A Non-Monotonic Logic for Distributed Access Control

Marcos Cramer [1] Diego Agustín Ambrossio [1]

[1] University of Luxembourg

LAP 2016

23 - Sept - 2016

UNIVERSITÉ DU LUXEMBOURG

Outline

UNIVERSITÉ DU LUXEMBOURG

Introduction

- Who has access to what resource?

Introduction

- Who has access to what resource?

Introduction

- Who has access to what resource?

Introduction

- Who has access to what resource?
- Many *says*-based logics.

Introduction

- Who has access to what resource?
- Many *says*-based logics.
  - "*A says* φ".

Introduction

- Who has access to what resource?
- Many *says*-based logics.
  - "*A says* φ".
  - "Principal *A supports* statement φ".

UNIVERSITÉ DU LUXEMBOURG

Introduction

- Who has access to what resource?
- Many *says*-based logics.
  - “*A says* φ”.
  - “Principal *A supports* statement φ”.
- Access is granted iff it is logically entailed by the access control policy.

UNIVERSITÉ DU LUXEMBOURG

Example

Consider the following example:

$$T_A = \left\{ \begin{array}{l} access(C,r) \wedge B \text{ says } access(C,s) \Rightarrow access(C,o) \\ access(C,r) \end{array} \right\}$$

$$T_B = \left\{ \begin{array}{l} access(C,s) \\ \neg access(C,s) \wedge A \text{ says } access(C,o) \Rightarrow access(C,o) \end{array} \right\}$$

Example

Consider the following example:

$$T_A = \left\{ \begin{array}{l} access(C,r) \wedge B \textit{ says } access(C,s) \Rightarrow access(C,o) \\ access(C,r) \end{array} \right\}$$

$$T_B = \left\{ \begin{array}{l} access(C,s) \\ \neg access(C,s) \wedge A \textit{ says } access(C,o) \Rightarrow access(C,o) \end{array} \right\}$$

- The *says*-statement is irrelevant.

Example

Consider the following example:

$$T_A = \left\{ \begin{array}{l} access(C,r) \wedge B \text{ } says \text{ } access(C,s) \Rightarrow access(C,o) \\ access(C,r) \end{array} \right\}$$

$$T_B = \left\{ \begin{array}{l} access(C,s) \\ \neg access(C,s) \wedge A \text{ } says \text{ } access(C,o) \Rightarrow access(C,o) \end{array} \right\}$$

- The *says*-statement is irrelevant.
- Communication Overload!

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- Monotonicity

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- Monotonicity
  - New statements cannot lead to *less* access.

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- Monotonicity
  - New statements cannot lead to *less* access.
- Non-Monotonic!

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- Monotonicity
  - New statements cannot lead to *less* access.
- Non-Monotonic!
  - Modeling Denial.

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- Monotonicity
  - New statements cannot lead to *less* access.
- Non-Monotonic!
  - Modeling Denial.
  - $\neg B\ says \neg\ access(C, r) \rightarrow\ access(C, r)$

Introduction - Cont.

- The statements issued by a principal completely characterize what a principal supports.

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- The statements issued by a principal completely characterize what a principal supports.
- Similar to the motivation for autoepistemic logic:

Introduction - Cont.

- The statements issued by a principal completely characterize what a principal supports.
- Similar to the motivation for autoepistemic logic:

> "An agent's knowledge base completely characterizes what the agent knows"

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- The statements issued by a principal completely characterize what a principal supports.
- Similar to the motivation for autoepistemic logic:

> "An agent's knowledge base completely characterizes what the agent knows"

- We use autoepistemic logic with well-founded semantics

Introduction - Cont.

- We adapt autoepistemic logic to the multi-agent case.

Introduction - Cont.

- We adapt autoepistemic logic to the multi-agent case.
- Need to specify how the agents' "knowledge" interacts.

Introduction - Cont.

- We adapt autoepistemic logic to the multi-agent case.
- Need to specify how the agents' "knowledge" interacts.
- Standard *says*-based logic:

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- We adapt autoepistemic logic to the multi-agent case.
- Need to specify how the agents' "knowledge" interacts.
- Standard *says*-based logic:
  - Mutual positive introspection:

$$k \textit{ says } \varphi \Rightarrow j \textit{ says } k \textit{ says } \varphi$$

Introduction - Cont.

- We adapt autoepistemic logic to the multi-agent case.
- Need to specify how the agents' "knowledge" interacts.
- Standard *says*-based logic:
  - Mutual positive introspection:

$$k \textit{ says } \varphi \Rightarrow j \textit{ says } k \textit{ says } \varphi$$

- For denial, we need:

UNIVERSITÉ DU LUXEMBOURG

Introduction - Cont.

- We adapt autoepistemic logic to the multi-agent case.
- Need to specify how the agents' "knowledge" interacts.
- Standard *says*-based logic:
  - Mutual positive introspection:

$$k \textit{ says } \varphi \Rightarrow j \textit{ says } k \textit{ says } \varphi$$

- For denial, we need:
  - Mutual negative introspection:

$$\neg k \textit{ says } \varphi \Rightarrow j \textit{ says } \neg k \textit{ says } \varphi$$

UNIVERSITÉ DU LUXEMBOURG

Outline

Syntax

D-ACL Syntax

t denotes an arbitrary term and x and arbitrary variable:

$$\varphi ::= P(t,\ldots,t) \mid t = t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid t \textit{ says } \varphi$$

Syntax

D-ACL Syntax

t denotes an arbitrary term and x and arbitrary variable:

$$\varphi ::= P(t,\dots,t) \mid t = t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\ \varphi \mid t \textit{ says } \varphi$$

Inductive Definition

An D-ACL *inductive definition* Δ is a finite set of rules of the form $P(t_1,\dots,t_n) \leftarrow \varphi$, where P is an n-ary predicate symbol and φ is a D-ACL formula.

