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

Internship supervised by:
Jean-François Baget and David Carral
INRIA, Montpellier, GRAPHIK team

Maël Abily

2021

Entailment: a fundamental problem in knowledge representation

Knowledge base $K =$

Rules	**Facts**
$\alpha = Mother(x) \rightarrow Parent(x)$ $\beta = Parent(x) \rightarrow \exists y.IsTheParentOf(x, y)$ $\gamma = IsTheParentOf(x, y) \rightarrow Human(y)$	$Mother(\textbf{Marie})$

Entailment: a fundamental problem in knowledge representation

Knowledge base $K =$

Rules	**Facts**
$\alpha = Mother(x) \rightarrow Parent(x)$ $\beta = Parent(x) \rightarrow \exists y.IsTheParentOf(x, y)$ $\gamma = IsTheParentOf(x, y) \rightarrow Human(y)$	$Mother(\textbf{Marie})$

Query $Q = \{IsTheParentOf(\textbf{Marie}, x), Human(x)\}$

Knowledge base K =

Rules	**Facts**
$\alpha = Mother(x) \rightarrow Parent(x)$ $\beta = Parent(x) \rightarrow \exists y.IsTheParentOf(x, y)$ $\gamma = IsTheParentOf(x, y) \rightarrow Human(y)$	$Mother(\textbf{Marie})$

Query $Q = \{IsTheParentOf(\textbf{Marie}, x), Human(x)\}$

Conjunctive query entailment problem

Determine if the knowledge base K entails the query Q

Entailment: a fundamental problem in knowledge representation

Conjunctive query entailment problem

Determine if the knowledge base K entails the query Q

Definition

A factbase U is a universal model for a knowledge base K if for every query Q, K entails Q if and only if $U \vDash Q$.

One of the solution to resolve the conjunctive query entailment problem: The Chase

Knowledge base $K =$

$\alpha = Mother(x) \rightarrow Parent(x)$ $\beta = Parent(x) \rightarrow \exists y.IsTheParentOf(x, y)$ $\gamma = IsTheParentOf(x, y) \rightarrow Human(y)$	$Mother(\mathbf{Marie})$

Query $Q = \{IsTheParentOf(\mathbf{Marie}, x), Human(x)\}$

One of the solution to resolve the conjunctive query entailment problem: The Chase

Knowledge base $K =$

$\alpha = \mathit{Mother}(x) \rightarrow \mathit{Parent}(x)$ $\beta = \mathit{Parent}(x) \rightarrow \exists y.\mathit{IsTheParentOf}(x, y)$ $\gamma = \mathit{IsTheParentOf}(x, y) \rightarrow \mathit{Human}(y)$	$\mathit{Mother}(\textbf{Marie})$ $\mathit{Parent}(\textbf{Marie})$

Query $Q = \{\mathit{IsTheParentOf}(\textbf{Marie}, x), \mathit{Human}(x)\}$
Trigger $t_1 = (\alpha, \{x \mapsto \textbf{Marie}\})$

One of the solution to resolve the conjunctive query entailment problem: The Chase

Knowledge base $K =$

$\alpha = \mathit{Mother}(x) \rightarrow \mathit{Parent}(x)$	$\mathit{Mother}(\mathbf{Marie})$
$\beta = \mathit{Parent}(x) \rightarrow \exists y.\mathit{IsTheParentOf}(x, y)$	$\mathit{Parent}(\mathbf{Marie})$
$\gamma = \mathit{IsTheParentOf}(x, y) \rightarrow \mathit{Human}(y)$	$\mathit{IsTheParentOf}(\mathbf{Marie}, y_0)$

Query $Q = \{\mathit{IsTheParentOf}(\mathbf{Marie}, x), \mathit{Human}(x)\}$
Trigger $t_1 = (\alpha, \{x \mapsto \mathbf{Marie}\})$
Trigger $t_2 = (\beta, \{x \mapsto \mathbf{Marie}\})$, $y_0 = f^y_\beta(\mathbf{Marie})$

One of the solution to resolve the conjunctive query entailment problem: The Chase

