Data Complexity of Natural and Controlled Languages

Camilo Thorne[1]

[1] **KRDB Research Centre for Knowledge and Data**
Free University of Bozen-Bolzano
cthorne@inf.unibz.it
http://www.inf.unibz.it/∼cathorne

DCC UdC - Concepción - 5/8/2011

Outline

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

EX: Consider: John saw a man with a telescope

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

EX: Consider: John saw a man with a telescope (**PP**-attachment ambiguity)

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

EX: Consider: John saw a man with a telescope $\Rightarrow$ one possibility!

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

EX: Consider: John saw a man with a telescope $\Rightarrow$ another possibility!

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

EX: Consider: John saw a man with a telescope $\Rightarrow$ two interpretations!

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English

- But, natural language is ambigous

EX: Consider: John saw a man with a telescope $\Rightarrow$ two interpretations!

1. ambiguity blows up processing (exponentially many semantic representations)
2. worst case $\Rightarrow$ computationally unsolvable

Motivation

- Suppose a system that on input

  Every Italian loves pasta and football
  Silvio is Italian

  outputs

  Silvio loves pasta

- Requires deep semantic analysis of English
- But, natural language is ambigous

EX: Consider: John saw a man with a telescope $\Rightarrow$ two interpretations!

1. ambiguity blows up processing (exponentially many semantic representations)
2. worst case $\Rightarrow$ computationally unsolvable

QS: What if we remove ambiguity?

Controlled Languages (CL)

- Controlled language (CLs) are ambiguity-free subset of natural languages

1. polynomially translate to logic meaning representations φ
2. translation modelled by formal semantics compositional translations $\tau(\cdot)$

Controlled Languages (CL)

- Controlled language (CLs) are ambiguity-free subset of natural languages

1. polynomially translate to logic meaning representations φ
2. translation modelled by formal semantics compositional translations $\tau(\cdot)$

- Defined by constraining syntax, semantics and vocabulary
- Standard machinery of formal semantics [Montague 1970]
- Controlled languages interfaces proposed [Sowa 2004, Fuchs et al. 2006]

Controlled Languages (CL)

- Controlled language (CLs) are ambiguity-free subset of natural languages

1. polynomially translate to logic meaning representations φ
2. translation modelled by formal semantics compositional translations $\tau(\cdot)$

- Defined by constraining syntax, semantics and vocabulary
- Standard machinery of formal semantics [Montague 1970]
- Controlled languages interfaces proposed [Sowa 2004, Fuchs et al. 2006]

QS: Efficient translation, but how costly is logical reasoning?

Controlled Languages - COP [Pratt & Third 2006]

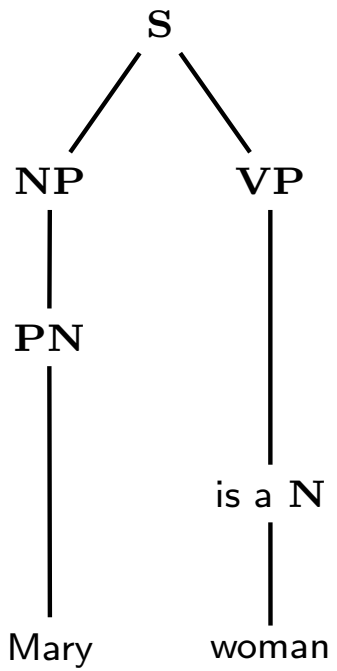
Syntax Rules	Semantics $(= \tau(\cdot))$
$\mathbf{S} \longrightarrow \mathbf{NP\ VP}$	$\tau(\mathbf{NP})(\tau(\mathbf{VP})) \rhd \tau(\mathbf{S})$
$\mathbf{VP} \longrightarrow$ is a $\mathbf{N}$	$\tau(\mathbf{VP}) = \tau(\mathbf{N})$
$\mathbf{VP} \longrightarrow$ is not a $\mathbf{N}$	$\tau(\mathbf{VP}) = \neg\tau(\mathbf{N})$
$\mathbf{NP} \longrightarrow \mathbf{PN}$	$\tau(\mathbf{NP}) = \tau(\mathbf{PN})$
$\mathbf{NP} \longrightarrow \mathbf{Det\ N}$	$\tau(\mathbf{Det})(\tau(\mathbf{N})) \rhd \tau(\mathbf{NP})$

Controlled Languages - COP [Pratt & Third 2006]

Syntax Rules	Semantics $(= \tau(\cdot))$
$\mathbf{S} \longrightarrow \mathbf{NP\ VP}$	$\tau(\mathbf{NP})(\tau(\mathbf{VP})) \rhd \tau(\mathbf{S})$
$\mathbf{VP} \longrightarrow$ is a $\mathbf{N}$	$\tau(\mathbf{VP}) = \tau(\mathbf{N})$
$\mathbf{VP} \longrightarrow$ is not a $\mathbf{N}$	$\tau(\mathbf{VP}) = \neg\tau(\mathbf{N})$
$\mathbf{NP} \longrightarrow \mathbf{PN}$	$\tau(\mathbf{NP}) = \tau(\mathbf{PN})$
$\mathbf{NP} \longrightarrow \mathbf{Det\ N}$	$\tau(\mathbf{Det})(\tau(\mathbf{N})) \rhd \tau(\mathbf{NP})$

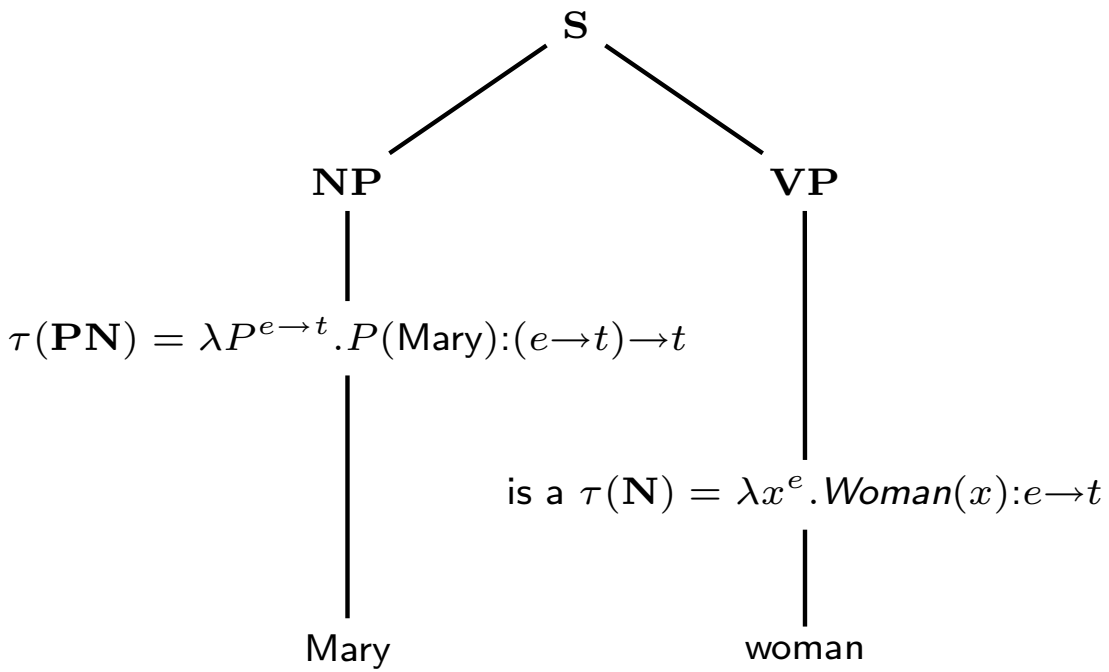
Lexicon	Semantics $(= \tau(\cdot))$
$\mathbf{N} \longrightarrow$ woman	$\tau(\mathbf{N}) = \lambda x^e.\mathit{Woman}(x){:}e{\to}t$
$\mathbf{N} \longrightarrow$ man	$\tau(\mathbf{N}) = \lambda x^e.\mathit{Man}(x){:}e{\to}t$
$\mathbf{PN} \longrightarrow$ Mary	$\tau(\mathbf{PN}) = \lambda P^{e\to t}.P(\mathsf{Mary}){:}(e{\to}t){\to}t$
$\mathbf{Det} \longrightarrow$ every	$\tau(\mathbf{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\forall x^e(P(x) \Rightarrow Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$
$\mathbf{Det} \longrightarrow$ some	$\tau(\mathbf{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\exists x^e(P(x) \wedge Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$
$\mathbf{Det} \longrightarrow$ no	$\tau(\mathbf{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\forall x^e(P(x) \Rightarrow \neg Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$

Controlled Languages - COP (cntd.)



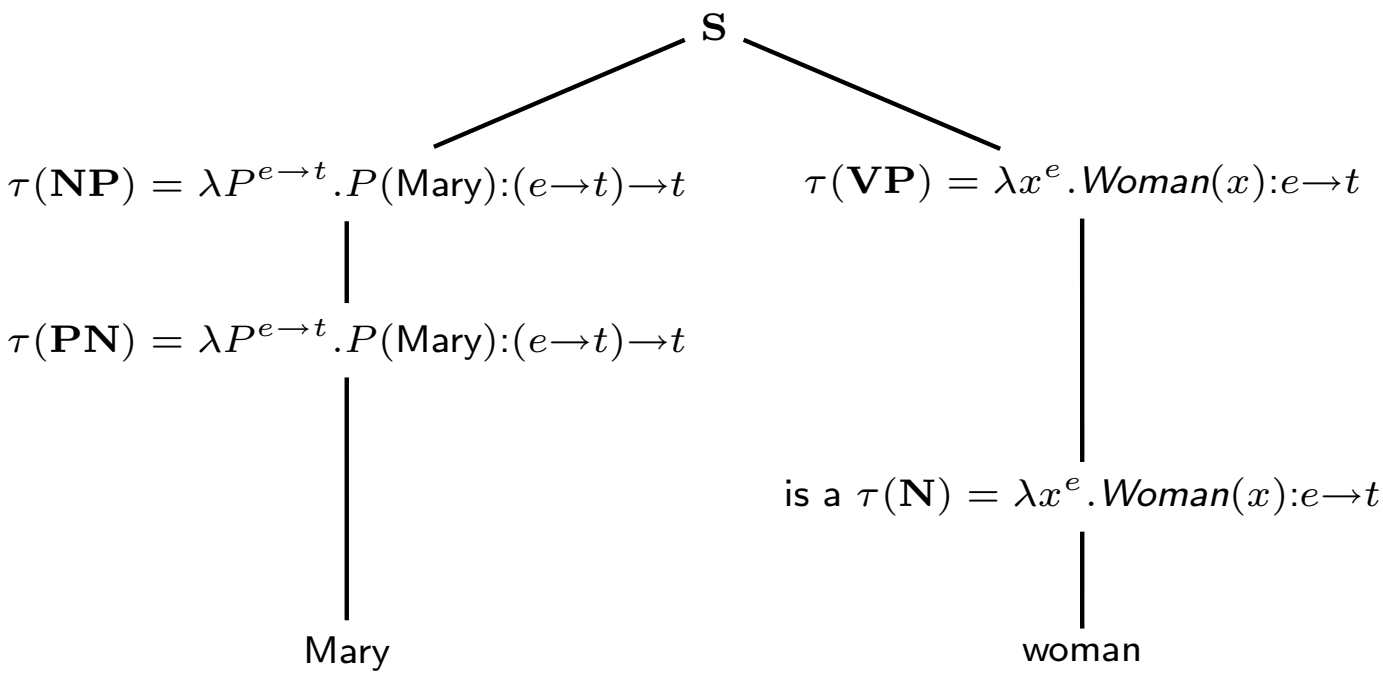
Parsing and interpreting "Mary is a woman"

Controlled Languages - COP (cntd.)



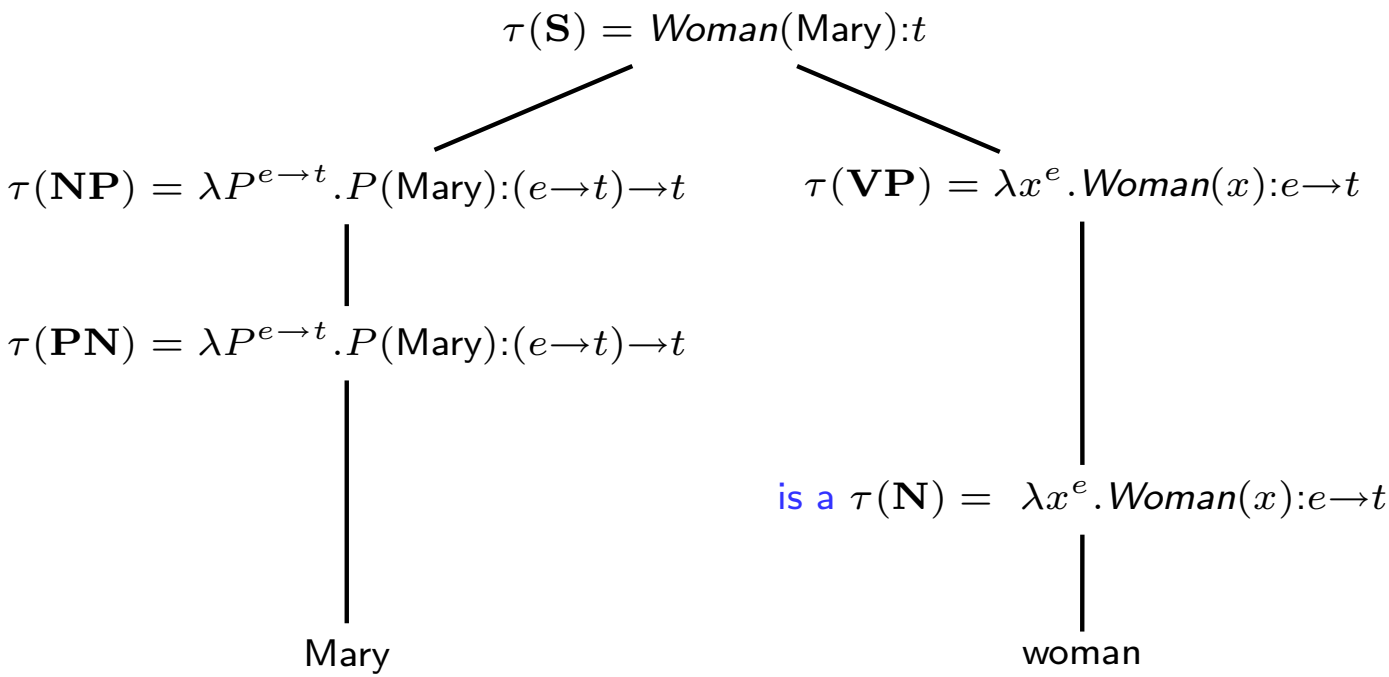
Parsing and interpreting "Mary is a woman"

Controlled Languages - COP (cntd.)