Syntax

D-ACL Syntax

t denotes an arbitrary term and x and arbitrary variable:

$$\varphi ::= P(t, \ldots, t) \mid t = t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid t \textit{ says } \varphi$$

Inductive Definition

An D-ACL *inductive definition* Δ is a finite set of rules of the form $P(t_1, \ldots, t_n) \leftarrow \varphi$, where P is an n-ary predicate symbol and φ is a D-ACL formula.

D-ACL Theory

A D-ACL *theory* is a set that consists of D-ACL formulas and D-ACL inductive definitions.

Example

$$T_A = \left\{ \begin{array}{l} \{p \leftarrow B \text{ says } p \\ p \leftarrow r\} \\ p \wedge s \wedge B \text{ says } q \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ B \text{ says } r \vee \neg(B \text{ says } r) \Rightarrow q \end{array} \right\}$$

$$T_B = \left\{ \begin{array}{l} p \\ C \text{ says } q \Rightarrow q \\ C \text{ says } r \Rightarrow r \end{array} \right\} \qquad T_C = \left\{ \begin{array}{l} \neg(B \text{ says } q) \Rightarrow q \\ B \text{ says } r \Rightarrow r \end{array} \right\}$$

Outline

UNIVERSITÉ DU LUXEMBOURG

Decision procedure

- We define a decision procedure for D-ACL.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.
- Implemented in IDP.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.
- Implemented in IDP.
- Well-founded semantics uses three truth-values: **t**, **f** and **u**.

UNIVERSITÉ DU LUXEMBOURG

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.
- Implemented in IDP.
- Well-founded semantics uses three truth-values: **t**, **f** and **u**.
- Three-valuedness arises only through the modal operator *says*.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.
- Implemented in IDP.
- Well-founded semantics uses three truth-values: **t**, **f** and **u**.
- Three-valuedness arises only through the modal operator *says*.
- We use $p^{+}_{A\_\mathtt{says}\_\varphi}$ for the upper bound for the truth value of *A says* φ and $p^{-}_{A\_\mathtt{says}\_\varphi}$ for the lower bound.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.
- Implemented in IDP.
- Well-founded semantics uses three truth-values: **t**, **f** and **u**.
- Three-valuedness arises only through the modal operator *says*.
- We use $p^{+}_{A\_\mathtt{says}\_\varphi}$ for the upper bound for the truth value of A *says* φ and $p^{-}_{A\_\mathtt{says}\_\varphi}$ for the lower bound.
- $p^{+}_{A\_\mathtt{says}\_\varphi}$ is used in positive contexts and $p^{-}_{A\_\mathtt{says}\_\varphi}$ in negative contexts.

Decision procedure

- We define a decision procedure for D-ACL.
  - It coincides with the well-founded semantics.
  - It minimizes communication.
- Implemented in IDP.
- Well-founded semantics uses three truth-values: **t**, **f** and **u**.
- Three-valuedness arises only through the modal operator *says*.
- We use $p^{+}_{A\_\mathtt{says}\_\varphi}$ for the upper bound for the truth value of *A says* φ and $p^{-}_{A\_\mathtt{says}\_\varphi}$ for the lower bound.
- $p^{+}_{A\_\mathtt{says}\_\varphi}$ is used in positive contexts and $p^{-}_{A\_\mathtt{says}\_\varphi}$ in negative contexts.
- In inductive definitions, subformulas cannot be meaningfully termed positive or negative.

UNIVERSITÉ DU LUXEMBOURG

Translation

$t(T)$

For every modal atom A *says* φ occurring in the body of an inductive definition in theory T,

- replace A *says* φ by the propositional variable $w_{A\_\mathtt{says}\_\varphi}$
- add to $t(T)$ the two formulae $w_{A\_\mathtt{says}\_\varphi} \Rightarrow A$ *says* φ and A *says* $\varphi \Rightarrow w_{A\_\mathtt{says}\_\varphi}$.

Translation - Cont.

$\tau(T)$

Let T be a D-ACL theory. $\tau(T)$ is constructed from $t(T)$ by performing the following replacements for every *says*-atom A *says* φ occurring in $t(T)$ that is not the sub-formula of another *says*-atom:

- Replace every positive occurrence of A *says* φ in T by $p^{+}_{A\_\mathtt{says}\_\varphi}$.
- Replace every negative occurrence of A *says* φ in T by $p^{-}_{A\_\mathtt{says}\_\varphi}$.

UNIVERSITÉ DU LUXEMBOURG

Example - Translation

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_\mathtt{says}\_p} \\ p \leftarrow r\} \\ w_{B\_\mathtt{says}\_p} \Rightarrow p^{+}_{B\_\mathtt{says}\_p} \\ p^{-}_{B\_\mathtt{says}\_p} \Rightarrow w_{B\_\mathtt{says}\_p} \\ p \wedge s \wedge p^{-}_{B\_\mathtt{says}\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_\mathtt{says}\_r} \vee \neg p^{+}_{B\_\mathtt{says}\_r} \Rightarrow q \end{array} \right\}$$

$$\mathcal{T}_B = \left\{ \begin{array}{l} p \\ p^{-}_{C\_\mathtt{says}\_q} \Rightarrow q \\ p^{-}_{C\_\mathtt{says}\_r} \Rightarrow r \end{array} \right\} \qquad \mathcal{T}_C = \left\{ \begin{array}{l} \neg p^{+}_{B\_\mathtt{says}\_q} \Rightarrow q \\ p^{-}_{B\_\mathtt{says}\_r} \Rightarrow r \end{array} \right\}$$

UNIVERSITÉ DU LUXEMBOURG

Theories and Structures

- We work with partial structures: They are like standard first-order structures, but with missing information.

Partial Model

We say S is a *partial model* for $\mathcal{T}$ if and only if there exists a total structure $S' \supseteq S$ such that $S' \models \mathcal{T}$.

Minimal Inconsistent Set

Let $\mathcal{T}$ a theory such that $S \not\models \mathcal{T}$. We define *min_incons_set*$(\mathcal{T}, S)$ as the set of minimal (under set inclusion) partial structure $S' \subseteq S$ such that the theory $\mathcal{T}$ has no models that expand S.

