can you answer while you wait?

luís cruz-filipe[1]
(joint work with graça gaspar[2] & isabel nunes[2])

[1]department of mathematics and computer science
university of southern denmark

[2]department of informatics
faculty of sciences, university of lisbon

acp seminar
june 17th, 2022

Outline

the context

continuous queries over data streams

- modern-day distributed systems
- information pouring in from e.g. sensors
- queries need to be answered in real-time
- answers are output as information arrives

the context

continuous queries over data streams

- modern-day distributed systems
- information pouring in from e.g. sensors
- queries need to be answered in real-time
- answers are output as information arrives

several models

common approach: rule-based reasoning

- usually based on variants of datalog
- set of facts dynamically obtained from a data stream D
- common problems: blocking queries, unbound wait

initial contribution (aaai'20)

online algorithm with offline pre-processing outputting partial information

- information that an answer may be output in the future
- fundamentation for such hypothetical answers

initial contribution (aaai'20)

online algorithm with offline pre-processing outputting partial information

- information that an answer may be output in the future
- fundamentation for such hypothetical answers

practical relevance

partial information allows for preventive measures to be taken

- an action might be required $\rightsquigarrow$ maybe prepare for it
- a failure might occur $\rightsquigarrow$ steps may be taken to prevent it

the justification for *why* the hypothetical answer is output can be used to evaluate its likelihood

detecting malfunctions in wind turbines

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

- a data center managing a set of wind turbines receives temperature readings $\mathsf{Temp}(Device, Level, Time)$ from sensors in each turbine
- the data centre tracks activation of cooling measures in each turbine, recording shutdowns by means of a program in temporal datalog

detecting malfunctions in wind turbines

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \land \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \land \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

query: $Q = \mathsf{Shdn}(X, T)$

if:

$$\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, i) \qquad i = 0, 1, 2$$

all arrive at the data stream, then $\{X := \mathsf{wt25}, T := 2\}$ is an answer to Q

detecting malfunctions in wind turbines

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

query: $Q = \mathsf{Shdn}(X, T)$

but: once

$$\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, 0)$$

arrives, we already know that $\{X := \mathsf{wt25}, T := 2\}$ *might* become an answer to Q

detecting malfunctions in wind turbines

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$

$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$

$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

query: $Q = \mathsf{Shdn}(X, T)$

in the original work, since

$$\mathsf{Temp}(\mathsf{wt42}, \mathsf{high}, 0)$$

does *not* arrive, we know that $\{X := \mathsf{wt42}, T := 2\}$ *cannot* become an answer to Q

detecting malfunctions in wind turbines

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

assumption (aaai'20)

we assume that the data stream D is complete at each time point, i.e. at time τ it contains all facts with timestamps $\leq \tau$

we call this set of facts the τ-*history* D_τ

detecting malfunctions in wind turbines

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

assumption (aaai'20)

we assume that the data stream D is complete at each time point, i.e. at time τ it contains all facts with timestamps $\leq \tau$
we call this set of facts the τ-*history* D_τ

current contribution

this work removes this assumption

Outline

syntax

ingredients

- temporal datalog: no negation, no function symbols, predicates have exactly one temporal argument (the last) – the *timestamp*
- a *datastream* D: a family $\{D|_\tau \mid \tau \in \mathbb{N}\}$, where $D|_\tau$ contains the facts that arrive at time instant τ
  $\rightsquigarrow$ for now: no delays – all elements of τ have timestamp τ
- a *program* Π: a set of rules defining additional predicates
- a query Q: a (typically non-ground) fact

terminology

two sorts of predicates

we assume that the predicate symbols occurring in D do not appear in heads of rules in Π

- *intensional predicates* are defined by their instances (from D)
- *extensional predicates* are defined by rules (in Π)

denotational semantics (i/ii)

hypothetical answers

a *hypothetical answer* to a query Q over a program Π and a history D_τ is a pair $\langle\theta, H\rangle$, where θ is a substitution and H is a finite set of ground extensional atoms (the hypotheses) such that:

- θ only instantiates variables free in Q
- H only contains atoms with timestamp $\tau' > \tau$
- $\Pi \cup D_\tau \cup H \models Q\theta$
- H is minimal with respect to set inclusion

$\rightsquigarrow$ the blue ingredients are new wrt classical sld-resolution

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \to \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \to \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \to \mathsf{Shdn}(X, T+1)$$

query

$Q = \mathsf{Shdn}(X, T)$

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

query

$Q = \mathsf{Shdn}(X, T)$

$\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, 0) \in D_0$

$\langle \{X := \mathsf{wt25}, T := 2\}, H \rangle$ is a hypothetical answer to Q for $H = \{\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, i) \mid i = 1, 2\}$

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \to \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \to \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \to \mathsf{Shdn}(X, T+1)$$

query

$Q = \mathsf{Shdn}(X, T)$

$\mathsf{Temp}(\mathsf{wt42}, \mathsf{high}, 0) \notin D_0$

$\langle \{X := \mathsf{wt42}, T := 2\}, H \rangle$ is not a hypothetical answer to Q for any H

denotational semantics (ii/ii)

supported answers

- a non-empty set of facts $E \subseteq D_\tau$ is *evidence* supporting a hypothetical answer $\langle \theta, H \rangle$ if E is a minimal set s.t. $\Pi \cup E \cup H \models P\theta$
- a *supported answer* to Q over D_τ is a triple $\langle \theta, H, E \rangle$ where E is evidence supporting $\langle \theta, H \rangle$

denotational semantics (ii/ii)

supported answers

- a non-empty set of facts $E \subseteq D_\tau$ is *evidence* supporting a hypothetical answer $\langle\theta, H\rangle$ if E is a minimal set s.t. $\Pi \cup E \cup H \models P\theta$
- a *supported answer* to Q over D_τ is a triple $\langle\theta, H, E\rangle$ where E is evidence supporting $\langle\theta, H\rangle$

in our example program

the fact

$$\mathrm{Temp}(\mathrm{wt25}, \mathrm{high}, 0) \in D_0$$

is evidence for the hypothetical answer

$$\langle\{X := \mathrm{wt25}, T := 2\}, \{\mathrm{Temp}(\mathrm{wt25}, \mathrm{high}, i) \mid i = 1, 2\}\rangle$$

Outline

intuition

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$

$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$

$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

intuition

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

$$\leftarrow \mathsf{Shdn}(X, T)$$

intuition

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \to \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \to \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \to \mathsf{Shdn}(X, T+1) \qquad \Leftarrow$$

$$\leftarrow \mathsf{Shdn}(X, T)$$
$$\downarrow$$
$$\leftarrow \mathsf{Cool}(X, T-1), \mathsf{Flag}(X, T)$$

intuition

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1) \qquad \Leftarrow$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

$$\leftarrow \mathsf{Shdn}(X, T)$$
$$\downarrow$$
$$\leftarrow \mathsf{Cool}(X, T-1), \mathsf{Flag}(X, T)$$
$$\downarrow$$
$$\leftarrow \mathsf{Flag}(X, T-2), \mathsf{Flag}(X, T-1), \mathsf{Flag}(X, T)$$

intuition

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T) \quad \Leftarrow$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

$$\leftarrow \mathsf{Shdn}(X, T)$$
$$\downarrow$$
$$\leftarrow \mathsf{Cool}(X, T-1), \mathsf{Flag}(X, T)$$
$$\downarrow$$
$$\leftarrow \mathsf{Flag}(X, T-2), \mathsf{Flag}(X, T-1), \mathsf{Flag}(X, T)$$
$$\downarrow^{*}$$
$$\leftarrow \mathsf{Temp}(X, \mathsf{high}, T-2), \mathsf{Temp}(X, \mathsf{high}, T-1), \mathsf{Temp}(X, \mathsf{high}, T)$$

future atom

an atom $P(t_1, \ldots, t_n)$ is a *future atom wrt* τ if the time term t_n either contains a temporal variable or is a time instant $t_n > \tau$