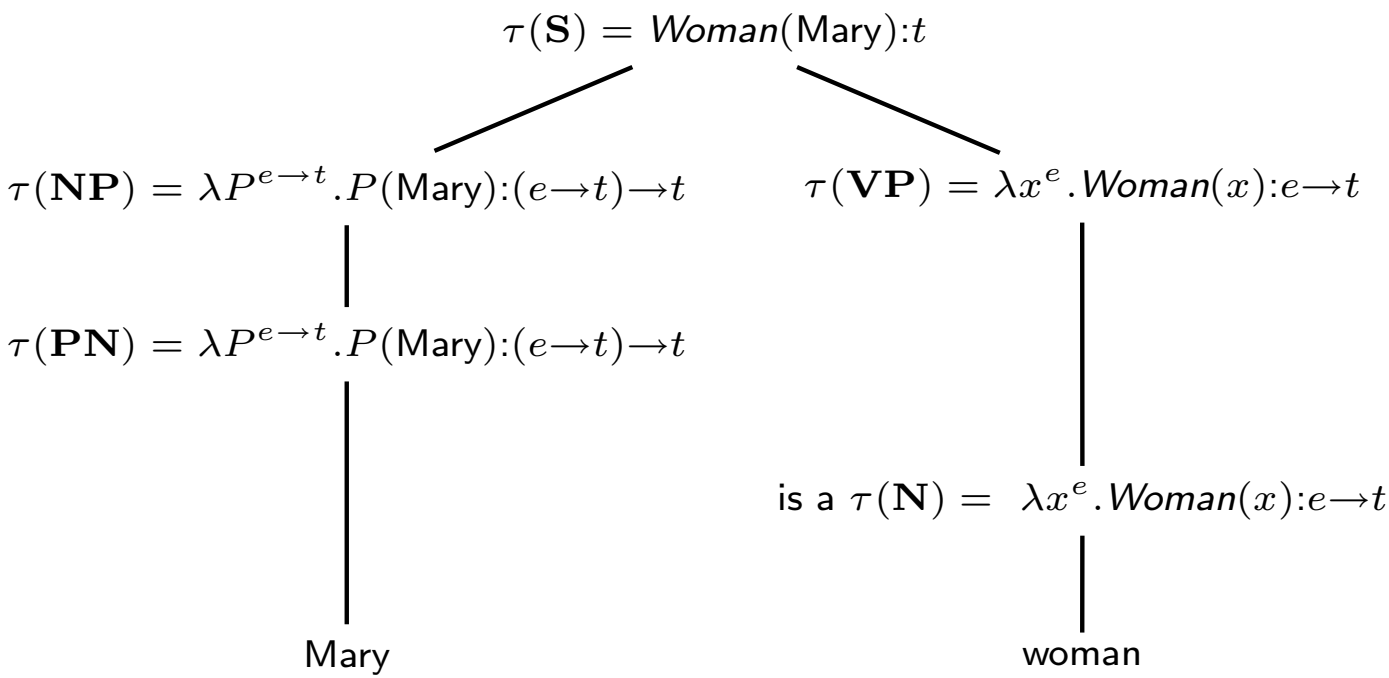
Parsing and interpreting "Mary is a woman"

Controlled Languages - COP (cntd.)



Parsing and interpreting "Mary is a woman"

Controlled Languages - COP (cntd.)



Parsing and interpreting "Mary is a woman"

NB: Computing $\tau(\cdot)$ is "easy" for context-free fragments

1. parsing: polynomial in the input utterance
2. interpreting: linear in the the parse tree

Engendered Logics and Complexity

- Modulo $\tau(\cdot)$ controlled fragments express a fragment of FO

EX: COP expresses

$\mathit{Woman}(\mathrm{Mary})$	Mary is a woman
$\neg \mathit{Man}(\mathrm{Mary})$	Mary is not a man.
$\forall x(\mathit{Man}(x) \Rightarrow \mathit{Person}(x))$	Every man is a person
$\forall x(\mathit{Woman}(x) \Rightarrow \neg \mathit{Man}(x))$	No woman is a man
$\forall x(\mathit{Person}(x) \Rightarrow \mathit{Human}(x))$	Every person is a human
$\forall x(\mathit{Person}(x) \Rightarrow \mathit{Human}(x))$	Every person is a human
$\exists x(\mathit{Person}(x) \wedge \mathit{Woman}(x))$	Some person is a woman
$\exists x(\mathit{Person}(x) \wedge \neg \mathit{Woman}(x))$	Some person is not a woman

Engendered Logics and Complexity

- Modulo $\tau(\cdot)$ controlled fragments express a fragment of FO

EX: COP expresses

$\mathit{Woman}(\mathrm{Mary})$	Mary is a woman
$\neg \mathit{Man}(\mathrm{Mary})$	Mary is not a man.
$\forall x(\mathit{Man}(x) \Rightarrow \mathit{Person}(x))$	Every man is a person
$\forall x(\mathit{Woman}(x) \Rightarrow \neg \mathit{Man}(x))$	No woman is a man
$\forall x(\mathit{Person}(x) \Rightarrow \mathit{Human}(x))$	Every person is a human
$\forall x(\mathit{Person}(x) \Rightarrow \mathit{Human}(x))$	Every person is a human
$\exists x(\mathit{Person}(x) \wedge \mathit{Woman}(x))$	Some person is a woman
$\exists x(\mathit{Person}(x) \wedge \neg \mathit{Woman}(x))$	Some person is not a woman

- By studying such FO fragments

1. exploit computational logic $\Rightarrow$ computational semantics

Engendered Logics and Complexity

- Modulo $\tau(\cdot)$ controlled fragments express a fragment of FO

EX: COP expresses

$\mathit{Woman}(\mathrm{Mary})$	Mary is a woman
$\neg \mathit{Man}(\mathrm{Mary})$	Mary is not a man.
$\forall x(\mathit{Man}(x) \Rightarrow \mathit{Person}(x))$	Every man is a person
$\forall x(\mathit{Woman}(x) \Rightarrow \neg \mathit{Man}(x))$	No woman is a man
$\forall x(\mathit{Person}(x) \Rightarrow \mathit{Human}(x))$	Every person is a human
$\forall x(\mathit{Person}(x) \Rightarrow \mathit{Human}(x))$	Every person is a human
$\exists x(\mathit{Person}(x) \wedge \mathit{Woman}(x))$	Some person is a woman
$\exists x(\mathit{Person}(x) \wedge \neg \mathit{Woman}(x))$	Some person is not a woman

- By studying such FO fragments

1. exploit computational logic $\Rightarrow$ computational semantics
2. understand semantic processing complexity $\Rightarrow$ semantic complexity

A Survery of Controlled Languages

CL (English)	Maps to	Goal
FOEs [Pratt & Third 2005]	FO fragments	Knowledge representation
ACE [Fuchs 2005]	FO	Knowledge representation
ACE-OWL [Kaaljurand 2007]	OWL-DL	Ontology authoring, querying
PENG [Schwitter 2003]	OWL-DL	Ontology authoring, querying
SOS [Schwitter2008]	OWL-DL	Ontology authoring, querying
CLCE [Sowa2004]	FOL	Knowledge representation
AECMA [Unwalla 2005]	?	User specifications
English Query (EQ) [Blum 1999]	SQL	DB querying/management
OWL-CNL [Schwitter 2006]	OWL-DL	Ontology authoring
Easy English [Bernth 1998]	?	User specifications
λ-SQL [Winter 2006]	SQL	Database querying
nRQL [Schwitter 2008]	FO queries	Ontology querying
Rabbit [Schwitter2008]	OWL	Ontology authoring
ACE-PQL [Bernstein 2005]	PQL	Ontology querying
QE-III [Clifford 1987]	IL	Database querying

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

1 SAT: Given quantified sentences S and facts D, is $\tau(S) \cup D$ satisfiable?

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

1. SAT: Given quantified sentences S and facts D, is $\tau(S) \cup D$ satisfiable?
2. QA: Given quantified sentences S, facts D and question Q, does $\tau(S) \cup F$ logically entail $\tau(Q)$?

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

1. SAT: Given quantified sentences S and facts D, is $\tau(S) \cup D$ satisfiable?
2. QA: Given quantified sentences S, facts D and question Q, does $\tau(S) \cup F$ logically entail $\tau(Q)$?

- Fine-grained analysis of semantic complexity possible

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

1. SAT: Given quantified sentences S and facts D, is $\tau(S) \cup D$ satisfiable?
2. QA: Given quantified sentences S, facts D and question Q, does $\tau(S) \cup F$ logically entail $\tau(Q)$?

- Fine-grained analysis of semantic complexity possible

1. if measured only in D: data complexity

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

1. SAT: Given quantified sentences S and facts D, is $\tau(S) \cup D$ satisfiable?
2. QA: Given quantified sentences S, facts D and question Q, does $\tau(S) \cup F$ logically entail $\tau(Q)$?

- Fine-grained analysis of semantic complexity possible

1. if measured only in D: data complexity
2. if measured in D, S and Q: combined complexity

Semantic Complexity

- We are interested in the semantic complexity of CLs
- Semantic complexity $\Rightarrow$ computational complexity of decision problems

1. SAT: Given quantified sentences S and facts D, is $\tau(S) \cup D$ satisfiable?
2. QA: Given quantified sentences S, facts D and question Q, does $\tau(S) \cup F$ logically entail $\tau(Q)$?

- Fine-grained analysis of semantic complexity possible

1. if measured only in D: data complexity
2. if measured in D, S and Q: combined complexity

NB: Possible when semantic complexity is decidable

The Fragments of English (FOEs) [Pratt & Third 2006]

FOE	Constructs
COP	Copula, common and proper nouns, negation, universal, existential quantifiers
COP+Rel	COP plus relative pronouns
COP+TV	COP plus transitive verbs
COP+TV+DTV	COP+TV plus ditransitive verbs
COP+Rel+TV	COP+Rel plus transitive verbs
COP+Rel+TV+DTV	COP+Rel+TV plus ditransitive verbs
COP+Rel+TV+RA	COP+Rel+TV plus anaphoric pronouns (e.g., he, him, it, herself) of bounded scope
COP+Rel+TV+GA	COP+Rel+TV plus unbounded anaphoric pronouns
COP+Rel+TV+DTV+RA	COP+Rel+TV+DTV plus bounded anaphoric pronouns

Combined Complexity of the FOEs [Pratt & Third 2006]

FOE	SAT (combined)
COP	in NL
COP+TV	NL-complete
COP+TV+DTV	in P
COP+Rel	NP-complete
COP+TV+Rel	EXP-complete
COP+TV+Rel+RA	EXP-complete
COP+TV+DTV+Rel	NEXP-complete
COP+TV+Rel+GA	undecidable

Different function words yield different complexity!

Combined Complexity of the FOEs [Pratt & Third 2006]

FOE	SAT (combined)
COP	in NL
COP+TV	NL-complete
COP+TV+DTV	in P
COP+Rel	NP-complete
COP+TV+Rel	EXP-complete
COP+TV+Rel+RA	EXP-complete
COP+TV+DTV+Rel	NEXP-complete
COP+TV+Rel+GA	undecidable

Not "Boolean-closed" $\Rightarrow$ tractable!

Combined Complexity of the FOEs [Pratt & Third 2006]

FOE	SAT (combined)
COP	in NL
COP+TV	NL-complete
COP+TV+DTV	in P
COP+Rel	NP-complete
COP+TV+Rel	EXP-complete
COP+TV+Rel+RA	EXP-complete
COP+TV+DTV+Rel	NEXP-complete
COP+TV+Rel+GA	undecidable

"Boolean-closed" $\Rightarrow$ intractable!

Combined Complexity of the FOEs [Pratt & Third 2006]

FOE	SAT (combined)
COP	in NL
COP+TV	NL-complete
COP+TV+DTV	in P
COP+Rel	NP-complete
COP+TV+Rel	EXP-complete
COP+TV+Rel+RA	EXP-complete
COP+TV+DTV+Rel	NEXP-complete
COP+TV+Rel+GA	undecidable

"Boolean-closed" + restricted anaphora $\Rightarrow$ decidable!