Theories and Structures

Set $\mathbb{S}$

We define $\mathbb{S}$ to be the set containing every partial structure S such that:

- For every symbol $\sigma \in \Sigma'$, if $\sigma \neq p^{+}_{A\_\mathtt{says}\_\varphi}$ or $\sigma \neq p^{-}_{A\_\mathtt{says}\_\varphi}$, then $(\sigma)^{I} = \mathbf{u}$
- For every *says*-atom *A says* φ occurring in T:
  - $(p^{+}_{A\_\mathtt{says}\_\varphi})^{I} \neq \mathbf{t}$.
  - $(p^{-}_{A\_\mathtt{says}\_\varphi})^{I} \neq \mathbf{f}$.
- For no *says*-atom *A says* ϕ, $(p^{+}_{A\_\mathtt{says}\_\varphi})^{I} = \mathbf{f}$ and $(p^{-}_{A\_\mathtt{says}\_\varphi})^{I} = \mathbf{t}$.

UNIVERSITÉ DU LUXEMBOURG

Outline

UNIVERSITÉ DU LUXEMBOURG

Query Minimization Procedure.

Input: theory $\mathcal{T}$, D-ACL query α
Output: set $\mathbb{L}$ of sets of modal atoms

1: $\mathbb{L} := \emptyset$
2: $\mathcal{T} := \tau(T \cup \{\{\neg\alpha\}\})$
3: **for each** $S \in \mathbb{S}$ **do**
4: **if** S is **not** a partial model of $\mathcal{T}$ **then**
5: pick a partial structure S_{min} from min_incons_set($\mathcal{T}$, S)
6: $\mathbb{L} := \mathbb{L} \cup \{L^{S_{min}}\}$
7: **return** $\mathbb{L}$

UNIVERSITÉ DU LUXEMBOURG

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_\mathtt{says}\_p} \\ p \leftarrow r\} \\ w_{B\_\mathtt{says}\_p} \Rightarrow p^{+}_{B\_\mathtt{says}\_p} \\ p^{-}_{B\_\mathtt{says}\_p} \Rightarrow w_{B\_\mathtt{says}\_p} \\ p \wedge s \wedge p^{-}_{B\_\mathtt{says}\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_\mathtt{says}\_r} \vee \neg p^{+}_{B\_\mathtt{says}\_r} \Rightarrow q \end{array} \right\}$$

Example - Query Procedure

Query: "q"

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_\mathtt{says}\_p} \\ p \leftarrow r\} \\ w_{B\_\mathtt{says}\_p} \Rightarrow p^{+}_{B\_\mathtt{says}\_p} \\ p^{-}_{B\_\mathtt{says}\_p} \Rightarrow w_{B\_\mathtt{says}\_p} \\ p \wedge s \wedge p^{-}_{B\_\mathtt{says}\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_\mathtt{says}\_r} \vee \neg p^{+}_{B\_\mathtt{says}\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_\mathtt{says}\_p}$ and $p^{-}_{B\_\mathtt{says}\_q}$: $\{B\ says\ p; B\ says\ q\}$

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_\mathtt{says}\_p} \\ p \leftarrow r\} \\ w_{B\_\mathtt{says}\_p} \Rightarrow p^{+}_{B\_\mathtt{says}\_p} \\ p^{-}_{B\_\mathtt{says}\_p} \Rightarrow w_{B\_\mathtt{says}\_p} \\ p \wedge s \wedge p^{-}_{B\_\mathtt{says}\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_\mathtt{says}\_r} \vee \neg p^{+}_{B\_\mathtt{says}\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_\mathtt{says}\_p}$ and $p^{-}_{B\_\mathtt{says}\_q}$: $\{B\ says\ p; B\ says\ q\}$

Example - Query Procedure

Query: "q"

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_says\_p} \\ p \leftarrow r\} \\ w_{B\_says\_p} \Rightarrow p^{+}_{B\_says\_p} \\ p^{-}_{B\_says\_p} \Rightarrow w_{B\_says\_p} \\ p \wedge s \wedge p^{-}_{B\_says\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_says\_r} \vee \neg p^{+}_{B\_says\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_says\_p}$ and $p^{-}_{B\_says\_q}$: $\{B\ says\ p; B\ says\ q\}$

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_says\_p} \\ p \leftarrow r\} \\ w_{B\_says\_p} \Rightarrow p^{+}_{B\_says\_p} \\ p^{-}_{B\_says\_p} \Rightarrow w_{B\_says\_p} \\ p \wedge s \wedge p^{-}_{B\_says\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_says\_r} \vee \neg p^{+}_{B\_says\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_says\_p}$ and $p^{-}_{B\_says\_q}$: $\{B\ says\ p; B\ says\ q\}$
- $p^{-}_{B\_says\_r}$: $\{B\ says\ r\}$

UNIVERSITÉ DU LUXEMBOURG

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_says\_p} \\ p \leftarrow r\} \\ w_{B\_says\_p} \Rightarrow p^{+}_{B\_says\_p} \\ p^{-}_{B\_says\_p} \Rightarrow w_{B\_says\_p} \\ p \wedge s \wedge p^{-}_{B\_says\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_says\_r} \vee \neg p^{+}_{B\_says\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_says\_p}$ and $p^{-}_{B\_says\_q}$: $\{B\ says\ p; B\ says\ q\}$
- $p^{-}_{B\_says\_r}$: $\{B\ says\ r\}$

UNIVERSITÉ DU LUXEMBOURG

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_says\_p} \\ p \leftarrow r\} \\ w_{B\_says\_p} \Rightarrow p^{+}_{B\_says\_p} \\ p^{-}_{B\_says\_p} \Rightarrow w_{B\_says\_p} \\ p \wedge s \wedge p^{-}_{B\_says\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_says\_r} \vee \neg p^{+}_{B\_says\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_says\_p}$ and $p^{-}_{B\_says\_q}$: $\{B\ says\ p; B\ says\ q\}$
- $p^{-}_{B\_says\_r}$: $\{B\ says\ r\}$
- $\neg p^{+}_{B\_says\_r}$: $\{\neg(B\ says\ r)\}$

