mathcal{R}), F\rangle$, there is a non-terminating derivation from $\langle \mathcal{R}, F\rangle$.

This will be done using the other fundamental idea of this proof: we will break down Datalog-first derivations between the existential triggers and the Datalog derivations.

Existential triggers are triggers that introduce an existential variable, whereas Datalog derivations are subderivations composed exclusively of triggers that introduce no existential variables. Thus, they are always finite.

Lemma introduction

What we want

For $\mathcal{D}^{ad} = (\emptyset, F^{ad}), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \ldots$ from $\langle 2ad(\mathcal{R}), F \rangle$, there is a Df-**R**-derivation $\mathcal{D} = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \ldots$ from $\langle \mathcal{R}, F \rangle$ such that if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is fair.

Assuming we have this lemma, if there is an infinite yet fair derivation $\mathcal{D}^{ad}$ from $\langle 2ad(\mathcal{R}), F \rangle$, there is a derivation $\mathcal{D}$ from $\langle \mathcal{R}, F \rangle$ such that:

- $\mathcal{D}$ has an infinite number of existential triggers: it is **infinite**.
- $\mathcal{D}$ is **fair**.

By contrapositive, if the Datalog-first restricted chase terminates on $\langle \mathcal{R}, F \rangle$, it also must terminate on $\langle 2ad(\mathcal{R}), F \rangle$: the two-way atomic decomposition preserves the termination of the Datalog-first restricted chase.

Full lemma

What we want

For $\mathcal{D}^{ad} = (\emptyset, F^{ad}), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \ldots$ from $\langle 2ad(\mathcal{R}), F\rangle$, there is a Df-**R**-derivation $\mathcal{D} = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \ldots$ from $\langle \mathcal{R}, F\rangle$ such that if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is fair.

To prove the lemma by induction, we need to add a few things to make the induction hypothesis stronger.

For $\mathcal{D}^{ad} = (\emptyset, F^{ad}), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \ldots$ from $\langle 2ad(\mathcal{R}), F\rangle$, there is a Df-**R**-derivation $\mathcal{D} = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \ldots$ from $\langle \mathcal{R}, F\rangle$ such that:

- There is an isomorphism h from $\gamma_{\mathcal{R}}(res(\mathcal{D}^{ad}))$ to $res(\mathcal{D})$.
- Forall i, if $t_i^{ad} = (B \rightarrow X_R, \pi)$, then $t_i = (B \rightarrow H, h \circ \pi)$, with $(B \rightarrow X_R) \in 2ad(B \rightarrow H)$.
- Forall i, $\mathcal{D}_i$ is $h(\mathcal{D}_i^{ad})$ in which we remove every trigger using a rule in $2ad(\mathcal{R}^{\exists})$, with $\mathcal{R}^{\exists}$ being the set of the non-Datalog rules of $\mathcal{R}$.

In addition, if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is fair.

Full lemma

What we want

For $\mathcal{D}^{ad} = (\emptyset, F^{ad}), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \dots$ from $\langle 2ad(\mathcal{R}), F\rangle$, there is a Df-**R**-derivation $\mathcal{D} = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \dots$ from $\langle \mathcal{R}, F\rangle$ such that if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is fair.

To prove the lemma by induction, we need to add a few things to make the induction hypothesis stronger.

For $\mathcal{D}^{ad} = (\emptyset, F^{ad}), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \dots$ from $\langle 2ad(\mathcal{R}), F\rangle$, there is a Df-**R**-derivation $\mathcal{D} = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \dots$ from $\langle \mathcal{R}, F\rangle$ such that:

- There is an isomorphism h from $\gamma_{\mathcal{R}}(res(\mathcal{D}^{ad}))$ to $res(\mathcal{D})$.
- Forall i, if $t_i^{ad} = (B \to X_R, \pi)$, then $t_i = (B \to H, h \circ \pi)$, with $(B \to X_R) \in 2ad(B \to H)$.
- Forall i, $\mathcal{D}_i$ is $h(\mathcal{D}_i^{ad})$ in which we remove every trigger using a rule in $2ad(\mathcal{R}^{\exists})$, with $\mathcal{R}^{\exists}$ being the set of the non-Datalog rules of $\mathcal{R}$.

In addition, if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is fair.

Fairness proof

Lemma (can be proven by induction)

For a derivation $\mathcal{D}^{ad}$ from $\langle 2ad(\mathcal{R}), F\rangle$, there is a derivation $\mathcal{D}$ from $\langle \mathcal{R}, F\rangle$ such that there is an isomorphism from $\gamma_{\mathcal{R}}(res(\mathcal{D}^{ad}))$ to $res(\mathcal{D})$.

Now we want to prove that if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is fair.