Knowledge base $K =$

$\alpha = Mother(x) \rightarrow Parent(x)$ $\beta = Parent(x) \rightarrow \exists y.IsTheParentOf(x, y)$ $\gamma = IsTheParentOf(x, y) \rightarrow Human(y)$	$Mother(\mathbf{Marie})$ $Parent(\mathbf{Marie})$ $IsTheParentOf(\mathbf{Marie}, y_0)$ $Human(y_0)$

Query $Q = \{IsTheParentOf(\mathbf{Marie}, x), Human(x)\}$
Trigger $t_1 = (\alpha, \{x \mapsto \mathbf{Marie}\})$
Trigger $t_2 = (\beta, \{x \mapsto \mathbf{Marie}\})$, $y_0 = f_\beta^y(\mathbf{Marie})$
Trigger $t_3 = (\gamma, \{x \mapsto \mathbf{Marie}, y \mapsto y_0\})$

One of the solution to resolve the conjunctive query entailment problem: The Chase

Knowledge base $K =$

$\alpha = Mother(x) \rightarrow Parent(x)$ $\beta = Parent(x) \rightarrow \exists y. IsTheParentOf(x, y)$ $\gamma = IsTheParentOf(x, y) \rightarrow Human(y)$	$Mother(\textbf{Marie})$ $Parent(\textbf{Marie})$ $IsTheParentOf(\textbf{Marie}, y_0)$ $Human(y_0)$

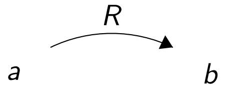
Query $Q = \{IsTheParentOf(\textbf{Marie}, x), Human(x)\}$

$$\bullet \xrightarrow{\text{IsTheParentOf}} y_0\text{: Human}$$

Marie: Mother, Parent

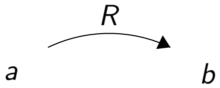
The oblivious chase

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$



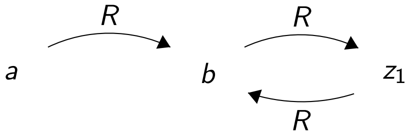
The oblivious chase

one rule: $\alpha = R(x, y) \to \exists z. R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
We apply the trigger $(\alpha, \{x \mapsto a, y \mapsto b\})$, $z_1 = f_\alpha^z(a, b)$



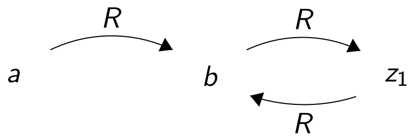
The oblivious chase

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
We apply the trigger $(\alpha, \{x \mapsto a, y \mapsto b\})$, $z_1 = f_\alpha^z(a, b)$



The oblivious chase

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
We apply the trigger $(\alpha, \{x \mapsto a, y \mapsto b\})$, $z_1 = f_\alpha^z(a, b)$
We apply the trigger $(\alpha, \{x \mapsto b, y \mapsto z_1\})$, $z_2 = f_\alpha^z(b, z_1)$



The oblivious chase

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
We apply the trigger $(\alpha, \{x \mapsto a, y \mapsto b\})$, $z_1 = f^z_\alpha(a, b)$
We apply the trigger $(\alpha, \{x \mapsto b, y \mapsto z_1\})$, $z_2 = f^z_\alpha(b, z_1)$

$$\begin{tikzcd}
a \arrow[r, bend left, "R"] & b \arrow[r, bend left, "R"] & z_1 \arrow[l, bend left, "R"] \arrow[r, bend left, "R"] & z_2 \arrow[l, bend left, "R"]
\end{tikzcd}$$

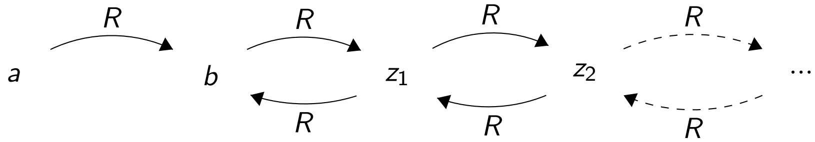
The oblivious chase

one rule: $\alpha = R(x, y) \to \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
We apply the trigger $(\alpha, \{x \mapsto a, y \mapsto b\})$, $z_1 = f^z_\alpha(a, b)$
We apply the trigger $(\alpha, \{x \mapsto b, y \mapsto z_1\})$, $z_2 = f^z_\alpha(b, z_1)$
We apply the trigger $(\alpha, \{x \mapsto z_1, y \mapsto z_2\})$

$$\begin{tikzcd} a \arrow[r, bend left, "R"] & b \arrow[r, bend left, "R"] & z_1 \arrow[l, bend left, "R"] \arrow[r, bend left, "R"] & z_2 \arrow[l, bend left, "R"] \end{tikzcd}$$