UNIVERSITÉ DU LUXEMBOURG

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_\mathtt{says}\_p} \\ p \leftarrow r\} \\ w_{B\_\mathtt{says}\_p} \Rightarrow p^{+}_{B\_\mathtt{says}\_p} \\ p^{-}_{B\_\mathtt{says}\_p} \Rightarrow w_{B\_\mathtt{says}\_p} \\ p \wedge s \wedge p^{-}_{B\_\mathtt{says}\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_\mathtt{says}\_r} \vee \neg p^{+}_{B\_\mathtt{says}\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_\mathtt{says}\_p}$ and $p^{-}_{B\_\mathtt{says}\_q}$: $\{B\ says\ p; B\ says\ q\}$
- $p^{-}_{B\_\mathtt{says}\_r}$: $\{B\ says\ r\}$
- $\neg p^{+}_{B\_\mathtt{says}\_r}$: $\{\neg(B\ says\ r)\}$

UNIVERSITÉ DU LUXEMBOURG

Example - Query Procedure

Query: “q“

$$\mathcal{T}_A = \left\{ \begin{array}{l} \{p \leftarrow w_{B\_\mathtt{says}\_p} \\ p \leftarrow r\} \\ w_{B\_\mathtt{says}\_p} \Rightarrow p^{+}_{B\_\mathtt{says}\_p} \\ p^{-}_{B\_\mathtt{says}\_p} \Rightarrow w_{B\_\mathtt{says}\_p} \\ p \wedge s \wedge p^{-}_{B\_\mathtt{says}\_q} \Rightarrow q \\ r \vee \neg r \Rightarrow s \\ p^{-}_{B\_\mathtt{says}\_r} \vee \neg p^{+}_{B\_\mathtt{says}\_r} \Rightarrow q \end{array} \right\}$$

- $p^{-}_{B\_\mathtt{says}\_p}$ and $p^{-}_{B\_\mathtt{says}\_q}$: $\{B\ says\ p; B\ says\ q\}$
- $p^{-}_{B\_\mathtt{says}\_r}$: $\{B\ says\ r\}$
- $\neg p^{+}_{B\_\mathtt{says}\_r}$: $\{\neg(B\ says\ r)\}$

$$\mathbb{L} = \{\{B\ says\ p; B\ says\ q\}; \{B\ says\ r\}; \{\neg(B\ says\ r)\}\}$$

UNIVERSITÉ DU LUXEMBOURG

Outline

Communication Procedure

(1) Apply Query Minimization Procedure.

Communication Procedure

(1) Apply Query Minimization Procedure.

(2) Build *query graph*:

Communication Procedure

(1) Apply Query Minimization Procedure.

(2) Build *query graph*:

- query vertices: $\langle A : \alpha \rangle : \{? \mid \mathbf{t} \mid \mathbf{f} \mid \mathbf{u}\}$.

UNIVERSITÉ DU LUXEMBOURG

Communication Procedure

(1) Apply Query Minimization Procedure.

(2) Build *query graph*:

- query vertices: $\langle A : \alpha \rangle : \{? \mid \mathbf{t} \mid \mathbf{f} \mid \mathbf{u}\}$.
- *says* vertices: $\{A \text{ says } \varphi\}$; $\{\neg A \text{ says } \varphi\}$; ....

Communication Procedure

(1) Apply Query Minimization Procedure.

(2) Build *query graph*:

- query vertices: $\langle A : \alpha \rangle : \{? \mid \mathbf{t} \mid \mathbf{f} \mid \mathbf{u}\}$.
- *says* vertices: $\{A \textit{ says } \varphi\}; \{\neg A \textit{ says } \varphi\}; \ldots$.
- unlabelled edges: from query vertices to *says* vertices (that make the query true).

Communication Procedure

(1) Apply Query Minimization Procedure.

(2) Build *query graph*:

- query vertices: $\langle A : \alpha \rangle : \{? \mid \mathbf{t} \mid \mathbf{f} \mid \mathbf{u}\}$.
- *says* vertices: $\{A \textit{ says } \varphi\}; \{\neg A \textit{ says } \varphi\}; \ldots$.
- unlabelled edges: from query vertices to *says* vertices (that make the query true).
- labelled edges: from *says* vertices to query vertices.

Example Query Graph

We query principal A about the truth value of q.

Example Query Graph

We query principal A about the truth value of q.

$$minimize\_query(A, q)$$

Example Query Graph

We query principal *A* about the truth value of *q*.

$$minimize\_query(A, q)$$

$$\mathbb{L} = \{\{B\ says\ p; B\ says\ q\}; \{B\ says\ r\}; \{\neg(B\ says\ r)\}\}$$

UNIVERSITÉ DU LUXEMBOURG

Example Query Graph

We query principal A about the truth value of q.

$$minimize\_query(A, q)$$

$$\mathbb{L} = \{\{B\ says\ p; B\ says\ q\}; \{B\ says\ r\}; \{\neg(B\ says\ r)\}\}$$

We start building the query graph:

UNIVERSITÉ DU LUXEMBOURG

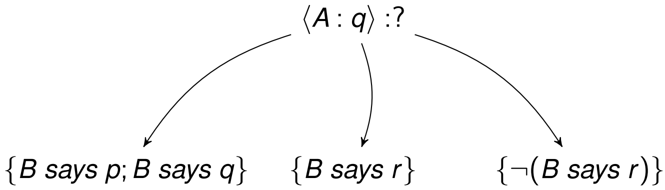
Example Query Graph

We query principal *A* about the truth value of *q*.

$$minimize\_query(A, q)$$

$$\mathbb{L} = \{\{B \text{ says } p; B \text{ says } q\}; \{B \text{ says } r\}; \{\neg(B \text{ says } r)\}\}$$

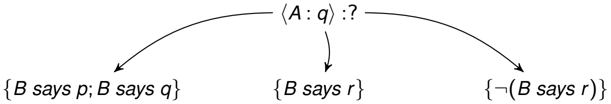
We start building the query graph:



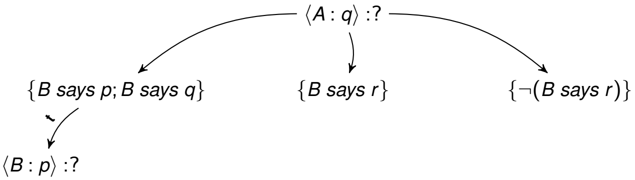
Example Query Graph - Complete

$\langle A : q \rangle$:?