future atom

an atom $P(t_1, \ldots, t_n)$ is a *future atom wrt* τ if the time term t_n either contains a temporal variable or is a time instant $t_n > \tau$

sld-refutation, revisited

an *sld-refutation with future premises* of Π and Q over D_τ is a finite sld-derivation of $\Pi \cup D_\tau \cup \{\neg Q\}$ whose last goal only contains extensional future atoms wrt τ

future atom

an atom $P(t_1, \ldots, t_n)$ is a *future atom wrt* τ if the time term t_n either contains a temporal variable or is a time instant $t_n > \tau$

sld-refutation, revisited

an *sld-refutation with future premises* of Π and Q over D_τ is a finite sld-derivation of $\Pi \cup D_\tau \cup \{\neg Q\}$ whose last goal only contains extensional future atoms wrt τ

computed answer with premises

if $\mathcal{D}$ is an sld-refutation with future premises of Q over D_τ with last goal $G = \neg \wedge_i \alpha_i$ and θ is the restriction of the composition of the substitutions in $\mathcal{D}$ to $\mathrm{var}(Q)$, then $\langle \theta, \wedge_i \alpha_i \rangle$ is a *computed answer with premises* to Q over D_τ

independence of the computation rule

from classical results about sld-resolution, we can reorder the steps of any sld-refutation with future premises to use the facts from D_τ in temporal order

independence of the computation rule

from classical results about sld-resolution, we can reorder the steps of any sld-refutation with future premises to use the facts from D_τ in temporal order

key idea

this simple observation gives us an incremental algorithm

- at each step, update any "ongoing" derivations with the new facts
- any derivations expecting facts that did not arrive are forgotten
- some pre-processing allows us to identify relevant facts

a two-stage algorithm

pre-processing step

we compute answers with premises to Q over $D_{-1} = \emptyset$

- we store the minimal answers wrt set inclusion in a set $\mathcal{P}_Q$
- we initialize the set $\mathcal{S}_{-1}$ of *schematic supported answers* to $\emptyset$

a two-stage algorithm

pre-processing step

we compute answers with premises to Q over $D_{-1} = \emptyset$

- we store the minimal answers wrt set inclusion in a set $\mathcal{P}_Q$
- we initialize the set $\mathcal{S}_{-1}$ of *schematic supported answers* to $\emptyset$

online step

to compute $\mathcal{S}_{\tau+1}$ from $\mathcal{S}_\tau$ and $D_{\tau+1} \setminus D_\tau$:

- for each answer in $\mathcal{P}_Q$, we perform sld-resolution between its set of elements with minimal timestamps and $D_{\tau+1} \setminus D_\tau$
- for each element of $\mathcal{S}_\tau$, we perform sld-resolution between its set of elements with timestamp $\tau + 1$ and $D_{\tau+1} \setminus D_\tau$

each refutation yields an element in $\mathcal{S}_{\tau+1}$

termination (i)

under suitable assumptions, the pre-processing step terminates

termination (i)

under suitable assumptions, the pre-processing step terminates

termination (ii)

the online step terminates in polynomial time in the size of $\mathcal{S}_\tau$, $\mathcal{P}_Q$ and $D_{\tau+1} \setminus D_\tau$

termination (i)

under suitable assumptions, the pre-processing step terminates

termination (ii)

the online step terminates in polynomial time in the size of $\mathcal{S}_\tau$, $\mathcal{P}_Q$ and $D_{\tau+1} \setminus D_\tau$

soundness

every instantiation of an element of $\mathcal{S}_\tau$ is a supported answer to Q over Π and D_τ

termination (i)

under suitable assumptions, the pre-processing step terminates

termination (ii)

the online step terminates in polynomial time in the size of $\mathcal{S}_\tau$, $\mathcal{P}_Q$ and $D_{\tau+1} \setminus D_\tau$

soundness

every instantiation of an element of $\mathcal{S}_\tau$ is a supported answer to Q over Π and D_τ

completeness

every supported answer to Q over Π and D_τ is an instantiation of an element of $\mathcal{S}_\tau$