- $\mathcal{D}^{ad}$ fair implies that $res(\mathcal{D}^{ad})$ is a universal model of $\langle 2ad(\mathcal{R}), F\rangle$.
- Since the two-way atomic decomposition is a universal-conservative extension, then $\gamma_{\mathcal{R}}(res(\mathcal{D}^{ad}))$ is a universal model of $\langle \mathcal{R}, F\rangle$.
- Since $\gamma_{\mathcal{R}}(res(\mathcal{D}^{ad}))$ is isomorphic to $res(\mathcal{D})$, $res(\mathcal{D})$ is a universal model of $\langle \mathcal{R}, F\rangle$.
- Thus, every rule of $\mathcal{R}$ is satisfied in $res(\mathcal{D})$: $\mathcal{D}$ is fair.

Which concludes the proof of the lemma, and the proof of the theorem.

Contents

Conclusion

We introduced a new notion to characterize transformations that preserve the termination of the equivalent chase, and new proof techniques, notably the one regarding the Datalog-first restricted chase.

Compilation of the results obtained:

	O	**SO**	**R**	Df-**R**	**E**
Single-piece translation	✓	✓	✗	✗	✓
One-way atomic decomposition	✓	✓	✗	✗	✗
Two-way atomic decomposition	✓	✓	✗	✓	✓

Thank you for listening!

Contents

Existential rules

We work in the setting of first-order logic (FOL) without the equality.

Definition

An **(existential) rule** is a logical formula of the form

$$\forall \vec{x} \forall \vec{y}.\ B[\vec{x}, \vec{y}] \rightarrow \exists \vec{z}.\ H[\vec{x}, \vec{z}]$$

with the **body** $B[\vec{x}, \vec{y}]$ and the **head** $H[\vec{x}, \vec{z}]$ conjunctions of atoms.

For instance:

$$\forall x.\ \ \mathit{Manager}(x)\ \rightarrow \exists y.\ \exists z.\ \ \mathit{ReportsTo}(x, y) \wedge \mathit{ReportsTo}(z, x)$$

Next, we denote an existential rule $B \rightarrow H$, omitting the variables.

Existential rules

We work in the setting of first-order logic (FOL) without the equality.

Definition

An **(existential) rule** is a logical formula of the form

$$\forall \vec{x} \forall \vec{y}.\ B[\vec{x}, \vec{y}] \rightarrow \exists \vec{z}.\ H[\vec{x}, \vec{z}]$$

with the **body** $B[\vec{x}, \vec{y}]$ and the **head** $H[\vec{x}, \vec{z}]$ conjunctions of atoms.

For instance:

$$\forall x.\ \underbrace{\boxed{Manager(x)}}_{\text{body}} \rightarrow \exists y.\ \exists z.\quad ReportsTo(x, y) \wedge ReportsTo(z, x)$$

Next, we denote an existential rule $B \rightarrow H$, omitting the variables.

Existential rules

We work in the setting of first-order logic (FOL) without the equality.

Definition

An **(existential) rule** is a logical formula of the form

$$\forall \vec{x} \forall \vec{y}.\ B[\vec{x}, \vec{y}] \rightarrow \exists \vec{z}.\ H[\vec{x}, \vec{z}]$$

with the **body** $B[\vec{x}, \vec{y}]$ and the **head** $H[\vec{x}, \vec{z}]$ conjunctions of atoms.

For instance:

$$\forall x.\ \underbrace{\boxed{Manager(x)}}_{\text{body}} \rightarrow \exists y.\ \exists z.\ \underbrace{\boxed{ReportsTo(x, y) \wedge ReportsTo(z, x)}}_{\text{head}}$$

Next, we denote an existential rule $B \rightarrow H$, omitting the variables.

Knowledge bases and Models

Definition

A **factbase** is a set of atoms. A **knowledge base** $\mathcal{K} = \langle \mathcal{R}, F \rangle$ is composed of a factbase F and a rule set $\mathcal{R}$.

Representation using graphs:
$\mathcal{M} = \{Manager(Bob), ReportsTo(Alice, Bob), ReportsTo(Bob, Carole)\}$
has for representation: $\text{Alice} \longrightarrow \boxed{\text{Bob}} \longrightarrow \text{Carole}$

Definition

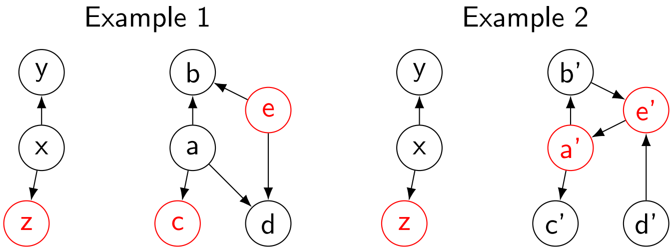
A **model** of a knowledge base $\mathcal{K} = \langle \mathcal{R}, F \rangle$ is a set of atoms $\mathcal{M}$ such that $F \subseteq \mathcal{M}$ and $\mathcal{M}$ satisfies every rule in $\mathcal{R}$.