Combined Complexity of the FOEs [Pratt & Third 2006]

FOE	SAT (combined)
COP	in NL
COP+TV	NL-complete
COP+TV+DTV	in P
COP+Rel	NP-complete
COP+TV+Rel	EXP-complete
COP+TV+Rel+RA	EXP-complete
COP+TV+DTV+Rel	NEXP-complete
COP+TV+Rel+GA	undecidable

"Boolean-closed" + full anaphora $\Rightarrow$ undecidable!

Data Complexity of the FOEs [Thorne 2011]

- Data complexity [Vardi 1982] measures whether reasoning scales to large data repositories
- Typical scenarios: interfaces to knowledge bases, ontologies
- It allows a substantial increase in expressivity:

	QA	SAT
COP	in L	in L
COP+TV	in P	in L
COP+TV+DTV	in coNP	in L
COP+Rel	coNP-complete	in L
COP+Rel+TV	coNP-complete	NP-complete
COP+Rel+DTV	coNP-complete	NP-complete
COP+Rel+DTV+TV	coNP-complete	NP-complete

NB: Questions express conjunction and existential quantification

Data Complexity of the FOEs [Thorne 2011]

- Data complexity [Vardi 1982] measures whether reasoning scales to large data repositories
- Typical scenarios: interfaces to knowledge bases, ontologies
- It allows a substantial increase in expressivity:

	QA	SAT
COP	in L	in L
COP+TV	in P	in L
COP+TV+DTV	in coNP	in L
COP+Rel	coNP-complete	in L
COP+Rel+TV	coNP-complete	NP-complete
COP+Rel+DTV	coNP-complete	NP-complete
COP+Rel+DTV+TV	coNP-complete	NP-complete

NB: Questions express conjunction and existential quantification

Data Complexity of the FOEs [Thorne 2011]

- Data complexity [Vardi 1982] measures whether reasoning scales to large data repositories
- Typical scenarios: interfaces to knowledge bases, ontologies
- But "Boolean-closedness" hits in quickly!

	QA	SAT
COP	in L	in L
COP+TV	in P	in L
COP+TV+DTV	in coNP	in L
COP+Rel	coNP-complete	in L
COP+Rel+TV	coNP-complete	NP-complete
COP+Rel+DTV	coNP-complete	NP-complete
COP+Rel+DTV+TV	coNP-complete	NP-complete

NB: Questions express conjunction and existential quantification

Ontology Languages

Semantic Web Language
(OWL)

```
<owl:Class rdf:about="#Employee">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#develops"/>
      <owl:someValuesFrom
        rdf:resource="#Project"/>
    </owl:Restriction>
  </rdfs:SubclassOf>
</owl:Class>
```

Description Logics
+ CL

$$Employee \sqsubseteq \exists develops{:}Project$$

Every employee develops
some project

- OWL is a machine-readable language developed for the semantic web
- CLs are human-readable, yet as unambigous as OWL

$\mathcal{ALCI}$ Ontologies and Ontology Languages

- Given concept names A and role names r, roles R and left C_l or right C_r concepts are defined by

$$R \longrightarrow r \mid r^- \qquad C_f \longrightarrow A \mid \exists R{:}C'_f \mid \neg C'_f \mid C'_f \sqcap C''_f, f \in \{l, r\}$$

- An assertion is a formula $C_l \sqsubseteq C_r$
- An ontology $\mathcal{O}$ is a set of assertions

$\mathcal{ALCI}$ Ontologies and Ontology Languages

- Given concept names A and role names r, roles R and left C_l or right C_r concepts are defined by

$$R \longrightarrow r \mid r^- \qquad C_f \longrightarrow A \mid \exists R{:}C'_f \mid \neg C'_f \mid C'_f \sqcap C''_f, f \in \{l, r\}$$

- An assertion is a formula $C_l \sqsubseteq C_r$
- An ontology $\mathcal{O}$ is a set of assertions

EX: *Manager* $\sqsubseteq$ *Employee* $\Rightarrow$ "every manager is an employee"

- Open world semantics
  1. concepts denote sets
  2. roles denote relations
  3. assertions denote set inclusion

$\mathcal{ALCI}$ Ontologies and Ontology Languages

- Given concept names A and role names r, roles R and left C_l or right C_r concepts are defined by

$$R \longrightarrow r \mid r^- \qquad C_f \longrightarrow A \mid \exists R{:}C'_f \mid \neg C'_f \mid C'_f \sqcap C''_f, f \in \{l, r\}$$

- An assertion is a formula $C_l \sqsubseteq C_r$
- An ontology $\mathcal{O}$ is a set of assertions

EX: *Manager* $\sqsubseteq$ *Employee* $\Rightarrow$ "every manager is an employee"

- Open world semantics
  1. concepts denote sets
  2. roles denote relations
  3. assertions denote set inclusion

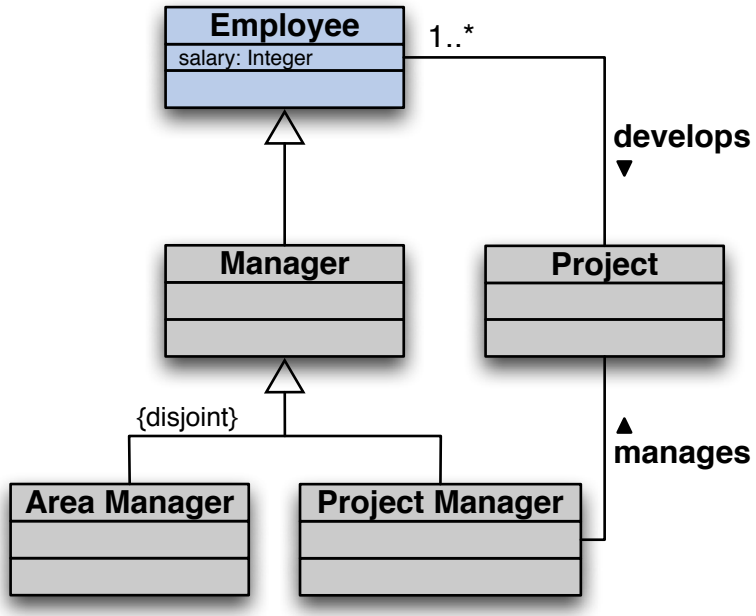
NB: If right or left concepts restricted $\Rightarrow$ ontology languages

Conceptual models and OBDA

- Description logic ontologies capture conceptual data models (UML class diagrams, ER-diagrams, etc.)
- In ontology-based data access (OBDA), relational data is accessed via such ontologies

Conceptual models and OBDA

- Description logic ontologies capture conceptual data models (UML class diagrams, ER-diagrams, etc.)
- In ontology-based data access (OBDA), relational data is accessed via such ontologies



$$\begin{aligned}
\mathit{Manager} &\sqsubseteq \mathit{Employee}\\
\mathit{AreaManager} &\sqsubseteq \mathit{Manager}\\
\mathit{ProjectManager} &\sqsubseteq \mathit{Manager}\\
\mathit{AreaManager} &\sqsubseteq \neg \mathit{ProjectManager}
\end{aligned}$$

$$\begin{aligned}
\exists \mathit{develops} &\sqsubseteq \mathit{Employee}\\
\exists \mathit{develops}^{-} &\sqsubseteq \mathit{Project}\\
\mathit{Project} &\sqsubseteq \exists \mathit{develops}^{-}
\end{aligned}$$

$$\begin{aligned}
\exists \mathit{manages} &\sqsubseteq \mathit{TopManager}\\
\exists \mathit{manages}^{-} &\sqsubseteq \mathit{Project}\\
\mathit{TopManager} &\sqsubseteq \exists \mathit{manages}\\
\mathit{Project} &\sqsubseteq \exists \mathit{manages}^{-}
\end{aligned}$$

$$\mathit{Employee} \sqsubseteq \exists \mathit{has}{:}\mathit{Salary}$$

Tree Shaped Queries

- A tree shaped query (TSQ) is defined by

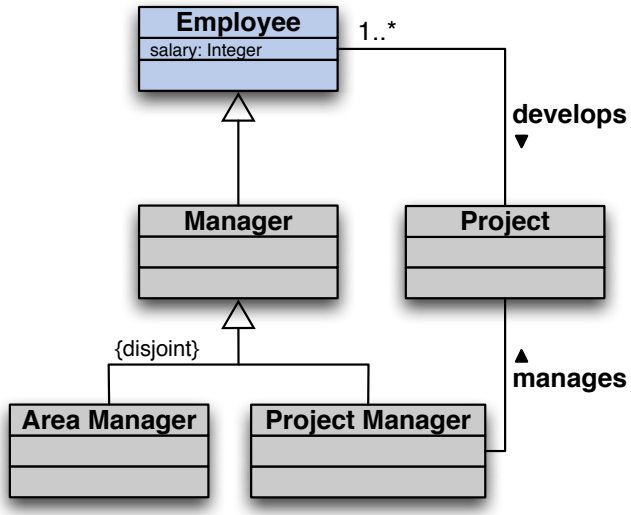
$$\phi(x) \longrightarrow A(x) \mid \exists y R(x,y) \mid \phi(x) \wedge \phi'(x) \mid \exists y(R(x,y) \wedge \phi(x))$$

Tree Shaped Queries

- A tree shaped query (TSQ) is defined by

$$\phi(x) \longrightarrow A(x) \mid \exists y R(x,y) \mid \phi(x) \wedge \phi'(x) \mid \exists y(R(x,y) \wedge \phi(x))$$

- Subset of SQL SELECT-PROJECT-JOIN queries



$$\mathit{Manager}(x) \wedge \mathit{ProjectManager}(x) \wedge \\ \wedge \exists y(\mathit{manages}(x,y) \wedge \mathit{Project}(y))$$

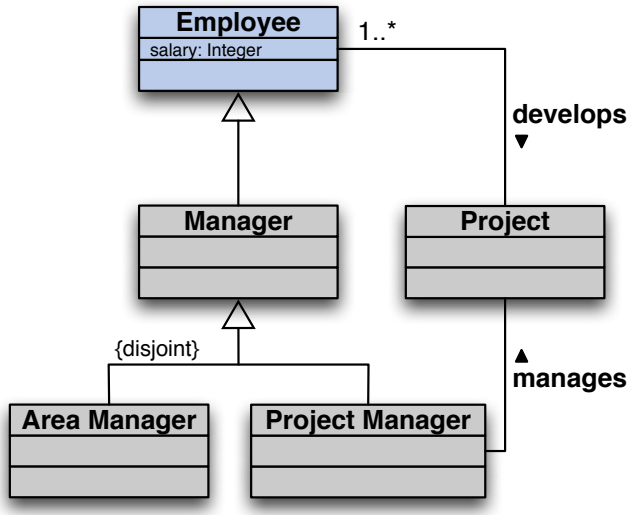
```
SELECT Manager.MName
  FROM Manager,ProjectManager, manages,Project
  WHERE Manager.MName=ProjectManager.MName
    AND Manager.MName=manages.MName
    AND Project.PName=manages.PName
```

Tree Shaped Queries

- A tree shaped query (TSQ) is defined by

$$\phi(x) \longrightarrow A(x) \mid \exists y R(x,y) \mid \phi(x) \wedge \phi'(x) \mid \exists y(R(x,y) \wedge \phi(x))$$

- Subset of SQL SELECT-PROJECT-JOIN queries



$$\mathit{Manager}(x) \wedge \mathit{ProjectManager}(x) \wedge \\ \wedge \exists y(\mathit{manages}(x,y) \wedge \mathit{Project}(y))$$

```
SELECT Manager.MName
  FROM Manager,ProjectManager, manages,Project
  WHERE Manager.MName=ProjectManager.MName
    AND Manager.MName=manages.MName
    AND Project.PName=manages.PName
```

NB: Semantics of query answering $\Rightarrow$ form of logical entailment!

The IS-A$_i$s [Thorne & Calvanese 2009]

QS: What is the data complexity of CLs that map to OWL?

The IS-A$_i$s [Thorne & Calvanese 2009]

QS: What is the data complexity of CLs that map to OWL?

- We want to single out

1. the maximal CLs that are tractable w.r.t. data complexity
2. the minimal CLs that are intractable w.r.t. data complexity

- Utterances should translate into assertions $C_l \sqsubseteq C_r$
- We partition [Bernardi et al. 2007] $\mathbf{N}$s and $\mathbf{VP}$s into

1. left components: $\mathbf{N}_l$ and $\mathbf{VP}_l$
2. right components: $\mathbf{N}_r$ and $\mathbf{VP}_r$

The IS-A$_i$s [Thorne & Calvanese 2009]

QS: What is the data complexity of CLs that map to OWL?

- We want to single out

1. the maximal CLs that are tractable w.r.t. data complexity
2. the minimal CLs that are intractable w.r.t. data complexity

- Utterances should translate into assertions $C_l \sqsubseteq C_r$
- We partition [Bernardi et al. 2007] $\mathbf{N}$s and $\mathbf{VP}$s into

1. left components: $\mathbf{N}_l$ and $\mathbf{VP}_l$
2. right components: $\mathbf{N}_r$ and $\mathbf{VP}_r$

- Engenders a family of CLs

The IS-A$_i$s [Thorne & Calvanese 2009]

QS: What is the data complexity of CLs that map to OWL?