The oblivious chase

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
We apply the trigger $(\alpha, \{x \mapsto a, y \mapsto b\})$, $z_1 = f^z_\alpha(a, b)$
We apply the trigger $(\alpha, \{x \mapsto b, y \mapsto z_1\})$, $z_2 = f^z_\alpha(b, z_1)$
We apply the trigger $(\alpha, \{x \mapsto z_1, y \mapsto z_2\})$



The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

For example, if $F = \{B(x, y), R(y, z), B(x, w), R(w, z)\}$, then $Core(F) = \{B(x, y), R(y, z)\}$ because $\{w \mapsto y\}$ is a homomorphism from F to $Core(F)$.

The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
$z_1 = f^z_\alpha(a, b), z_2 = f^z_\alpha(b, z_1)$

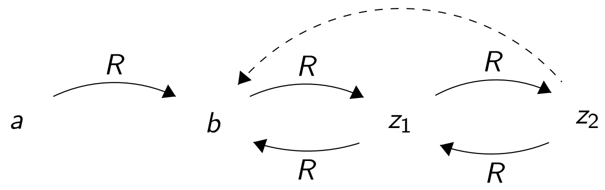
$$\begin{tikzcd} a \arrow[r, bend left, "R"] & b \arrow[r, bend left, "R"] & z_1 \arrow[l, bend left, "R"] \arrow[r, bend left, "R"] & z_2 \arrow[l, bend left, "R"] \end{tikzcd}$$

The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
$z_1 = f_\alpha^z(a, b), z_2 = f_\alpha^z(b, z_1)$

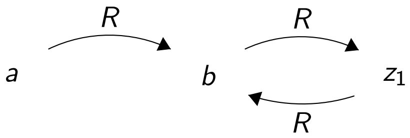


The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
$z_1 = f_\alpha^z(a, b), z_2 = f_\alpha^z(b, z_1)$

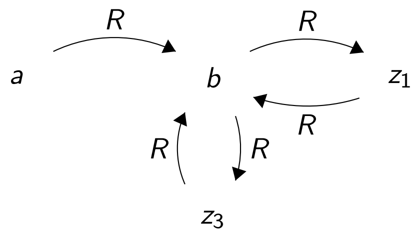


The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

one rule: $\alpha = R(x, y) \rightarrow \exists z.R(y, z) \wedge R(z, y)$
one fact: $R(a, b)$
$z_1 = f_\alpha^z(a, b), z_2 = f_\alpha^z(b, z_1)$

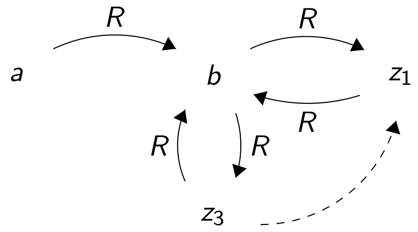


The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

one rule: $\alpha = R(x,y) \rightarrow \exists z.R(y,z) \wedge R(z,y)$
one fact: $R(a,b)$
$z_1 = f_\alpha^z(a,b), z_2 = f_\alpha^z(b,z_1)$

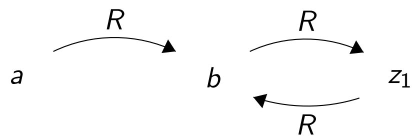


The core chase

Definition

A core of a factbase F is one of the minimal subset of F that is logically equivalent to F.

one rule: $\alpha = R(x,y) \rightarrow \exists z.R(y,z) \wedge R(z,y)$
one fact: $R(a,b)$
$z_1 = f^z_\alpha(a,b), z_2 = f^z_\alpha(b,z_1)$



Some results

Theorem

For a knowledge base K, the result of the oblivious chase is an universal model of K.