Outline

1. introduction
2. denotational semantics
3. operational semantics
4. what about delays?
5. conclusions

but in practice. . .

communication delays

communication is not perfect

- there are delays
- there are errors
- there are losses

but in practice. . .

communication delays

communication is not perfect

- there are delays
- there are errors
- there are losses

delays are the most essential!

good communication protocols can ensure* no errors and no losses

(* with a high enough probability assuming no hardware failures)

a revised hypothesis

delay

for each fact α we assume there is a value $\delta(\alpha)$ (the *delay* of α) such that: if the timestamp of α is τ, then α can only appear in $D|_{\tau}, \ldots, D|_{\tau+\delta(\alpha)}$

$\rightsquigarrow$ reasonable in practice

a revised hypothesis

delay

for each fact α we assume there is a value $\delta(\alpha)$ (the *delay* of α) such that: if the timestamp of α is τ, then α can only appear in $D|_{\tau}, \ldots, D|_{\tau+\delta(\alpha)}$

- extends in the obvious way: if some of $t_1, \ldots, t_n$ are not ground, then $\delta(P(t_1, \ldots, t_n)) = \max\{\delta(P(t'_1, \ldots, t'_n))\}$ where $(t'_1, \ldots, t'_n)$ range over the ground instances of $t_1, \ldots, t_n$
- in particular, $\delta(P) = \delta(P(X_1, \ldots, X_n)) = \max\{\delta(P(t'_1, \ldots, t'_n))\}$ where $(t'_1, \ldots, t'_n)$ range over all (valid) ground terms
- we assume $\delta(P) < \infty$

do the old definitions work?

denotational semantics: only minor tweaks

an atom $P(t_1, \ldots, t_n)$ is *future-possible wrt* τ if the time term t_n either contains a temporal variable or is a time instant $t_n > \tau - \delta(P(t_1, \ldots, t_n))$

$\rightsquigarrow$ all definitions work, replacing *future* with *future-possible*

do the old definitions work?

denotational semantics: only minor tweaks

an atom $P(t_1, \ldots, t_n)$ is *future-possible wrt* τ if the time term t_n either contains a temporal variable or is a time instant $t_n > \tau - \delta(P(t_1, \ldots, t_n))$

$\leadsto$ all definitions work, replacing *future* with *future-possible*

but operational semantics, alas...

one must be careful with unification

- previously, schematic hypothetical answers were progressively unified; non-generated substitutions were irrelevant
- in the presence of delays, unification may lose answers

a look at the problem

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

$D|_0 = \{\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, 0)\}$

- $\langle \{X := \mathsf{wt25}, T := 2\}, H \rangle$ is a hypothetical answer to Q for $H = \{\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, i) \mid i = 1, 2\}$
- $\langle \{X := \mathsf{wt42}, T := 2\}, H \rangle$ *is* a hypothetical answer to Q for $H = \{\mathsf{Temp}(\mathsf{wt42}, \mathsf{high}, i) \mid i = 0, 1, 2\}$

(assuming $\delta(\mathsf{Temp}) > 0$)

a look at the problem

our example program

$$\mathsf{Temp}(X, \mathsf{high}, T) \rightarrow \mathsf{Flag}(X, T)$$
$$\mathsf{Flag}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Cool}(X, T+1)$$
$$\mathsf{Cool}(X, T) \wedge \mathsf{Flag}(X, T+1) \rightarrow \mathsf{Shdn}(X, T+1)$$

$D|_0 = \{\mathsf{Temp}(\mathsf{wt25}, \mathsf{high}, 0)\}$

but unifying

$$\{\mathsf{Temp}(X, \mathsf{high}, T), \mathsf{Temp}(X, \mathsf{high}, T+1), \mathsf{Temp}(X, \mathsf{high}, T+2)\}$$

with $D|_0$ will only keep the first answer...