Example: $F = \{Manager(Bob)\}$ and
$\mathcal{R} = \{\forall x.\ Manager(x) \rightarrow \exists y.\ \exists z.\ ReportsTo(x, y) \wedge ReportsTo(z, x)\}$.

Homomorphisms

Definition

A **homomorphism** from F to G is a function mapping the variables of F to the terms of G such that $h(F) \subseteq G$.



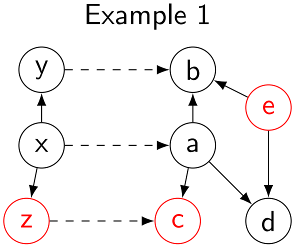
Homomorphism theorem

There is a homomorphism from F to G if and only if $G \vDash F$.

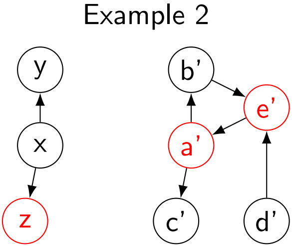
Homomorphisms

Definition

A **homomorphism** from F to G is a function mapping the variables of F to the terms of G such that $h(F) \subseteq G$.



This is a homomorphism



Homomorphism theorem

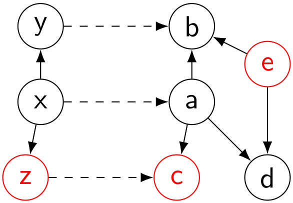
There is a homomorphism from F to G if and only if $G \vDash F$.

Homomorphisms

Definition

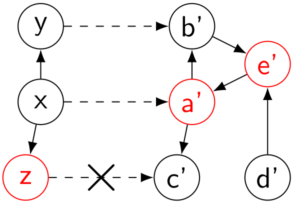
A **homomorphism** from F to G is a function mapping the variables of F to the terms of G such that $h(F) \subseteq G$.

Example 1



This is a homomorphism

Example 2



This is not a homomorphism

Homomorphism theorem

There is a homomorphism from F to G if and only if $G \vDash F$.

Query entailment and Universal models

Definition

A **boolean conjunctive query (BCQ)** is an existentially closed conjunction of atoms.

Note that a BCQ can be seen as a finite factbase.

The problem of BCQ entailment

Given a BCQ q and a knowledge base $\mathcal{K}$, does $\mathcal{M} \vDash q$ hold for every model $\mathcal{M}$ of $\mathcal{K}$?

If that is the case, we write $\mathcal{K} \vDash q$.

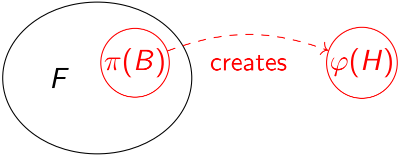
Definition

A **universal model** $\mathcal{U}$ of a knowledge base $\mathcal{K}$ is a model of $\mathcal{K}$ such that for every model $\mathcal{M}$ of $\mathcal{K}$, $\mathcal{M} \vDash \mathcal{U}$.

If $\mathcal{U} \vDash q$ then $\mathcal{K} \vDash q$, because for each $\mathcal{M}$ model of $\mathcal{K}$, $\mathcal{M} \vDash \mathcal{U} \vDash q$ and we conclude by transitivity of $\vDash$

The chase algorithm

The chase aims at finding a universal model for the knowledge base by expanding the initial factbase, via rule application.



Rule application: $R = B \to H$ is a rule, F a factbase, $\pi : B \to F$ a homomorphism and φ extends π by assigning a fresh variable to each existential variable in H.

In this presentation, we will focus our interest on a specific variant, the **equivalent chase**, which stops whenever all rule applications yield an equivalent factbase.

Theorem (A. Deutsch, A. Nash, and J. B. Remmel., 2008)

The equivalent chase terminates on a knowledge base $\mathcal{K}$ if and only if $\mathcal{K}$ admits a finite universal model.

Contents

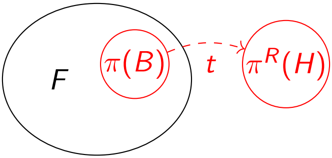
Triggers

We only consider the equivalent chase (or **E**-chase) for the sake of this presentation.

For π a homomorphism and R a rule, we define π^R the extension of π s.t. for every existential variable z in R, $\pi^R(z) = z_{(R,\pi)}$ with $z_{(R,\pi)}$ fresh and unique for z, π and R.