Theorem

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

Our restriction: **Horn-$\mathcal{ALCH}$**

The rules are of the form:

$$A(x) \land B(x) \to C(x)$$

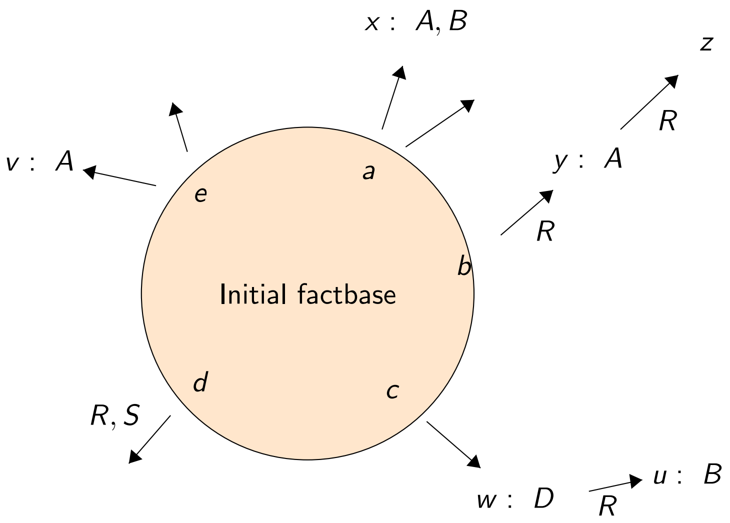
$$A(x) \land R(x,y) \to B(y)$$

$$R(x,y) \land B(y) \to A(x)$$

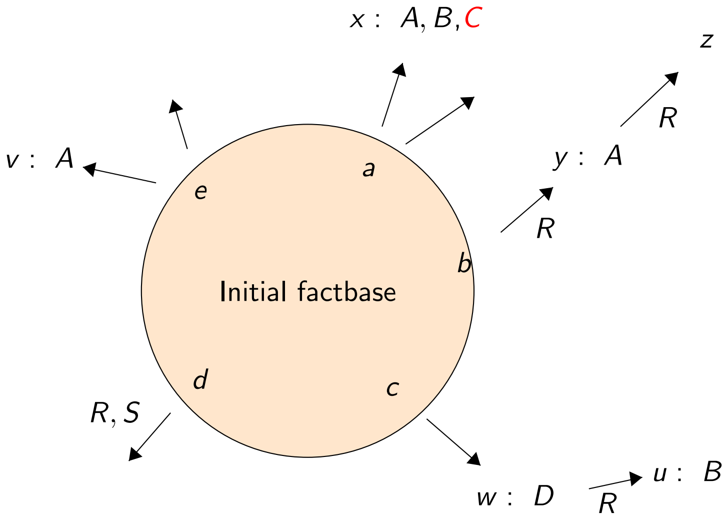
$$R(x,y) \land S(x,y) \to V(x,y)$$

$$A(x) \to \exists y.R(x,y) \land B(y)$$

Exemple

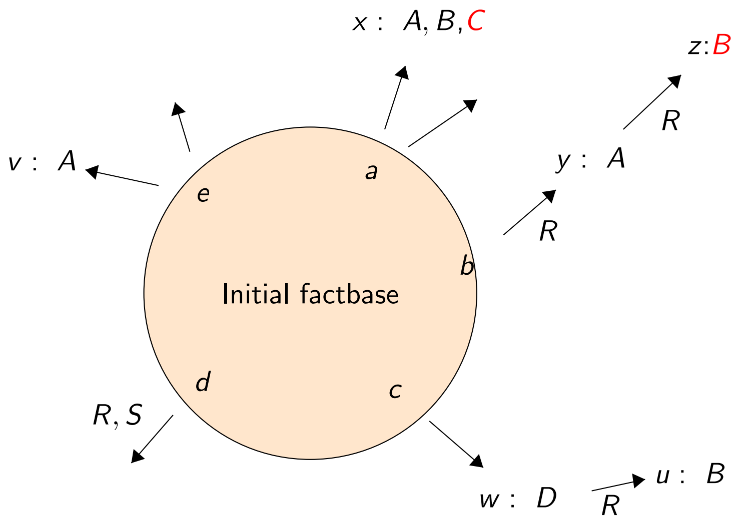


Exemple



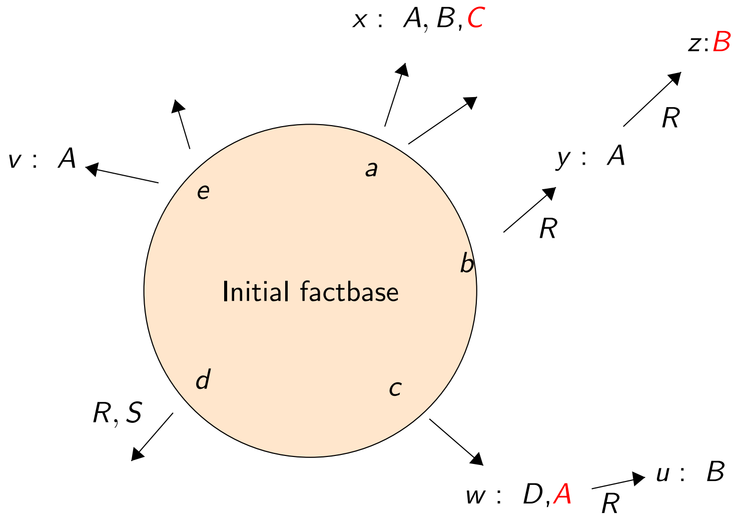
$$A(x) \wedge B(x) \rightarrow C(x)$$

Exemple



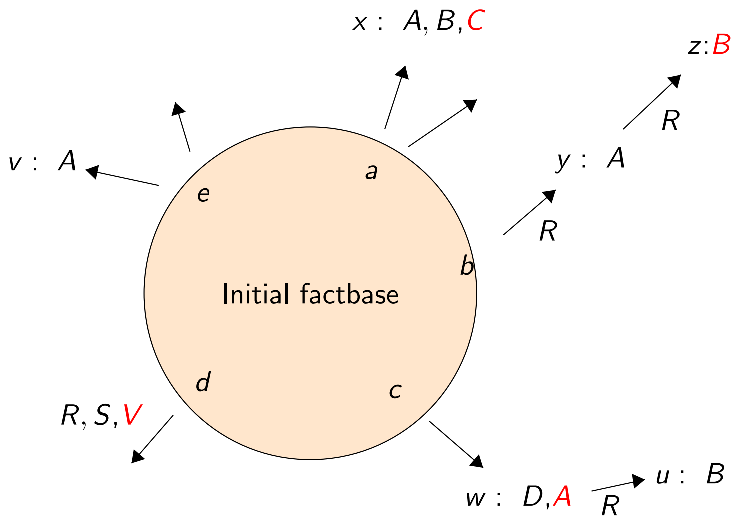
$$A(x) \wedge R(x,y) \rightarrow B(y)$$

Exemple



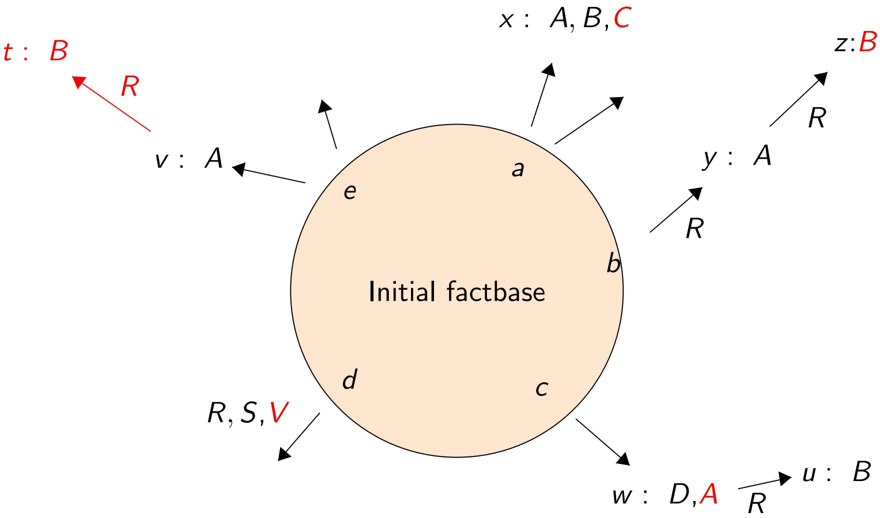
$$R(x,y) \wedge B(y) \rightarrow A(x)$$

Exemple



$$R(x,y) \wedge S(x,y) \rightarrow V(x,y)$$

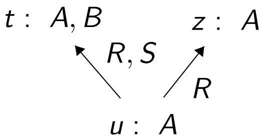
Exemple



$$A(x) \rightarrow \exists y.R(x,y) \wedge B(y)$$

A better algorithm in this restriction: The merge chase

The variable z is mergeable on the term t:



The variable z is mergeable on the term t:

$$\begin{tikzcd}