- We want to single out

1. the maximal CLs that are tractable w.r.t. data complexity
2. the minimal CLs that are intractable w.r.t. data complexity

- Utterances should translate into assertions $C_l \sqsubseteq C_r$
- We partition [Bernardi et al. 2007] $\mathbf{N}$s and $\mathbf{VP}$s into

1. left components: $\mathbf{N}_l$ and $\mathbf{VP}_l$
2. right components: $\mathbf{N}_r$ and $\mathbf{VP}_r$

- Engenders a family of CLs

NB: A similar technique captures TSQs

The IS-A$_i$s [Thorne & Calvanese 2009]

$$\overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{every}} \overbrace{C_l}^{N_r} \overbrace{C_r}^{VP_l}$$

$$\overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{everybody who}} \overbrace{C_l}^{VP_l} \overbrace{C_r}^{VP_r}$$

The IS-A$_i$s [Thorne & Calvanese 2009]

$$\overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{every}}\ \overbrace{N_r}^{C_l}\ \overbrace{VP_l}^{C_r} \qquad \overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{everybody who}}\ \overbrace{VP_l}^{C_l}\ \overbrace{VP_r}^{C_r}$$

Concept C_f	Constituent α_f	Grammar Rules
A	$\mathbf{N}_f$, $\mathbf{VP}_f$	$\mathbf{VP}_f \longrightarrow$ is a $\mathbf{N}_f \mid \mathbf{IV} \mid$ is $\mathbf{Adj}$ $\mathbf{N}_f \longrightarrow \mathbf{N}$
$\exists r : C_f$	$\mathbf{TV}$ only $\mathbf{N}_f$	$\mathbf{VP}_f \longrightarrow \mathbf{TV}\,\mathbf{NP}_f$ $\mathbf{NP}_f \longrightarrow \mathbf{Det}\,\mathbf{N}_f$
$C_f \sqcap C_f$	$\mathbf{Adj}\ \mathbf{N}_f$	$\mathbf{VP}_f \longrightarrow \mathbf{Adj}\,\mathbf{N}_f$ $\mathbf{N}_f \longrightarrow \mathbf{Adj}\,\mathbf{N}_f$

$\vdots$

The IS-A$_i$s [Thorne & Calvanese 2009]

$$\overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{every}}\ \overbrace{N_r}^{C_l}\ \overbrace{VP_l}^{C_r} \qquad \overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{everybody who}}\ \overbrace{VP_l}^{C_l}\ \overbrace{VP_r}^{C_r}$$

Concept C_f	Constituent α_f	Grammar Rules
A	$\mathbf{N}_f$, $\mathbf{VP}_f$	$\mathbf{VP}_f \longrightarrow$ is a $\mathbf{N}_f \mid \mathbf{IV} \mid$ is $\mathbf{Adj}$ $\mathbf{N}_f \longrightarrow \mathbf{N}$
$\exists r : C_f$	$\mathbf{TV}$ only $\mathbf{N}_f$	$\mathbf{VP}_f \longrightarrow \mathbf{TV}\,\mathbf{NP}_f$ $\mathbf{NP}_f \longrightarrow \mathbf{Det}\,\mathbf{N}_f$
$C_f \sqcap C_f$	$\mathbf{Adj}\ \mathbf{N}_f$	$\mathbf{VP}_f \longrightarrow \mathbf{Adj}\,\mathbf{N}_f$ $\mathbf{N}_f \longrightarrow \mathbf{Adj}\,\mathbf{N}_f$

⋮

CL	Assertions	Sample Sentence(s)
IS-A$_1$	$A \sqsubseteq \forall r{:}A'$	Every project manager manages only projects.
IS-A$_2$	$A \sqcap A' \sqsubseteq \forall r{:}(A'' \sqcap A''')$	Every good manager manages only good projects.

⋮

Data Complexity [Thorne & Calvanese 2011]

	SAT (data)	QA (data)	QA (combined)
IS-A$_1$	in P	NL-complete	in PSPACE
IS-A$_2$	in P	P-complete	in PSPACE
IS-A$_3$	in P	P-complete	in NEXP
IS-A$_4$	in P	P-complete	in PSPACE
IS-A$_5$	in P	coNP-complete	in NEXP
IS-A$_6$	in P	coNP-complete	coNP-complete
IS-A$_7$	in P	coNP-complete	coNP-complete

Data Complexity [Thorne & Calvanese 2011]

	SAT (data)	QA (data)	QA (combined)
IS-A$_1$	in P	NL-complete	in PSPACE
IS-A$_2$	in P	P-complete	in PSPACE
IS-A$_3$	in P	P-complete	in NEXP
IS-A$_4$	in P	P-complete	in PSPACE
IS-A$_5$	in P	coNP-complete	in NEXP
IS-A$_6$	in P	coNP-complete	coNP-complete
IS-A$_7$	in P	coNP-complete	coNP-complete

Data Complexity [Thorne & Calvanese 2011]

	SAT (data)	QA (data)	QA (combined)
IS-A$_1$	in P	NL-complete	in PSPACE
IS-A$_2$	in P	P-complete	in PSPACE
IS-A$_3$	in P	P-complete	in NEXP
IS-A$_4$	in P	P-complete	in PSPACE
IS-A$_5$	in P	coNP-complete	in NEXP
IS-A$_6$	in P	coNP-complete	coNP-complete
IS-A$_7$	in P	coNP-complete	coNP-complete

$\Rightarrow$ Data intractability is caused by "Boolean-closedness"

Data Complexity [Thorne & Calvanese 2011]

	SAT (data)	QA (data)	QA (combined)
IS-A$_1$	in P	NL-complete	in PSPACE
IS-A$_2$	in P	P-complete	in PSPACE
IS-A$_3$	in P	P-complete	in NEXP
IS-A$_4$	in P	P-complete	in PSPACE
IS-A$_5$	in P	coNP-complete	in NEXP
IS-A$_6$	in P	coNP-complete	coNP-complete
IS-A$_7$	in P	coNP-complete	coNP-complete

$\Rightarrow$ Data intractability is caused by "Boolean-closedness"

NB: Maximal tractable CL obtained by eliminating negation!

Semantic Parsers and Aggregations

- Boxer [Bos 2005]: state-of-the art wide-coverage English semantic parser generating logic meaning representations

  Which is the number of students who study something, per country?

$$\forall x_1(\textit{thing}(x_1) \Rightarrow \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6$$
$$(\textit{number}(x_2) \wedge \textit{of}(x_2, x_3) \wedge$$
$$\textit{student}(x_3) \wedge \textit{study}(x_5) \wedge \textit{event}(x_5)$$
$$\wedge \textit{agent}(x_3, x_5) \wedge \textit{patient}(x_6, x_4) \wedge \textit{thing}(x_6)$$
$$\textit{per}(x_2, x_4) \wedge \textit{country}(x_4)) \wedge$$
$$\exists x_7(\textit{event}(x_7) \wedge x_1 = x_2))$$

Semantic Parsers and Aggregations

- Boxer [Bos 2005]: state-of-the art wide-coverage English semantic parser generating logic meaning representations

  Which is the number of students who study something, per country?

$$\forall x_1(\textit{thing}(x_1) \Rightarrow \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6$$
$$(\textit{number}(x_2) \wedge \textit{of}(x_2, x_3) \wedge$$
$$\textit{student}(x_3) \wedge \textit{study}(x_5) \wedge \textit{event}(x_5)$$
$$\wedge \textit{agent}(x_3, x_5) \wedge \textit{patient}(x_6, x_4) \wedge \textit{thing}(x_6)$$
$$\textit{per}(x_2, x_4) \wedge \textit{country}(x_4)) \wedge$$
$$\exists x_7(\textit{event}(x_7) \wedge x_1 = x_2))$$

1. Aggregations modify nominals compound expressing TSQs
2. Aggregations are very common in English

Semantic Parsers and Aggregations

- Boxer [Bos 2005]: state-of-the art wide-coverage English semantic parser generating logic meaning representations

  Which is the number of students who study something, per country?

$$\forall x_1(\textit{thing}(x_1) \Rightarrow \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6$$
$$(\textit{number}(x_2) \wedge \textit{of}(x_2, x_3) \wedge$$
$$\textit{student}(x_3) \wedge \textit{study}(x_5) \wedge \textit{event}(x_5)$$
$$\wedge \textit{agent}(x_3, x_5) \wedge \textit{patient}(x_6, x_4) \wedge \textit{thing}(x_6)$$
$$\textit{per}(x_2, x_4) \wedge \textit{country}(x_4)) \wedge$$
$$\exists x_7(\textit{event}(x_7) \wedge x_1 = x_2))$$

1. Aggregations modify nominals compound expressing TSQs
2. Aggregations are very common in English

QS: Which is the data complexity of CLs with aggregations?

Aggregate Determiners [Thorne & Calvanese 2009]

- Aggregate determiners combine with nominals to produce **NP**s denoting numbers

$$\begin{array}{rcl} \mathbf{Det} & \longrightarrow & \text{the average} \\ \mathbf{Det} & \longrightarrow & \text{the number of} \end{array}$$

$$\begin{array}{rcl} \tau(\mathbf{Det}) & := & \lambda P^{\mathbb{Q}\rightarrow\mathbb{N}}.\mathbf{avg}(P)\colon (e\rightarrow\mathbb{N})\rightarrow\mathbb{Q} \\ \tau(\mathbf{Det}) & := & \lambda P^{e\rightarrow\mathbb{N}}.\mathbf{count}(P)\colon (e\rightarrow\mathbb{N})\rightarrow\mathbb{Q} \end{array}$$

Aggregate Determiners [Thorne & Calvanese 2009]

- Aggregate determiners combine with nominals to produce **NP**s denoting numbers

$$\begin{array}{rcl} \mathbf{Det} & \longrightarrow & \text{the average} \\ \mathbf{Det} & \longrightarrow & \text{the number of} \end{array}$$

$$\begin{array}{rcl} \tau(\mathbf{Det}) & := & \lambda P^{\mathbb{Q}\rightarrow\mathbb{N}}.\mathbf{avg}(P)\colon (e\rightarrow\mathbb{N})\rightarrow\mathbb{Q} \\ \tau(\mathbf{Det}) & := & \lambda P^{e\rightarrow\mathbb{N}}.\mathbf{count}(P)\colon (e\rightarrow\mathbb{N})\rightarrow\mathbb{Q} \end{array}$$

- We use again semantically annotated grammars

$$\begin{array}{rcl} \mathbf{Q}_{wh} \rightarrow \mathbf{I}_i\ \mathbf{S}_{g_i}? & & \lambda\bar{z}.\tau(\mathbf{Intpro}_i)(\lambda i.\tau(\mathbf{S}_{g_i})) \rhd \tau(\mathbf{Q}_{wh}) \\ \mathbf{S}_{g_i} \rightarrow \mathbf{NP}_{g_i}\ \mathbf{VP}_i & & \tau(\mathbf{NP}_{g_i})(\tau(\mathbf{VP}_i)) \rhd \tau(\mathbf{S}_{g_i}) \end{array}$$

Aggregate Determiners [Thorne & Calvanese 2009]

- Aggregate determiners combine with nominals to produce **NP**s denoting numbers

$$\begin{array}{rcl} \mathbf{Det} & \longrightarrow & \text{the average} \\ \mathbf{Det} & \longrightarrow & \text{the number of} \end{array}$$

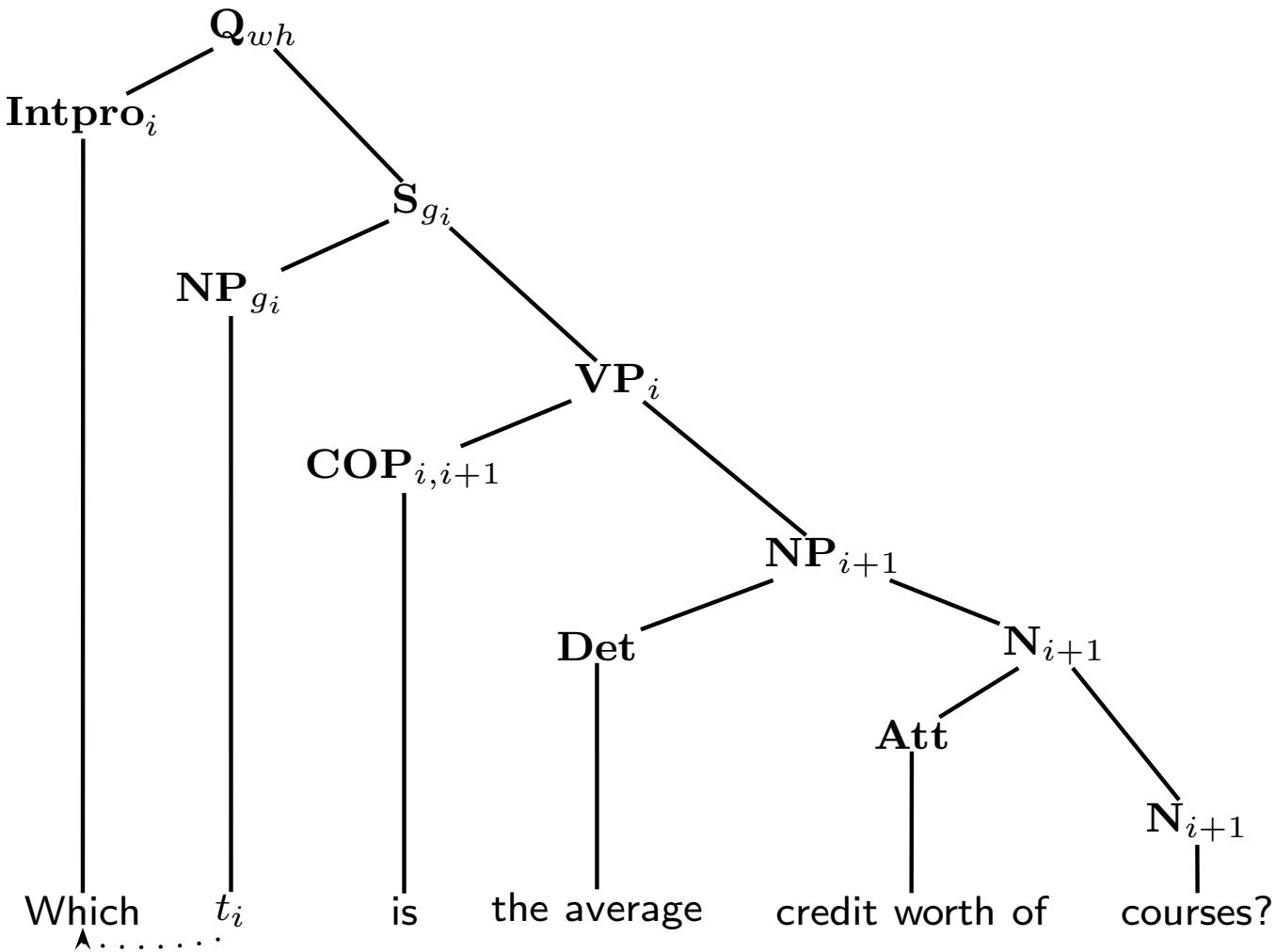
$$\begin{array}{rcl} \tau(\mathbf{Det}) & := & \lambda P^{\mathbb{Q}\rightarrow\mathbb{N}}.\mathbf{avg}(P)\colon (e\rightarrow\mathbb{N})\rightarrow\mathbb{Q} \\ \tau(\mathbf{Det}) & := & \lambda P^{e\rightarrow\mathbb{N}}.\mathbf{count}(P)\colon (e\rightarrow\mathbb{N})\rightarrow\mathbb{Q} \end{array}$$