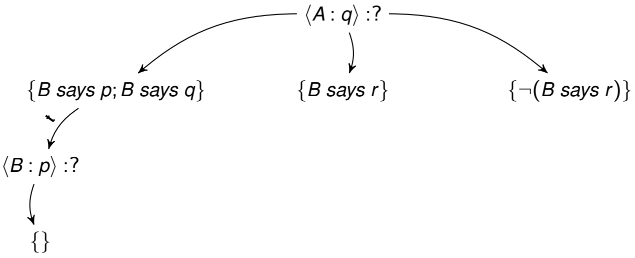
Example Query Graph - Complete



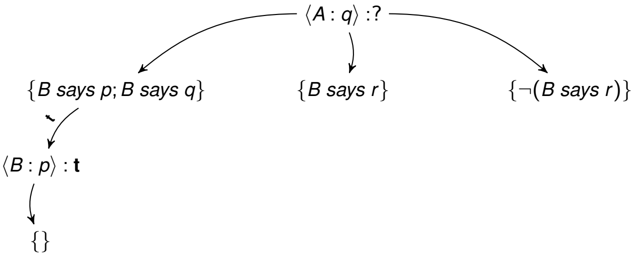
Example Query Graph - Complete



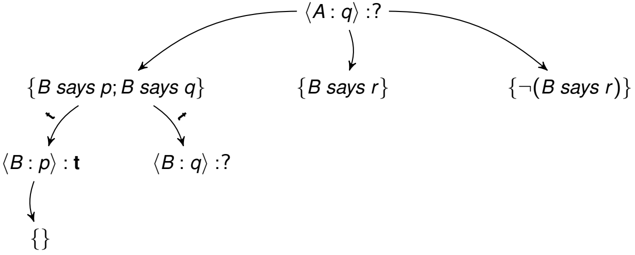
Example Query Graph - Complete



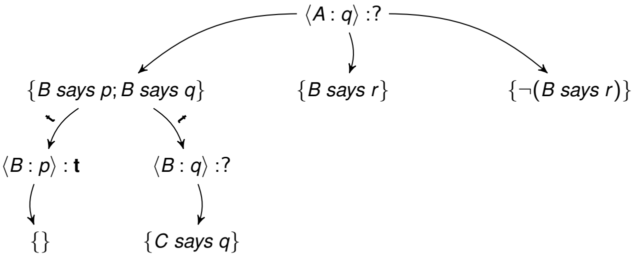
Example Query Graph - Complete



Example Query Graph - Complete

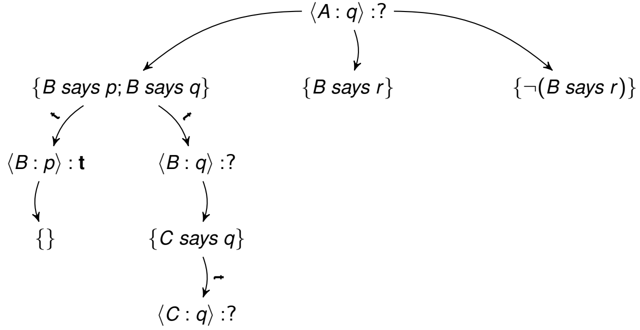


Example Query Graph - Complete

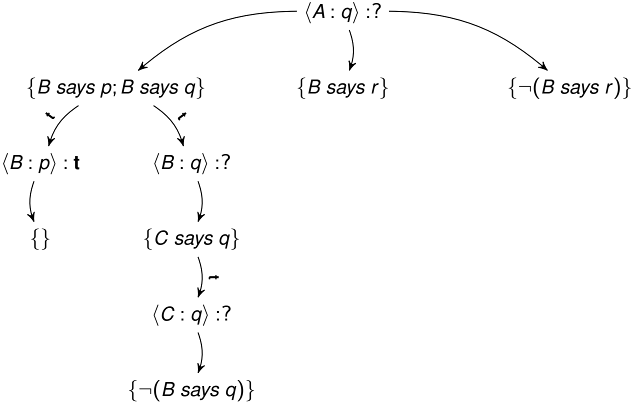