t : A, B & & z : A \\
& u : A \arrow[ul, "{R, S}"'] \arrow[ur, "R"'] &
\end{tikzcd}$$

Theorem

If the variable z is mergeable on the term t, the facts generated on z can be generated on t. (à reformuler)

A better algorithm in this restriction: The merge chase

The variable z is mergeable on the term t:

$$A(x) \wedge B(x) \rightarrow C(x)$$

$$\begin{tikzcd} t : A, B & & z : A \\ & u : A \arrow[ul, "{R, S}"] \arrow[ur, "R"'] & \end{tikzcd}$$

Theorem

If the variable z is mergeable on the term t, the facts generated on z can be generated on t. (à reformuler)

A better algorithm in this restriction: The merge chase

The variable z is mergeable on the term t:

$$A(x) \wedge R(x, y) \rightarrow B(y)$$

$$\begin{tikzcd}
t : A, B & & z : A \\
& u : A \arrow[ul, "{R, S}"] \arrow[ur, "R"'] &
\end{tikzcd}$$

Theorem

If the variable z is mergeable on the term t, the facts generated on z can be generated on t. (à reformuler)

A better algorithm in this restriction: The merge chase

The variable z is mergeable on the term t:

$$R(x, y) \land B(y) \to A(x)$$

$$\begin{tikzcd} t : A, B & & z : A \\ & u : A \arrow[ul, "{R, S}"] \arrow[ur, "R"'] & \end{tikzcd}$$

Theorem

If the variable z is mergeable on the term t, the facts generated on z can be generated on t. (à reformuler)

A better algorithm in this restriction: The merge chase

The variable z is mergeable on the term t:

$$R(x, y) \wedge S(x, y) \rightarrow V(x, y)$$

$$\begin{tikzcd}
t : A, B & & z : A \\
& u : A \arrow[ul, "{R, S}"] \arrow[ur, "R"'] &
\end{tikzcd}$$

Theorem

If the variable z is mergeable on the term t, the facts generated on z can be generated on t. (à reformuler)

A better algorithm in this restriction: The merge chase

The variable z is mergeable on the term t:

$$A(x) \rightarrow \exists y.R(x,y) \wedge B(y)$$

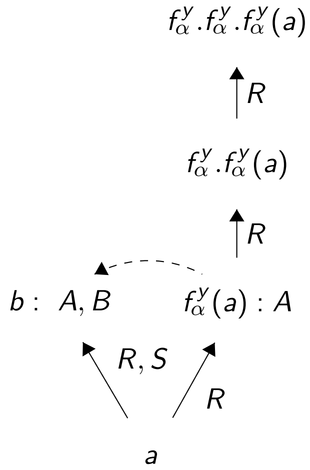
$$\begin{tikzcd}
t : A, B & & z : A \\
& u : A \arrow[ul, "{R, S}"] \arrow[ur, "R"'] &
\end{tikzcd}$$

Theorem

If the variable z is mergeable on the term t, the facts generated on z can be generated on t. (à reformuler)