- We use again semantically annotated grammars

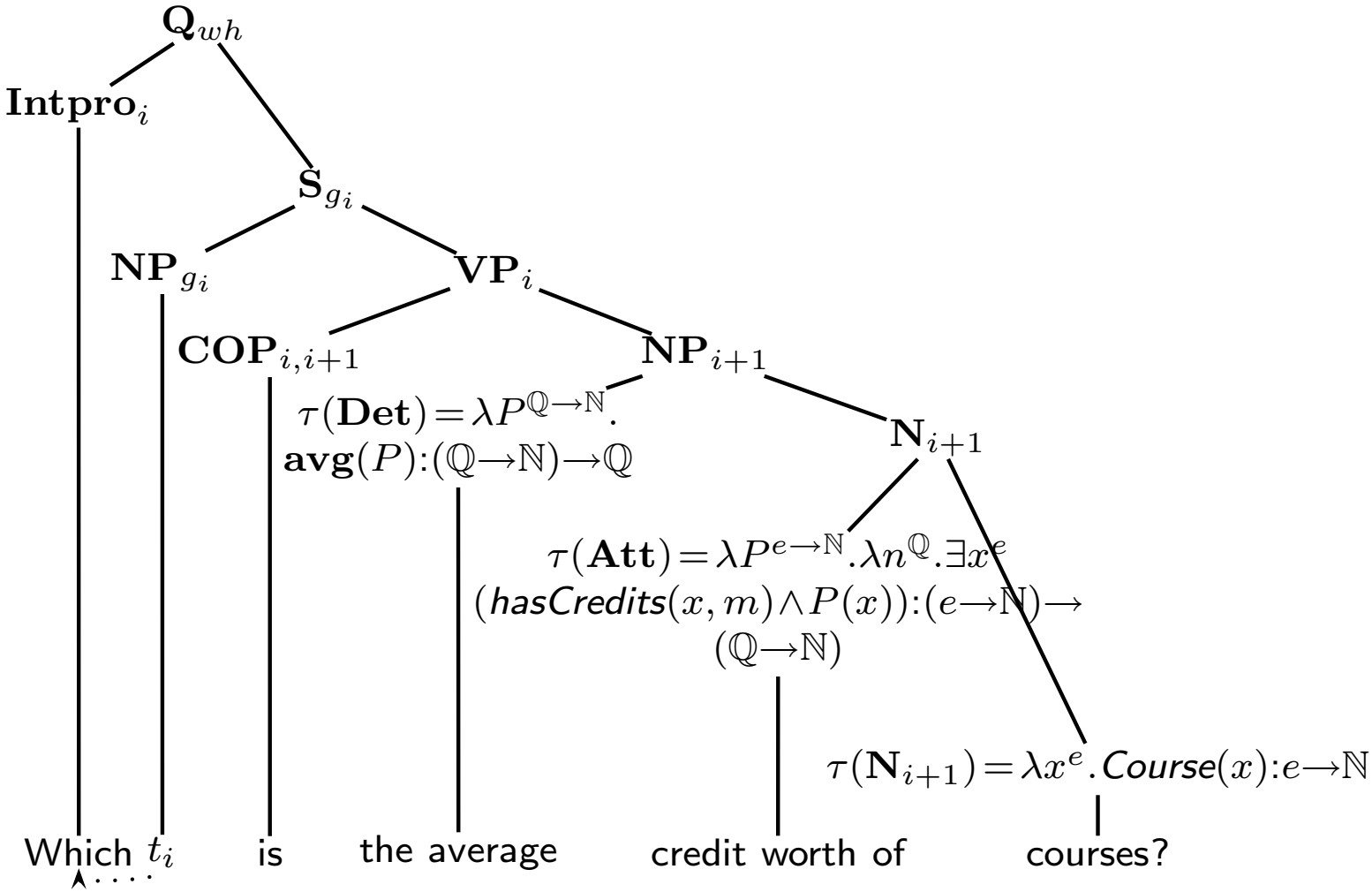
$$\begin{array}{rcl} \mathbf{Q}_{wh} \rightarrow \mathbf{I}_i\ \mathbf{S}_{g_i}? & & \lambda\bar{z}.\tau(\mathbf{Intpro}_i)(\lambda i.\tau(\mathbf{S}_{g_i})) \rhd \tau(\mathbf{Q}_{wh}) \\ \mathbf{S}_{g_i} \rightarrow \mathbf{NP}_{g_i}\ \mathbf{VP}_i & & \tau(\mathbf{NP}_{g_i})(\tau(\mathbf{VP}_i)) \rhd \tau(\mathbf{S}_{g_i}) \end{array}$$

- We consider bag-typed (i.e., $e\rightarrow\mathbb{N}$, $\mathbb{Q}\rightarrow\mathbb{N}$, etc.) constituents