UNIVERSITÉ DU LUXEMBOURG

Example Query Graph - Complete

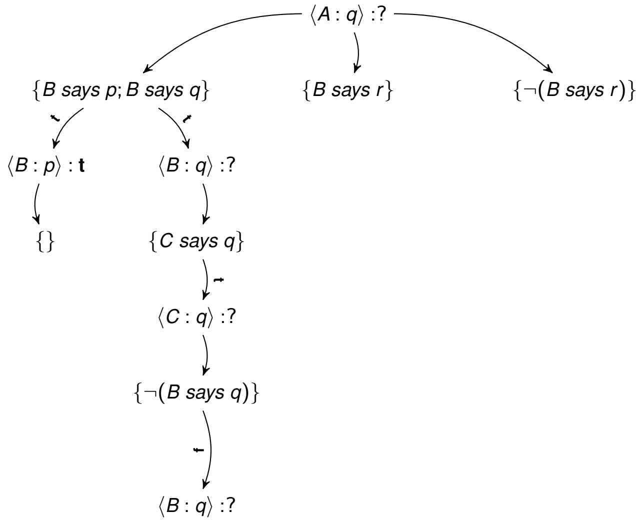


Example Query Graph - Complete

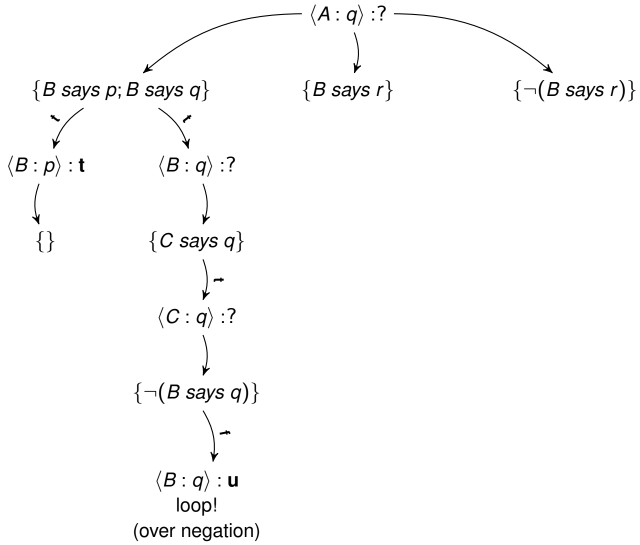


UNIVERSITÉ DU LUXEMBOURG

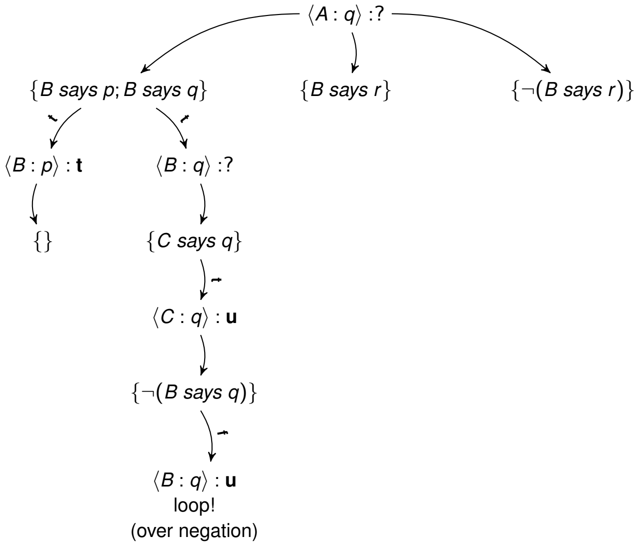
Example Query Graph - Complete



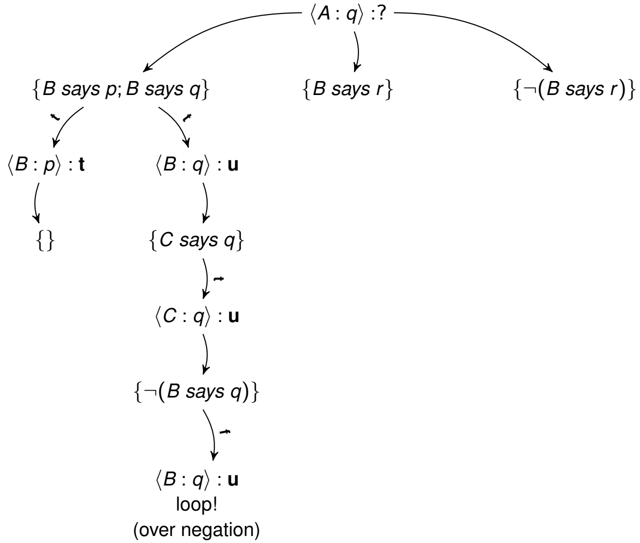
Example Query Graph - Complete



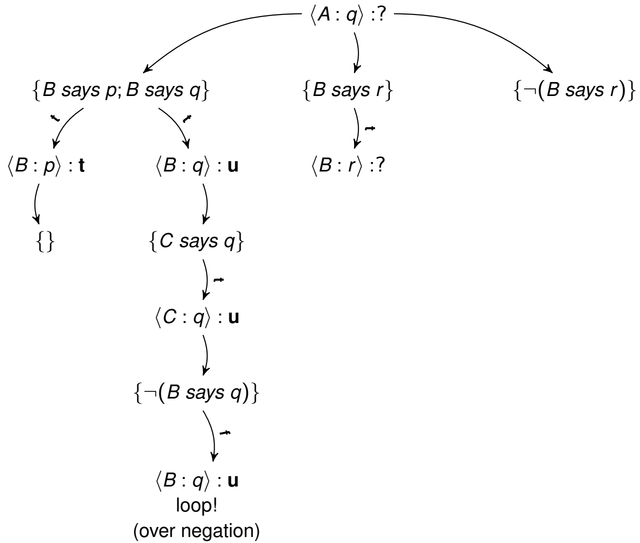
Example Query Graph - Complete



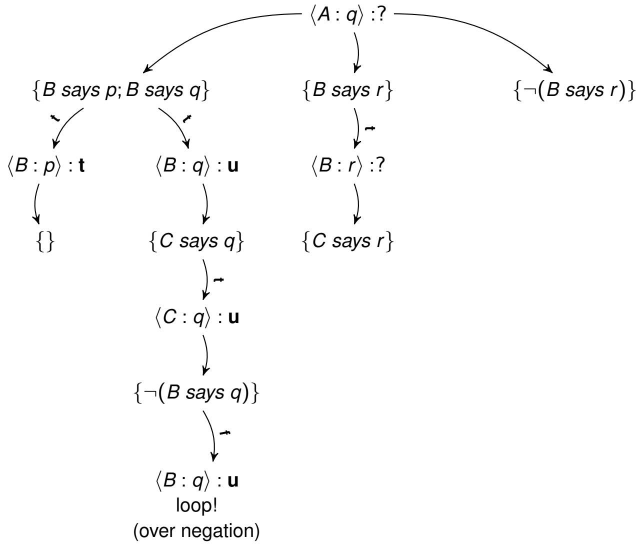