A better algorithm in this restriction: The merge chase

The atomic merge



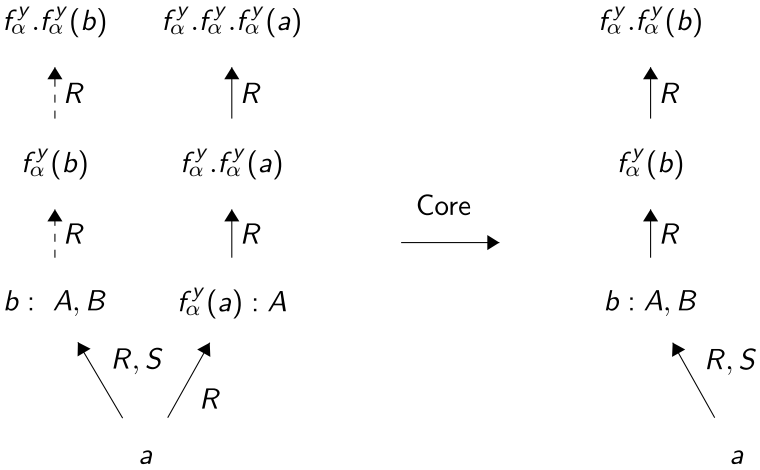
A better algorithm in this restriction: The merge chase

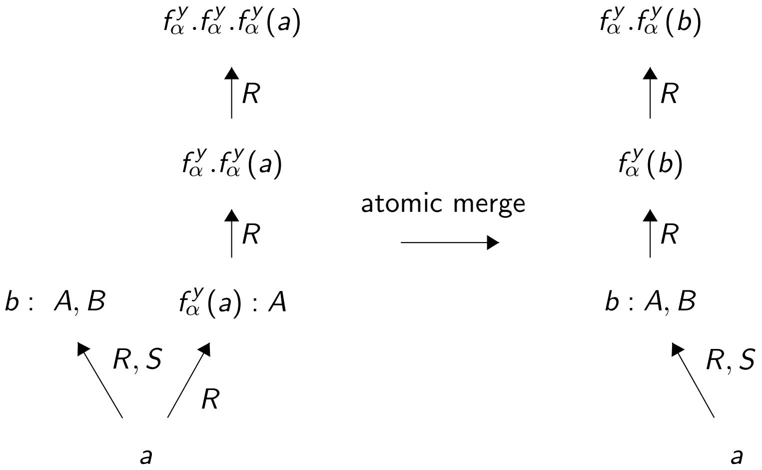
The atomic merge

$$
\begin{tikzcd}
f_\alpha^y.f_\alpha^y(b) & & f_\alpha^y.f_\alpha^y.f_\alpha^y(a) \\
f_\alpha^y(b) \arrow[u, dashed, "R"'] & & f_\alpha^y.f_\alpha^y(a) \arrow[u, "R"'] \\
b : \; A, B \arrow[u, dashed, "R"'] & & f_\alpha^y(a) : A \arrow[u, "R"'] \\
& a \arrow[ul, "{R, S}"'] \arrow[ur, "R"'] &
\end{tikzcd}
$$

A better algorithm in this restriction: The merge chase

The atomic merge

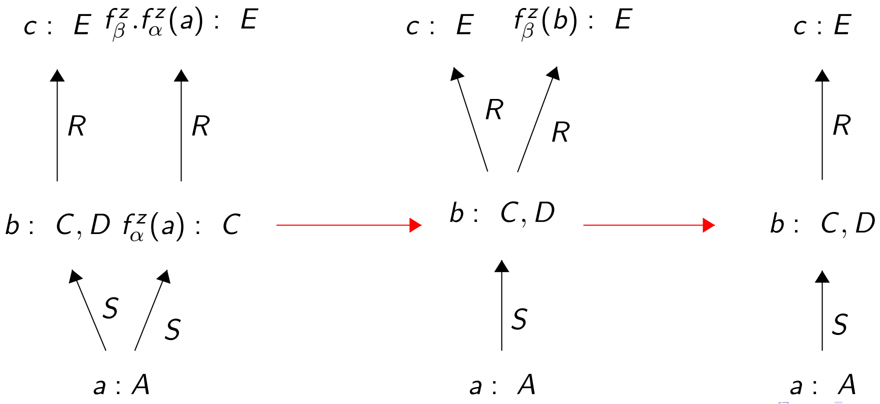




$f^y_\alpha.f^y_\alpha.f^y_\alpha(a)$
R
$f^y_\alpha.f^y_\alpha(a)$
R
$b:\ A, B$
$f^y_\alpha(a) : A$
R, S
R
a
atomic merge
$f^y_\alpha.f^y_\alpha(b)$
R
$f^y_\alpha(b)$
R
$b : A, B$
R, S
a