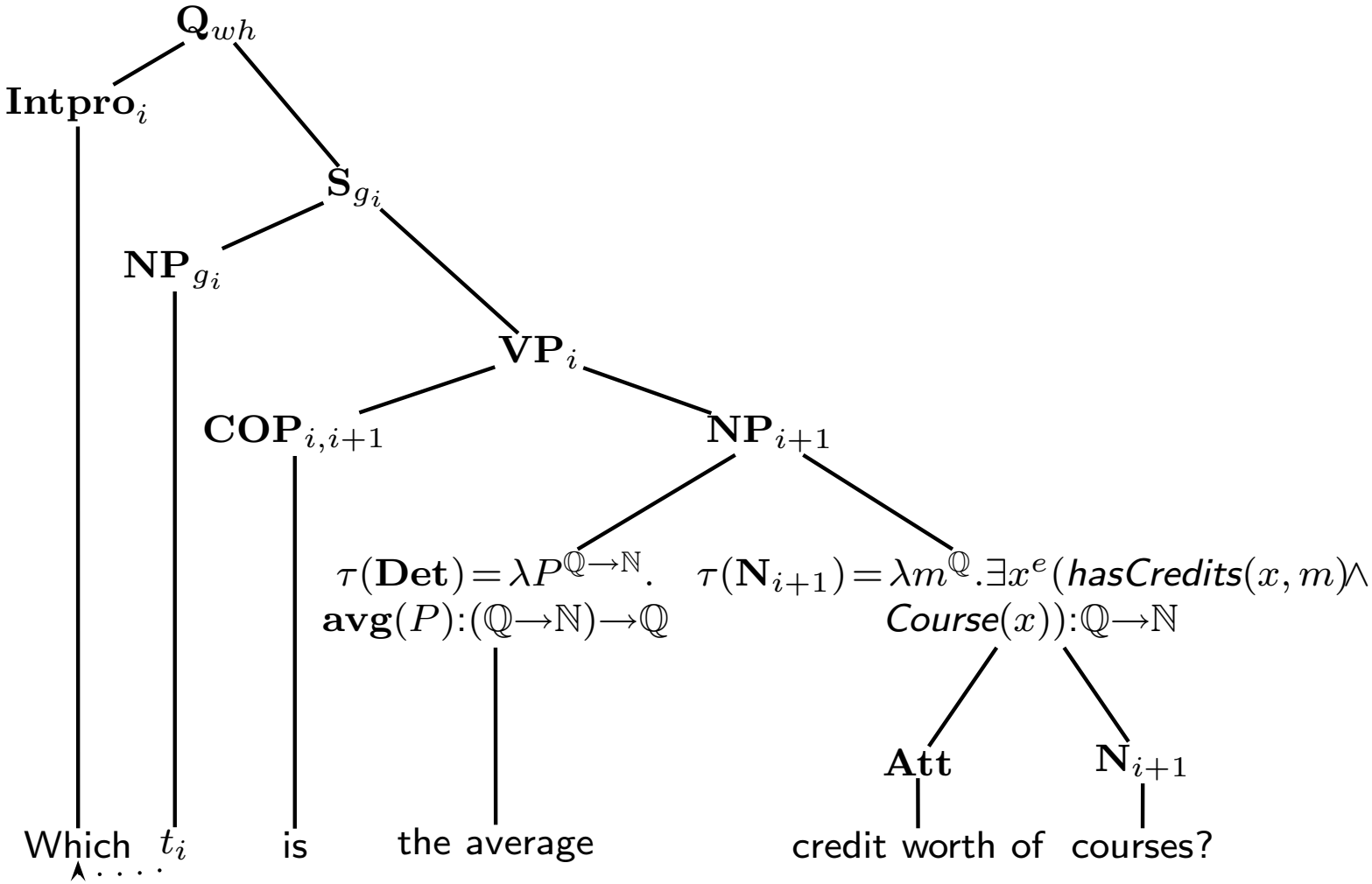
Aggregate Questions: Parsing Example



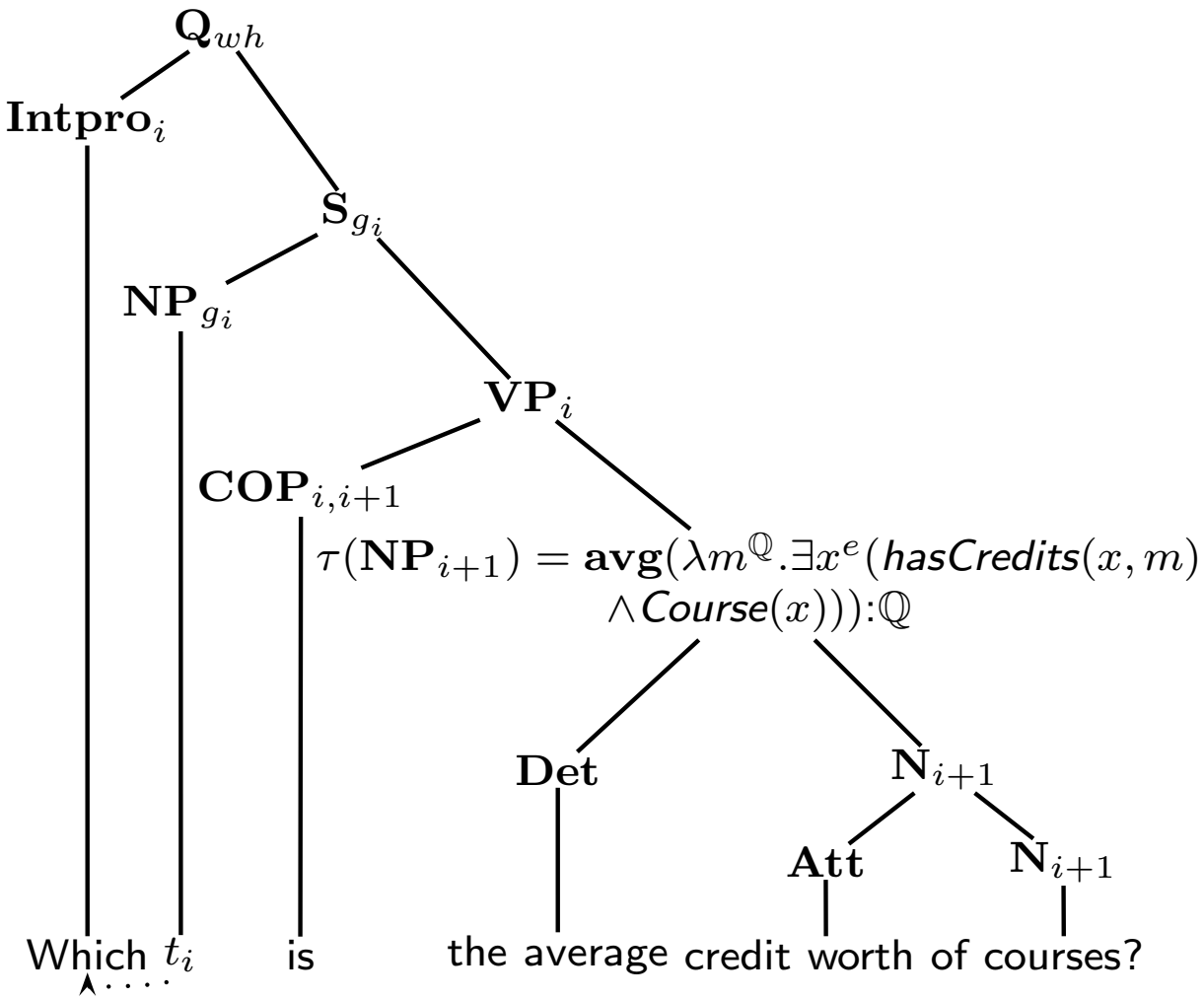
Aggregate Questions: Parsing Example



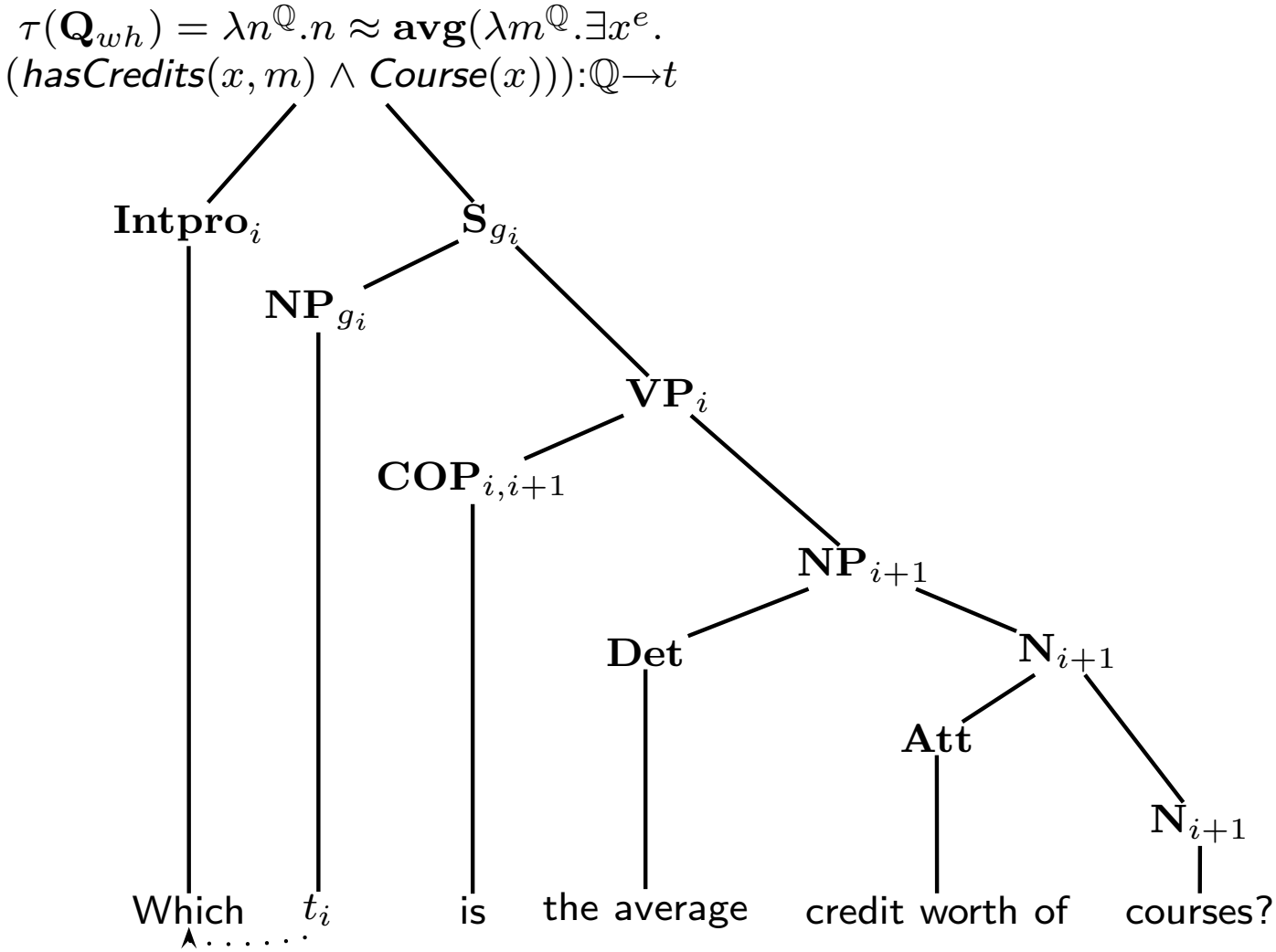
Aggregate Questions: Parsing Example



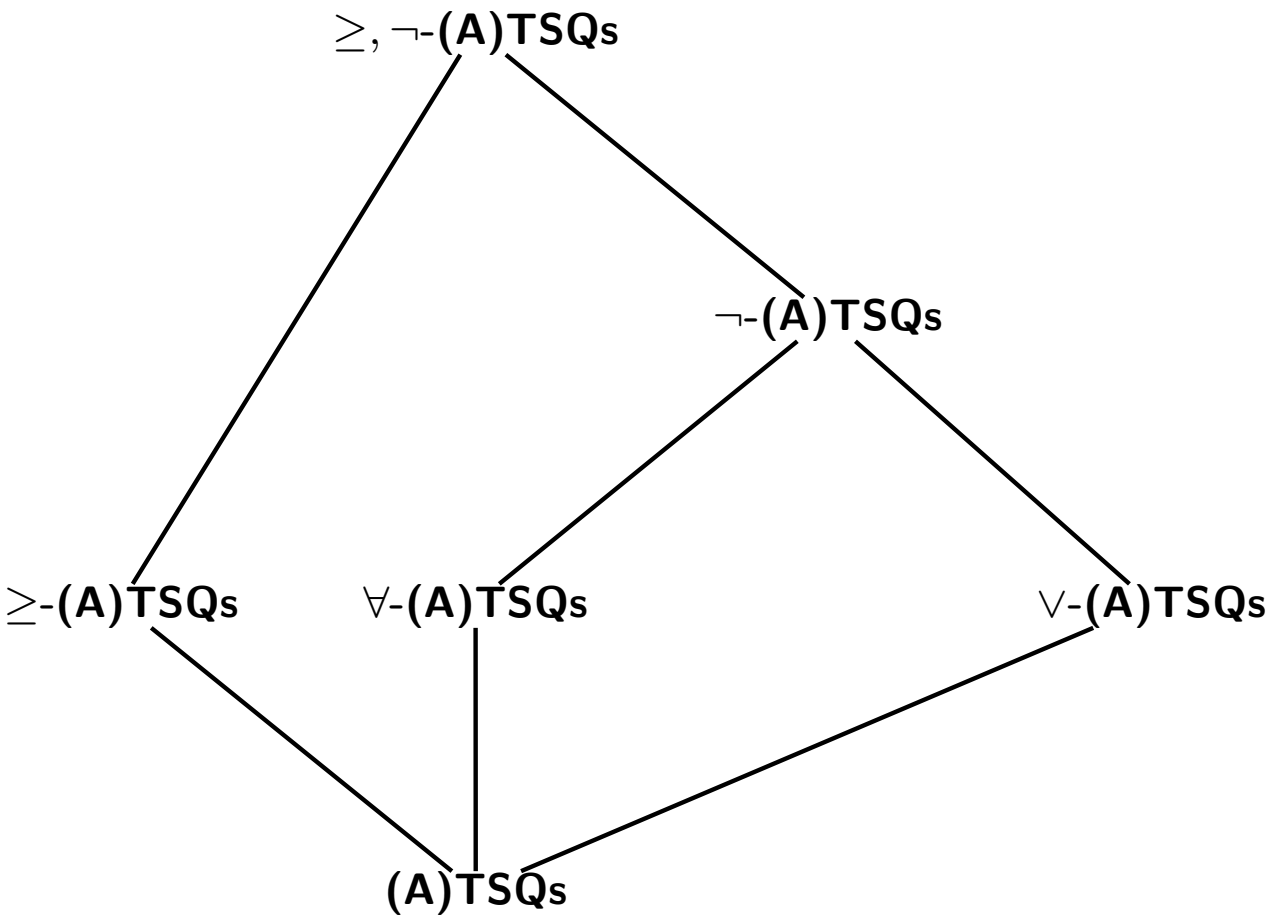
Aggregate Questions: Parsing Example



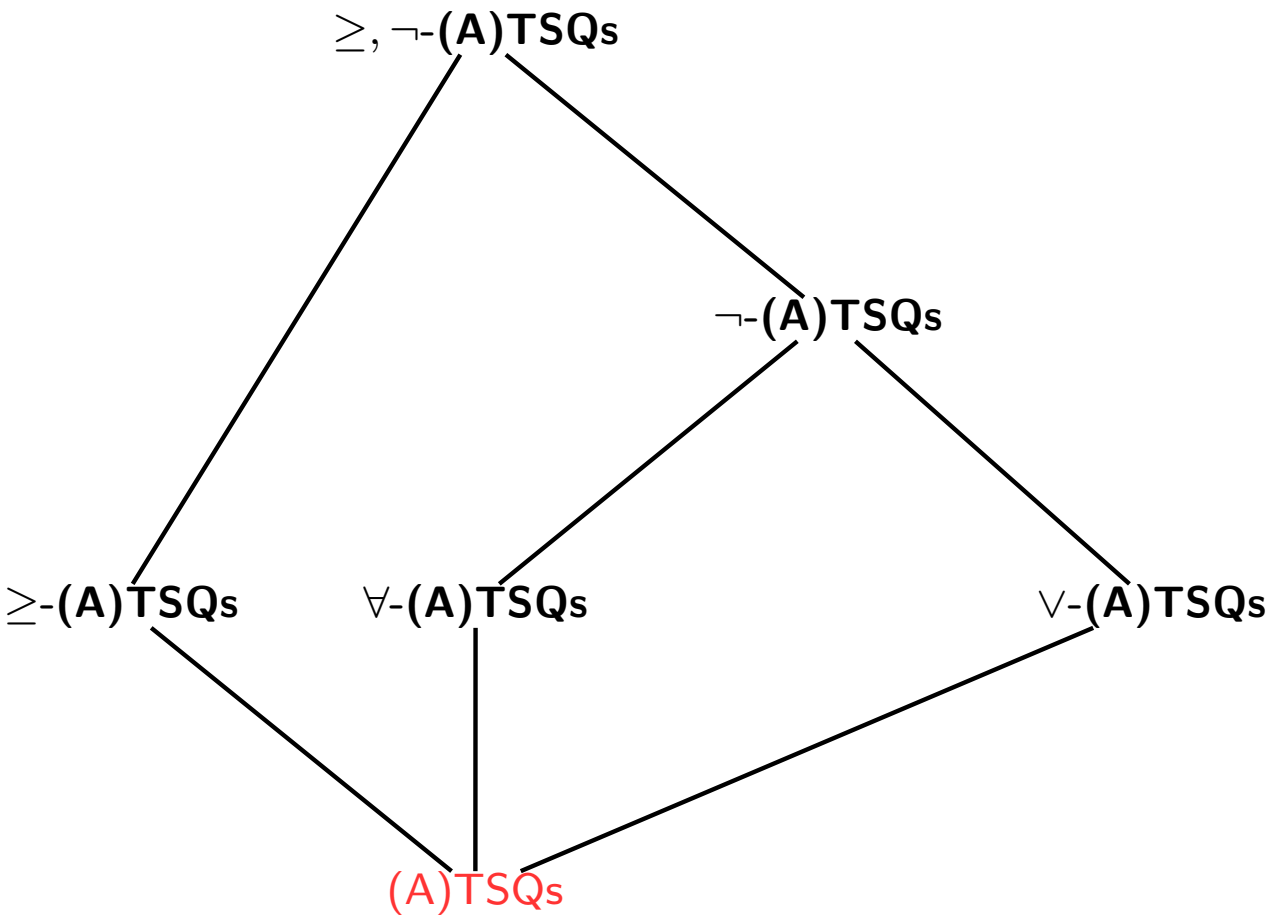
Aggregate Questions: Parsing Example



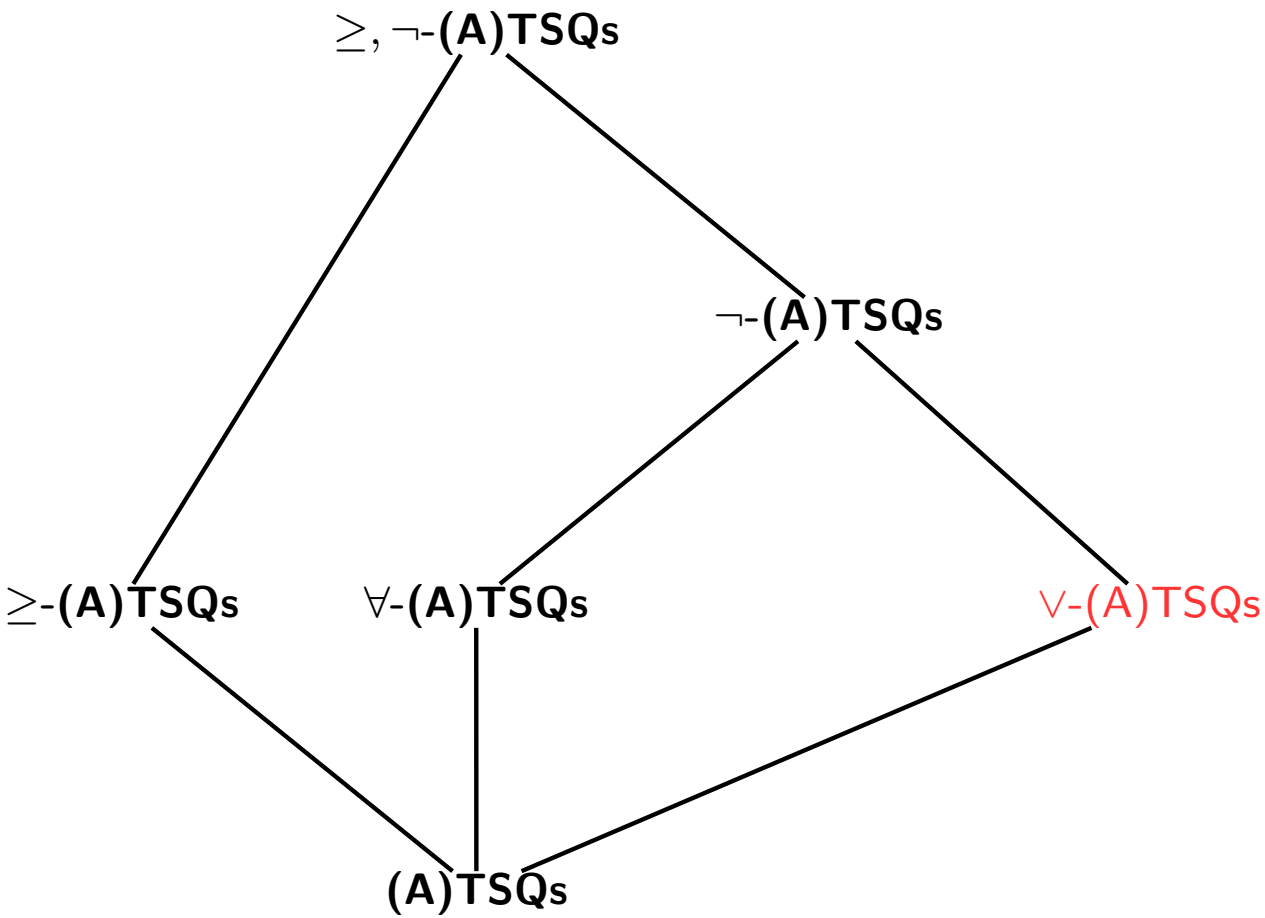
Aggregate Questions and Beyond [Thorne & Calvanese 2009]