Example Query Graph - Complete



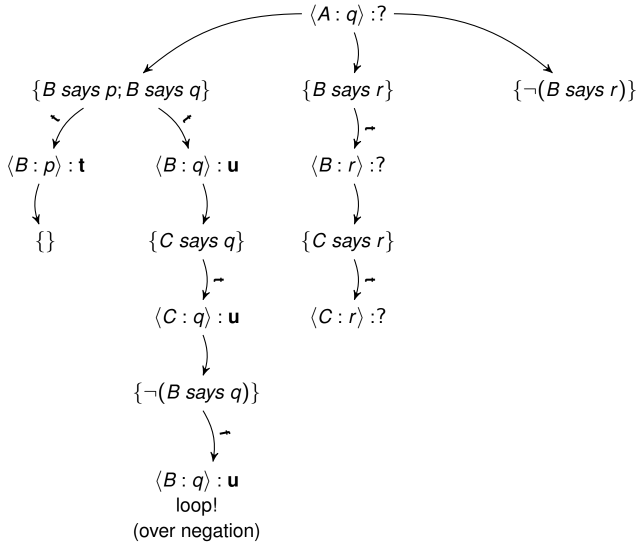
Example Query Graph - Complete



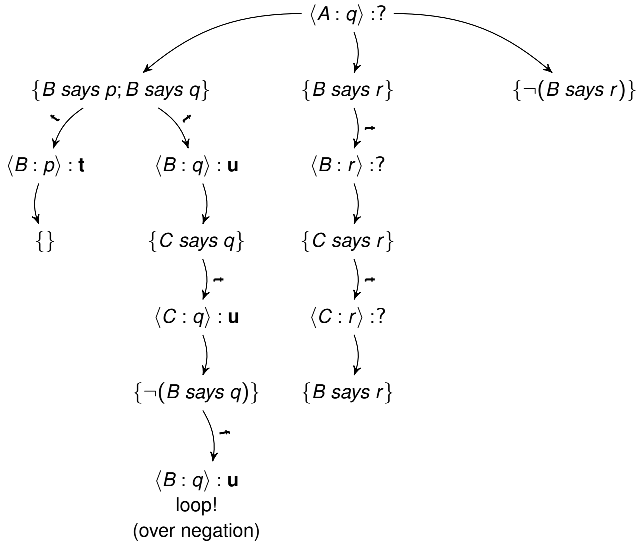
Example Query Graph - Complete



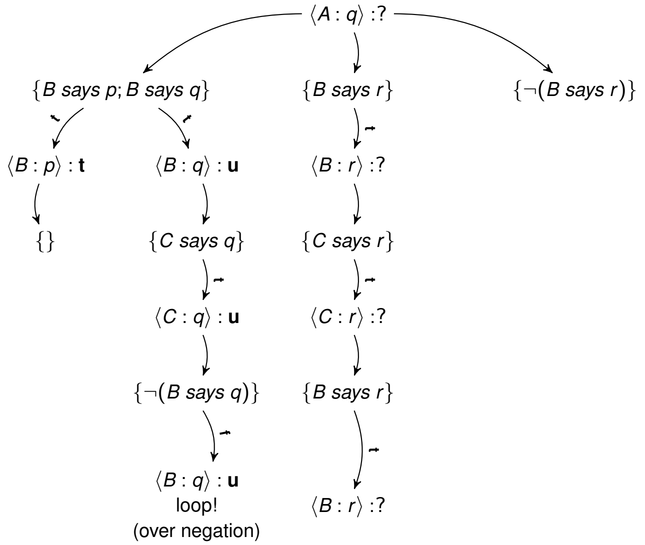
Example Query Graph - Complete



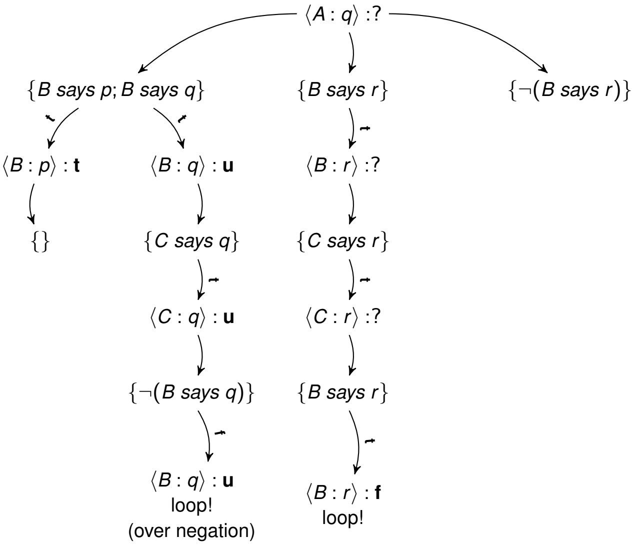
Example Query Graph - Complete



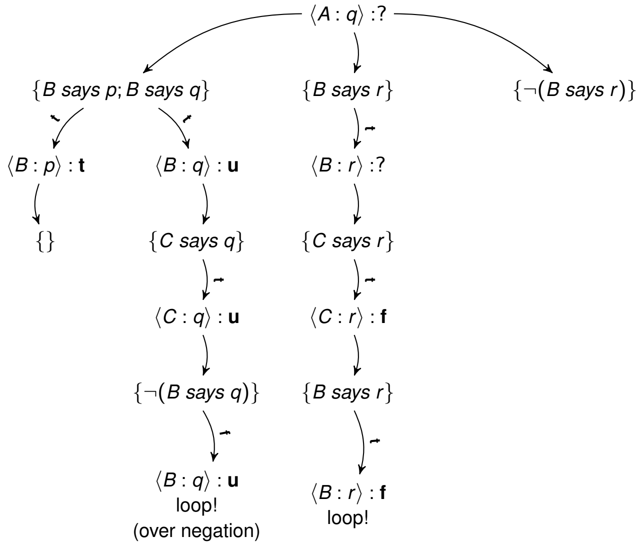
Example Query Graph - Complete



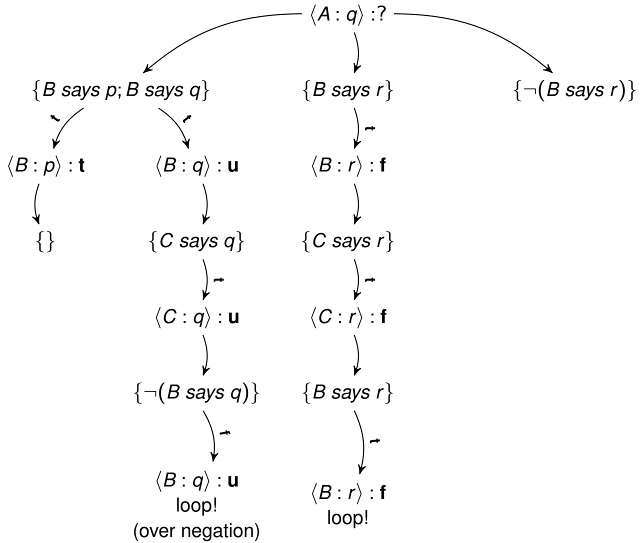