Definition

Let F be a factbase and $R = B \to H$ an existential rule. The pair $t = (R, \pi)$ is an **E**-applicable **trigger** for F if π is a homomorphism from B to F and there is no homomorphism from $F \cup \pi^R(H)$ to F.

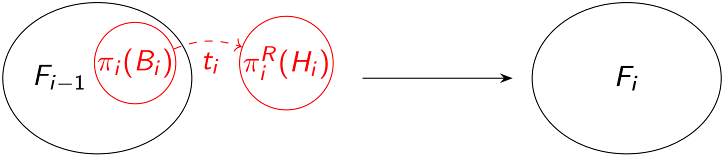


Derivations

Definition

An **E-derivation** from a knowledge base $\mathcal{K} = \langle \mathcal{R}, F \rangle$ is a sequence of the form $\mathcal{D} = (\emptyset, F_0), (t_1, F_1), (t_2, F_2), \ldots$ such that:

- $F_0 = F$ and for all $t_i = (R_i, \pi_i)$, $R_i \in \mathcal{R}$
- for all i, t_i is an **E**-applicable trigger for F_{i-i} and $F_i = F_{i-1} \cup \pi^R(H)$.



The result of a derivation $\mathcal{D}$, denoted with $res(\mathcal{D})$, is the set $\bigcup_i F_i$.

Termination

Definition

- We say that a **E**-derivation $\mathcal{D}$ from $\mathcal{K}$ is **fair** if no trigger is **E**-applicable on $res(\mathcal{D})$.
- We say that the **E**-chase **terminates** on a knowledge base $\mathcal{K}$ if every **E**-derivation from $\mathcal{K}$ is fair and finite.

Contents

Proof part 1

Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a knowledge base and let $\mathcal{K}_{2ad} = \langle 2ad(\mathcal{R}), F \rangle$. We can now show the first point:

Result we want

The restriction of a UM of $\mathcal{K}_{2ad}$ to $\mathcal{R}$ is a UM of $\mathcal{K}$.

- Let $\mathcal{V}$ be a UM of $\mathcal{K}_{2ad}$
- Let $\mathcal{U}$ be its restriction to $\mathcal{R}$. Show that $\mathcal{U}$ is a UM.
- Let $\mathcal{M}$ be a model of $\mathcal{K}$ and $\mathcal{N}$ be its extension to $\mathcal{K}_{2ad}$.
- $\mathcal{V}$ is a UM so let h be a homomorphism from $\mathcal{V}$ to $\mathcal{N}$.
- $\mathcal{U} \subseteq \mathcal{V}$ and $\mathcal{M} \subseteq \mathcal{N}$ so h is a mapping from $\mathcal{U}$ to $\mathcal{M}$.
- Let $P(\vec{y}) \in \mathcal{U}$. Show that $P(h(\vec{y})) \in \mathcal{M}$.
- Since h is a homomorphism, $P(h(\vec{y})) \in \mathcal{N}$.
- Since $\mathcal{U}$ is a model of $\mathcal{K}$, P is a predicate in $\mathcal{R}$.
- Thus, $P(h(\vec{y})) \in \mathcal{M}$, so $\mathcal{U}$ is a UM.

Proof part 2

Let us now continue with the second part of the result:

Result we want

Any UM of $\mathcal{K}$ can be extended into a UM of $\mathcal{K}_{2ad}$.

- Let $\mathcal{U}$ be a UM of K
- Let $\mathcal{V}$ be $\mathcal{U}$'s extension as defined. Show that $\mathcal{V}$ is a UM.
- Let $\mathcal{N}$ be a model of $\mathcal{K}_{2ad}$ and $\mathcal{M}$ its restriction.
- $\mathcal{U}$ is a UM so let h be a homomorphism from $\mathcal{U}$ to $\mathcal{M}$.
- Let $P(\vec{y}) \in \mathcal{V}$. Show that $P(h(\vec{y})) \in \mathcal{N}$. Two cases:
- If P is in $\mathcal{R}$, $P(\vec{y}) \in \mathcal{U}$ so $P(h(\vec{y})) \in \mathcal{M}$. $\mathcal{M} \subseteq \mathcal{N}$ so $P(h(\vec{y})) \in \mathcal{N}$.
- Else, if $P = X_R$ does not appear in $\mathcal{R}$:
- Rules $X_R \to H_i$ are satisfied by $\mathcal{V}$ so for all i, $H_i(\vec{y_i}) \in \mathcal{V}$.
- H_i are in $\mathcal{R}$, so for all i, $H_i(\vec{y_i}) \in \mathcal{N}$.
- Since $\mathcal{N}$ is a model, $\bigwedge_i H_i \to X_R$ is satisfied.
- Thus, either case, $P(h(\vec{y})) \in \mathcal{N}$.