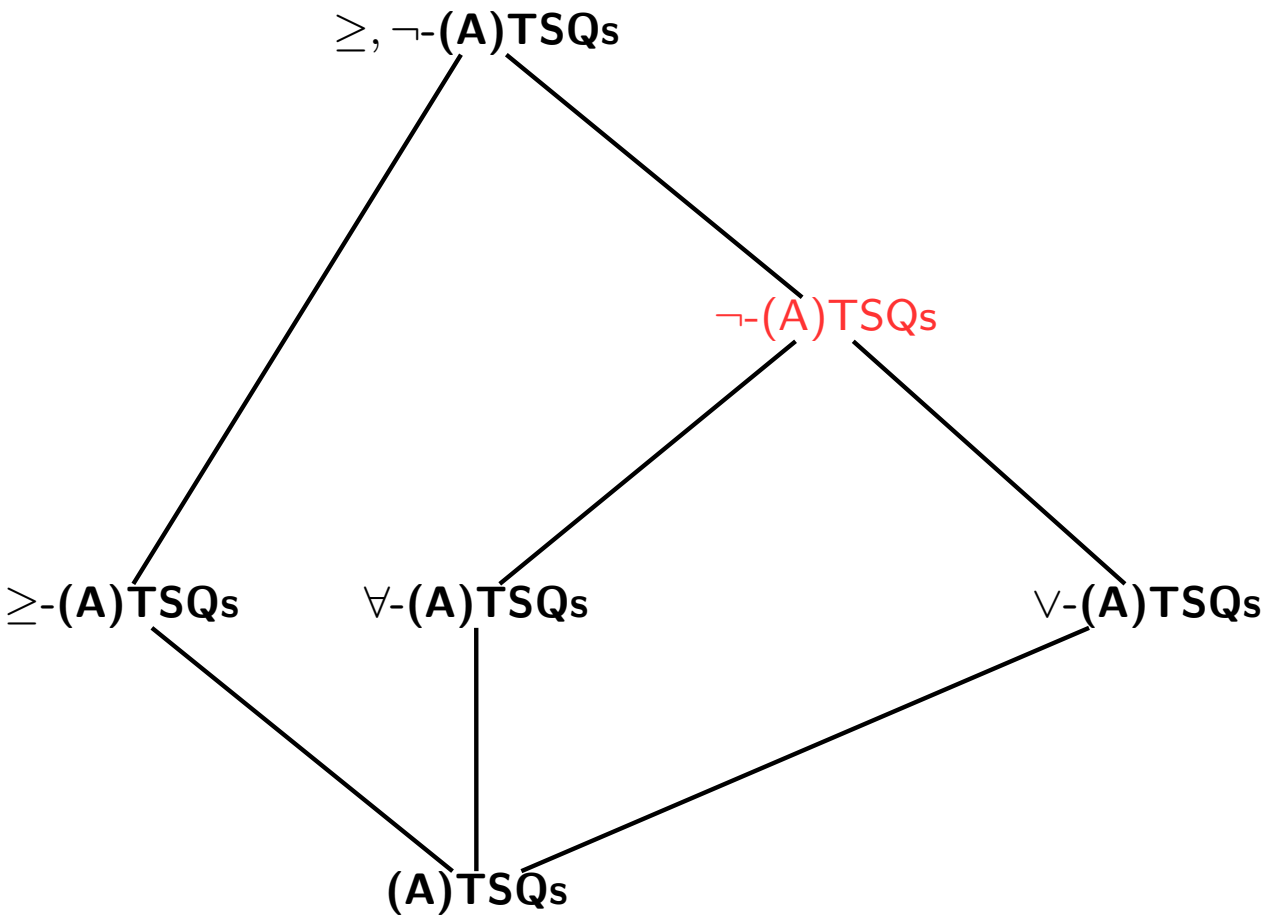
Aggregate Questions and Beyond [Thorne & Calvanese 2009]



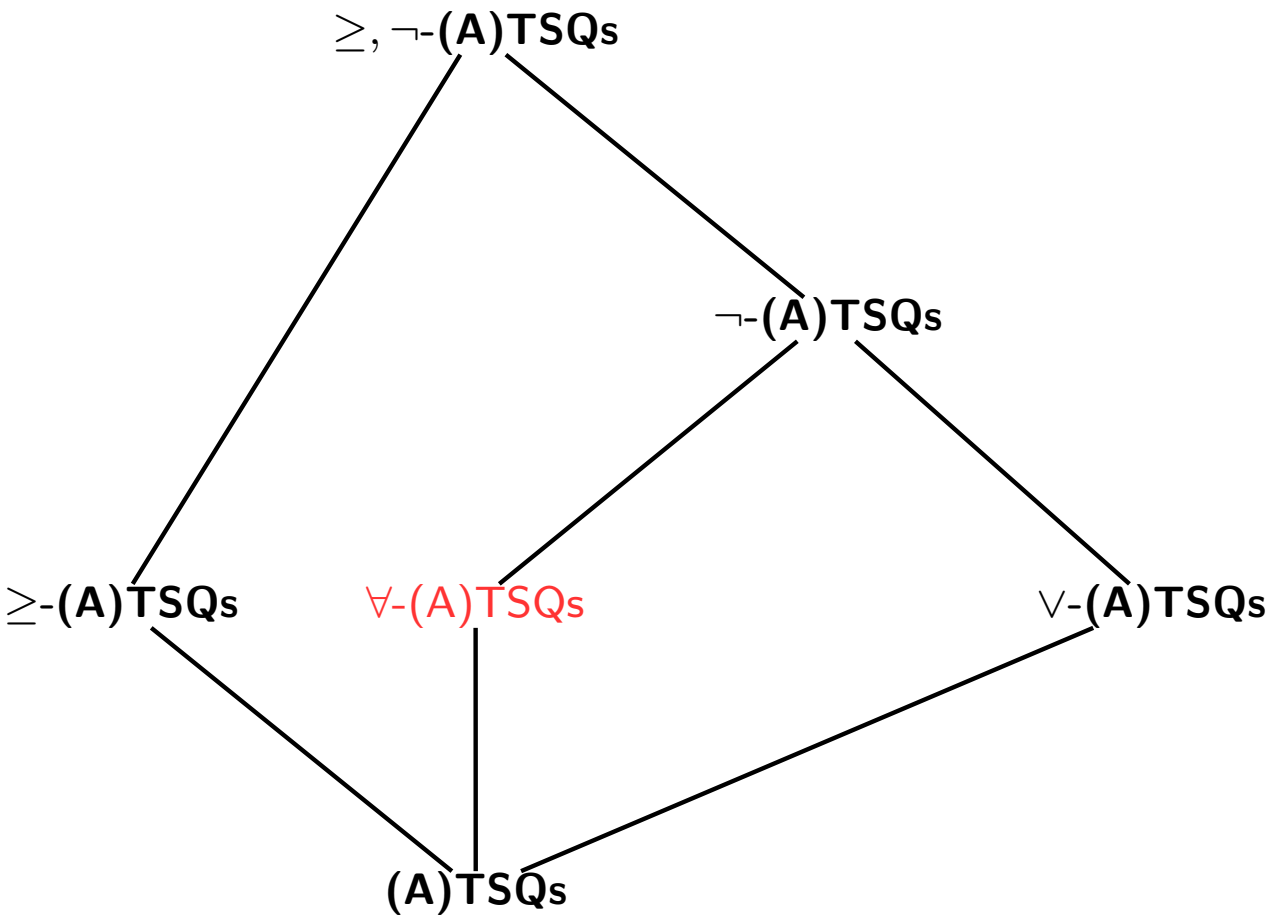
Aggregate Questions and Beyond [Thorne & Calvanese 2009]



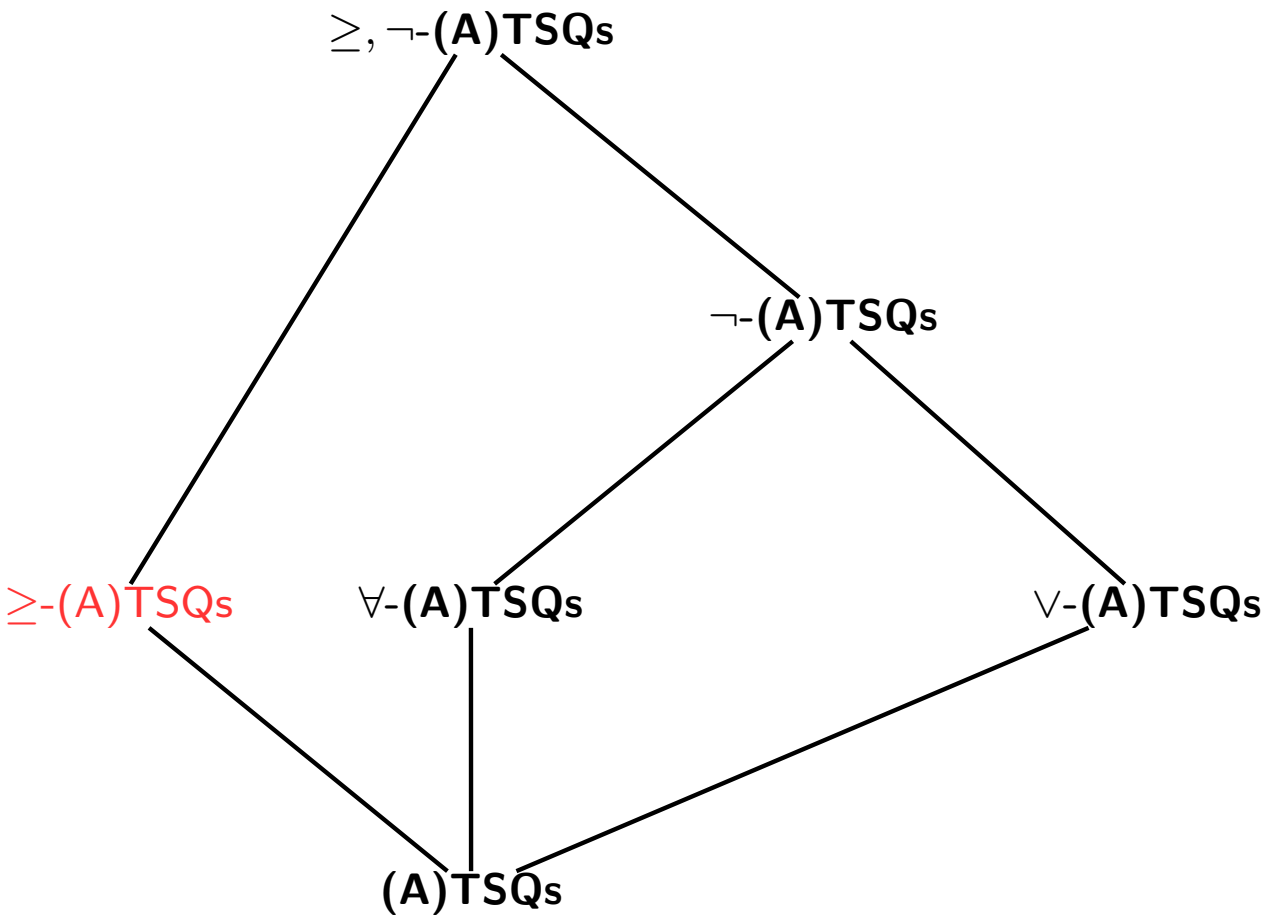
Aggregate Questions and Beyond [Thorne & Calvanese 2009]