Example Query Graph - Complete



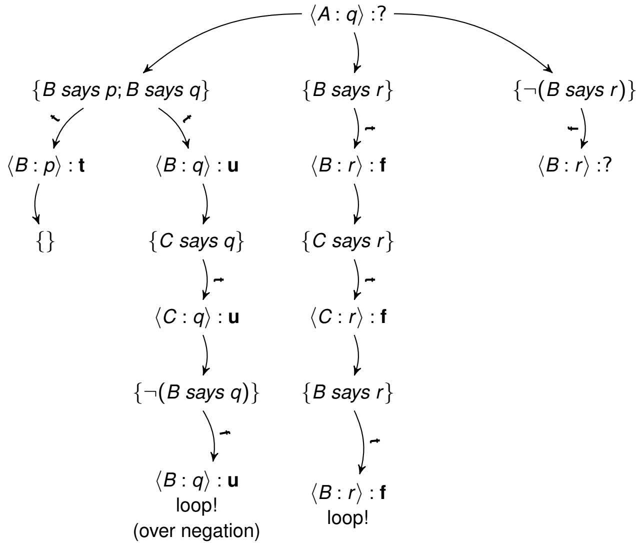
Example Query Graph - Complete



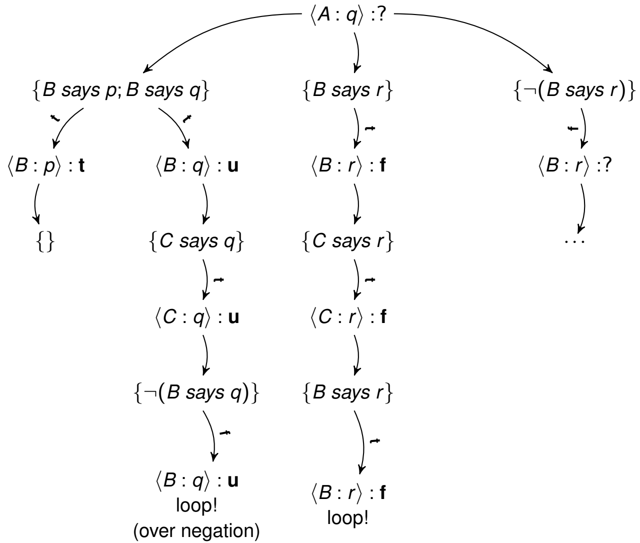
Example Query Graph - Complete



Example Query Graph - Complete

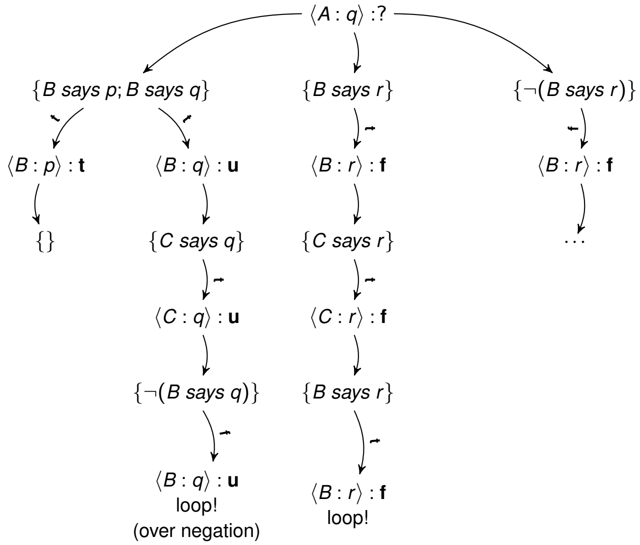


Example Query Graph - Complete

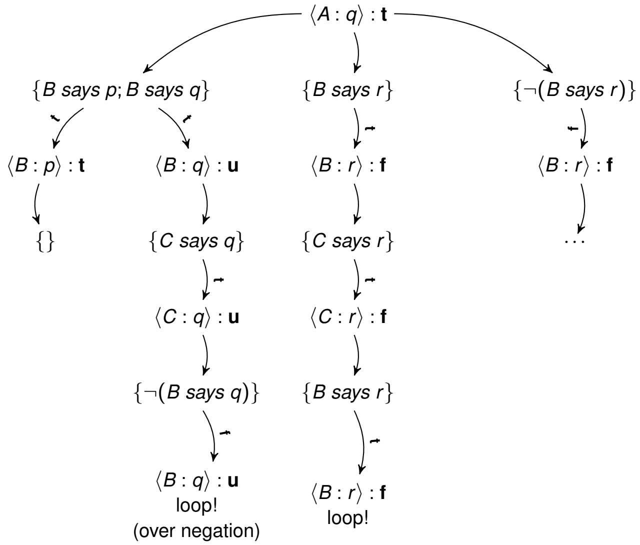


UNIVERSITÉ DU LUXEMBOURG

Example Query Graph - Complete



Example Query Graph - Complete



Last Slide

Thanks!

Questions?