Definition

For a factbase F, $Merge(F)$ is obtained by applying all the possible atomic merging on F.



The merge chase

Theorem

For a factbase F, $Merge(F)$ is a core.

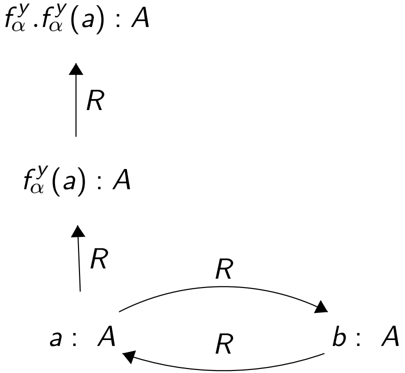
Definition

The merge chase for a **Horn-**$\mathcal{ALCH}$ knowledge base K is the core chase where we replace the computation of a core by the merge operation.

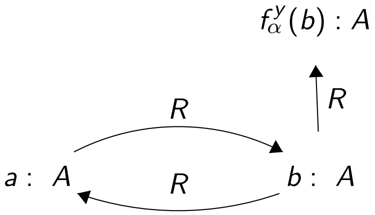
Corollary

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

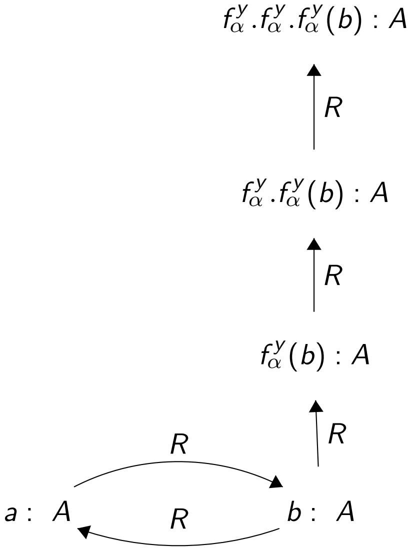
the atomic merging operation has to be applied exhaustively



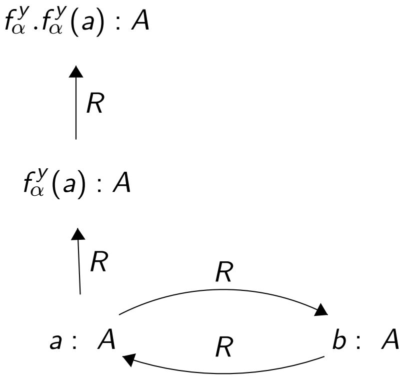
the atomic merging operation has to be applied exhaustively



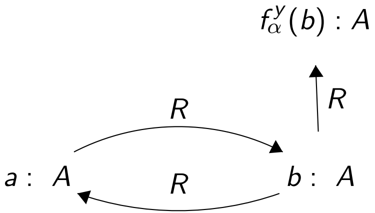
the atomic merging operation has to be applied exhaustively



the atomic merging operation has to be applied exhaustively



the atomic merging operation has to be applied exhaustively



the atomic merging operation has to be applied exhaustively

$$\begin{tikzcd} a : A \arrow[r, bend left, "R"] & b : A \arrow[l, bend left, "R"] \end{tikzcd}$$

Conclusion

- Find **Horn-**$\mathcal{ALCH}$
- define the merge chase;
- prove that the merge chase works;
- Extend our work to **Horn-**$\mathcal{ALCHI}$: $R(x, y) \wedge S(x, y) \rightarrow T(y, x)$;
- Implement the merge chase to compare.