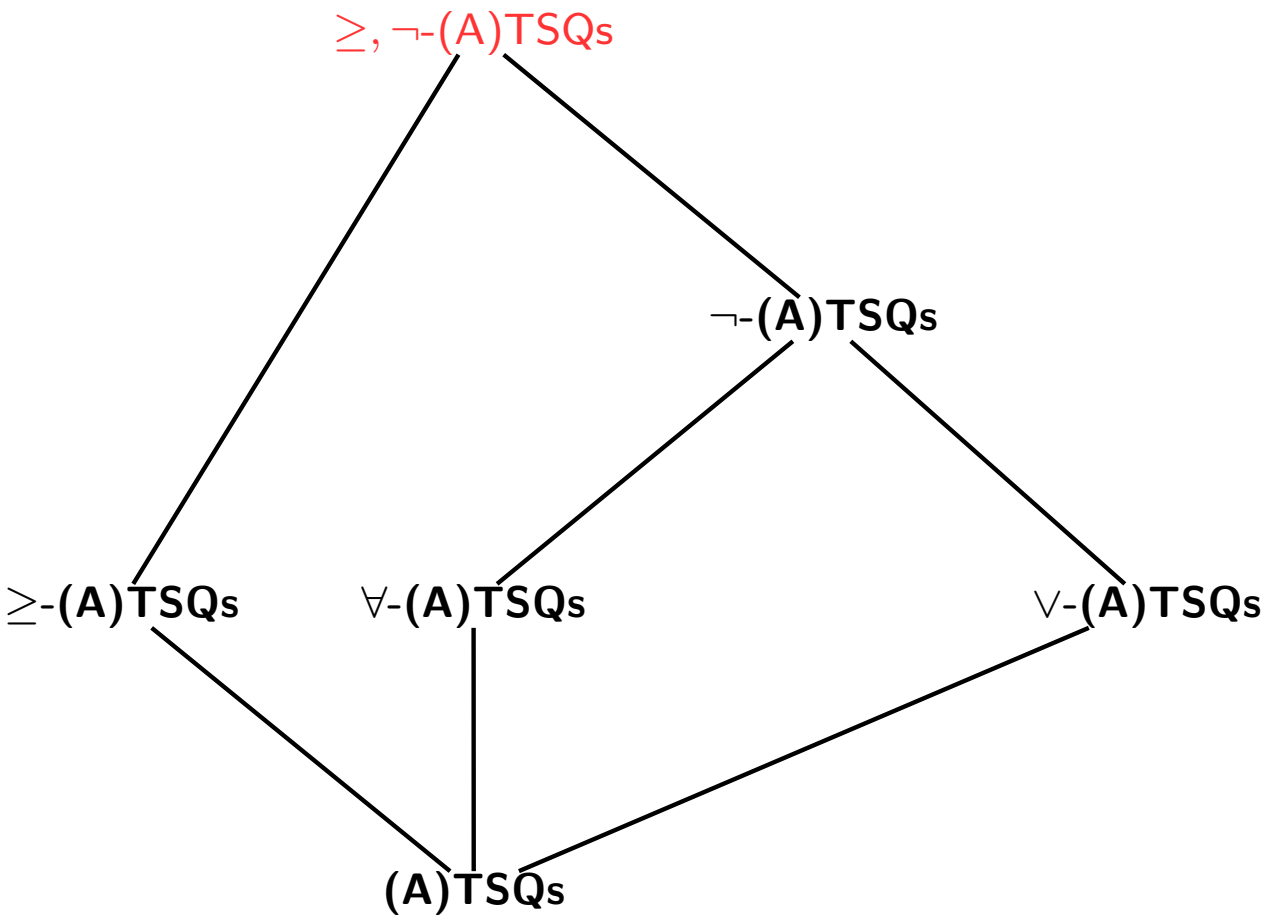
Aggregate Questions and Beyond [Thorne & Calvanese 2009]



Aggregate Questions and Beyond [Thorne & Calvanese 2009]



Aggregate Questions and Beyond [Thorne & Calvanese 2009]



Data Complexity [Thorne & Calvanese 2009]

Data Complexity [Thorne & Calvanese 2009]

- Complexity is tractable (in P) with

1. aggregates over TSQs (space logarithmic overhead)
2. disjunctions of TSQs, $\phi(x) \vee \phi'(x)$

Data Complexity [Thorne & Calvanese 2009]

- Complexity is tractable (in P) with

1. aggregates over TSQs (space logarithmic overhead)
2. disjunctions of TSQs, $\phi(x) \vee \phi'(x)$

- Complexity is intractable (coNP-hard) with

1. negations of TSQs, $\neg\phi(x)$
2. universally restricted TSQs, $\forall x(R(x,y) \rightarrow \phi(y))$
3. comparisons in TSQs, $t\,\theta\,t'$, $\theta \in \{=, >, <, \geq, \leq\}$

- $\geq$, $\neg$-(A)TSQs are in coNP

Data Complexity [Thorne & Calvanese 2009]

- Complexity is tractable (in P) with

1. aggregates over TSQs (space logarithmic overhead)
2. disjunctions of TSQs, $\phi(x) \vee \phi'(x)$

- Complexity is intractable (coNP-hard) with

1. negations of TSQs, $\neg\phi(x)$
2. universally restricted TSQs, $\forall x(R(x,y) \rightarrow \phi(y))$
3. comparisons in TSQs, $t\,\theta\,t'$, $\theta \in \{=, >, <, \geq, \leq\}$

- $\geq$, $\neg$-(A)TSQs are in coNP

NB: We consider queries over ontologies (i.e., OBDA), not relational databases

Conclusions and Further Work

- Studied controlled languages and semantic complexity
- Argued that semantic complexity is an important issue
- Eliminating ambiguity makes translation efficient, but reasoning becomes intractable with simple fragments
- Pinpoited (data) tractable and intractable fragments

Conclusions and Further Work

- Studied controlled languages and semantic complexity
- Argued that semantic complexity is an important issue
- Eliminating ambiguity makes translation efficient, but reasoning becomes intractable with simple fragments
- Pinpoited (data) tractable and intractable fragments
- Further work: how can theoretical semantic complexity be
  1. empirically validated by behavior (processing time) of systems?
  2. empirically validated by behavior of automated reasoners?

Any Questions?

THANK YOU!

Some References

- DLs and OBDA:
  - Diego Calvanese et al., "Tractable reasoning and efficient query answering in description logics: the *DL-Lite* family", 2007, JAIR
  - Diego Calvanese et al., "The *DL-Lite* Family and Relations", 2009, JAIR
- CLs and semantic complexity:
  - Norbert Fuchs et al., "Controlled English for reasoning on the semantic web", 2009, Semantic Techniques for the Web: the REWERSE Perspective
  - Ian Pratt and Allan Third, "More Fragments of Language", 2006, Notre Dame Journal of Formal Logic
- Data complexity of CLs:
  - Camilo Thorne, "Querying the Fragments of English", 2011, Proceedings of WoLLIC 2011
  - Camilo Thorne, Diego Calvanese, "Tractability and Intractability of Controlled Languages for Data Acess", 2011, Studia Logica
  - Camilo Thorne, Diego Calvanese, "The Data Complexity of the Syllogistic Fragments of English", Proceedings of AC 2009
  - Camilo Thorne, Diego Calvanese, "Tree-shaped aggregate questions over ontologies", 2009, Proceedings of FQAS 2009
  - Raffaella Bernardi et al., "Lite natural language", 2007, Proceedings of IWCS-7

(Data) Scalable and Non-Scalable Constructs [Thorne 2010]

	Declarations	Questions
Constructs that scale (P or less)	– Negation in predicate **VP**s, relatives in predicate **VP**s, conjunction in predicate **VP**s – Relatives and conjunction in subject **NP**s and predicate **VP**s, but no negation	– Existential quantifiers, conjunction, relatives, disjunctions, aggregations
Constructs that do not (coNP-hard)	– Negation in subject **NP**s – Relatives and negation in subject **NP**s and predicate **VP**s	– Full negation – Comparisons – Universal restrictions
Undecidable Constructs	– **TV**s, relatives, negation, existential and universal quantifiers, restricted anaphoric pronouns and indeterminate pronouns in subject **NP**s and predicate **VP**s, copula	– **TV**s, existential indeterminate pronouns, relatives and restricted anaphoric pronouns

The IS-A_is [Thorne & Calvanese 2009]

Concept C_f	Constituent α_f	Grammar Rules
A	$\mathbf{N}_f$, $\mathbf{VP}_f$	
$\exists P: A$	**TV** some $\mathbf{N}_f$, **TV** somebody who $\mathbf{VP}_f$	$\mathbf{VP}_f \longrightarrow$ is a $\mathbf{N}_f \mid \mathbf{IV} \mid$ is **Adj**
$\exists P^- : A$	**TV** by some $\mathbf{N}_f$, **TV** by somebody who $\mathbf{VP}_f$	$\mathbf{N}_f \longrightarrow \mathbf{N}$
$\forall P: A$	**TV** only $\mathbf{VP}_f$, **TV** only who $\mathbf{VP}_f$	
$\exists P$	**TV** something, **TV** somebody	$\emptyset$
$A_1 \sqcap \cdots \sqcap A_n$	**Adj** $\mathbf{N}_f$, $\mathbf{N}_f$ who $\mathbf{VP}_f$	$\mathbf{VP_f} \longrightarrow$ is a $\mathbf{N}_f \mid \mathbf{IV} \mid$ is **Adj** $\mid \mathbf{VP_f}$ and $\mathbf{VP}_f$
	$\mathbf{N}_f$ and $\mathbf{N}_f$, $\mathbf{VP}_f$ and $\mathbf{VP}_f$	$\mathbf{N}_f \longrightarrow \mathbf{N} \mid$ **Adj** $\mathbf{N}_f$ $\mid \mathbf{N}_f$ and $\mathbf{N}_f$
$A_1 \sqcup \cdots \sqcup A_n$	$\mathbf{VP}_f$ or $\mathbf{VP}_f$	$\mathbf{VP}_f \longrightarrow$ is a $\mathbf{N}_f \mid \mathbf{IV} \mid$ is **Adj** $\mid \mathbf{VP}_f$ and $\mathbf{VP}_f$ $\mathbf{N}_f \longrightarrow \mathbf{N} \mid \mathbf{N}_f$ and $\mathbf{N}_f$
$\neg A$	is not **Adj**, does not **IV**, is not a $\mathbf{N}_f$	$\mathbf{N}_f \longrightarrow \mathbf{N}$

The IS-A_is [Thorne & Calvanese 2009]

CL	Assertions	Sample Sentence(s)
IS-A_1	$A \sqsubseteq \forall r{:}A'$	Every project manager manages only projects.
IS-A_2	$A \sqcap A' \sqsubseteq \forall r{:}(A'' \sqcap A''')$	Every good manager manages only good projects.
IS-A_3	$\exists r{:}A \sqsubseteq A' \sqcap A''$	Anybody who manages some project is an employee and is a manager.
	$\exists r^{-}{:}A \sqsubseteq A' \sqcap A''$	Anything that is managed by some important manager is a big project.
	$A \sqsubseteq \exists P$	Every manager manages something.
IS-A_4	$A \sqcap A' \sqsubseteq A'' \sqcap A'''$	Every cruel manager is a bad manager.
	$\exists r{:}(A \sqcap A') \sqsubseteq A'' \sqcap A'''$	Anybody who manages some bankrupt project is a bad manager.
IS-A_5	$\forall r{:}A \sqsubseteq A' \sqcap A''$	Anybody who manages only projects is a manager and a project manager.
IS-A_6	$A \sqsubseteq A' \sqcup A''$	Every manager is a project manager or is an area manager.
IS-A_7	$\neg A \sqsubseteq A' \sqcap A''$	Anybody who is not an area manager is an employee who is a project manager.

Fo Semantics of $\mathcal{ALCI}$ [Baader et al. 2004]

Consider Fo interpretations $\mathcal{I} := (\Delta, \cdot^{\mathcal{I}})$:

$$
\begin{aligned}
A^{\mathcal{I}} &\subseteq \Delta \\
(\exists R{:}C_f)^{\mathcal{I}} &:= \{d \mid \text{exists } d' \text{ s.t. } (d,d') \in R^{\mathcal{I}} \text{ and } d' \in C_f^{\mathcal{I}}\} \\
(\neg C_f)^{\mathcal{I}} &:= \Delta \setminus {C_f}^{\mathcal{I}} \\
(C_f \sqcap C'_f)^{\mathcal{I}} &:= {C_f}^{\mathcal{I}} \cap {C'_f}^{\mathcal{I}}
\end{aligned}
$$

$$
\begin{aligned}
r^{\mathcal{I}} &\subseteq \Delta \times \Delta \\
(r^-)^{\mathcal{I}} &:= \{(d,d') \mid (d',d) \in r^{\mathcal{I}}\}
\end{aligned}
$$

$\mathcal{I} \models C_l \sqsubseteq C_r$ iff $C_l^{\mathcal{I}} \subseteq C_r^{\mathcal{I}}$

$\mathcal{I} \models \mathcal{O}$ iff for all $C_l \sqsubseteq C_r \in \mathcal{O}$, $\mathcal{I} \models C_l^{\mathcal{I}} \subseteq C_r^{\mathcal{I}}$

Open world semantics: $\mathcal{O}$ may have many models!

Research Methodology [Thorne 2010]

We proposed the following strategy:

1. Define a declarative CL L_D and a grammar for it
2. Define a meaning representation $\tau(L_D)$
3. Define an interrogative CL L_Q and a grammar for it
4. Define a meaning representation $\tau(L_Q)$
5. Study the properties of
    1. the ontology language $\tau(L_D)$
    2. the query language $\tau(L_Q)$

$\Rightarrow$ infer data complexity by reasoning on $\tau(L_D)$ and $\tau(L_Q)$