solution: local mgus

we must be able to wait

uninstantiated answers must be kept in case they may be unified later

- theoretically: a local mgu (see paper)
- in practice: consider *all* substitutions (not just the leaves) appearing in *any* sld-derivation (order matters)

solution: local mgus

we must be able to wait

uninstantiated answers must be kept in case they may be unified later

- theoretically: a local mgu (see paper)
- in practice: consider *all* substitutions (not just the leaves) appearing in *any* sld-derivation (order matters)

was there a typo in there?

consider all substitutions appearing in any sld-derivation

- yes, that's an exponential blowup
- hopefully, minor in practice
- ...and it's *substitutions*, not nodes (so not all is lost)

a hint of the complexity

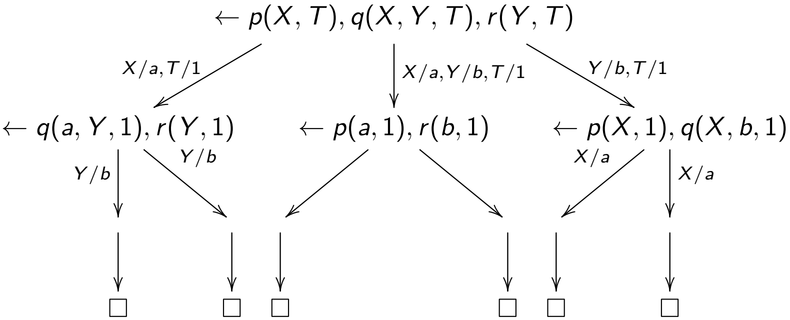
minimal example

hypotheses: $\{p(X, T), q(X, Y, T), r(Y, T)\}$
data stream: $\{p(a, 1), q(a, b, 1), r(b, 1)\}$

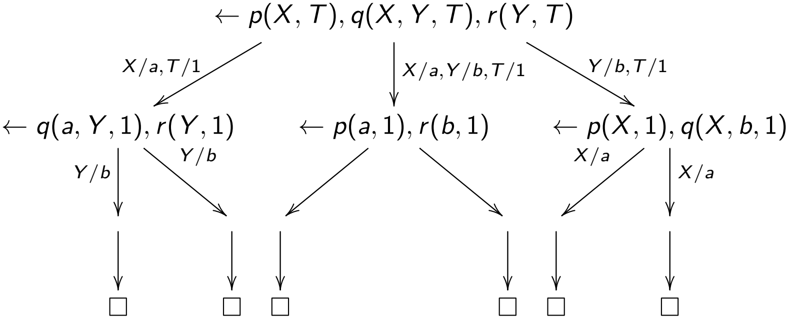
a hint of the complexity

minimal example

hypotheses: $\{p(X,T), q(X,Y,T), r(Y,T)\}$
data stream: $\{p(a,1), q(a,b,1), r(b,1)\}$



a hint of the complexity



only 4 instantiations

- we have e.g. $[X := a, T := 1]$ and $\emptyset$, but not $[T := 1]$

a hint of the complexity

minimal example

hypotheses: $\{p(X,T), q(X,Y,T), r(Y,T)\}$
data stream: $\{p(a,1), q(a,b,1), r(b,1)\}$

only 4 schematic answers

we now apply each substitution to the set of hypotheses and unify with the data stream to obtain schematic answers

- $\langle \emptyset, \emptyset, \{p(X,T), q(X,Y,T), r(Y,T)\}\rangle$
- $\langle [X := a, T := 1], \{p(a,1)\}, \{q(a,Y,1), r(Y,1)\}\rangle$
- $\langle [Y := b, T := 1], \{r(b,1)\}, \{p(X,1), q(X,b,1)\}\rangle$
- $\langle [X := a, Y := b, T := 1], \{p(a,1), q(a,b,1), r(b,1)\}, \emptyset\rangle$

Outline

main achievements

achievements:

- extended previous formalism with communication delays
- denotational semantics for more general setting
- operational semantics for more general setting, uses *local mgu*

what we lost:

- negation :-(

future work

- negation :-)
- an implementation. . .

thank you!