Query Answering over Ontologies Using Controlled Languages

Camilo Thorne

Supervisor: D. Calvanese Co-Supervisor: R. Bernardi

KRDB Research Centre – Free University of Bozen-Bolzano

PhD Final Examination - Cycle XXI

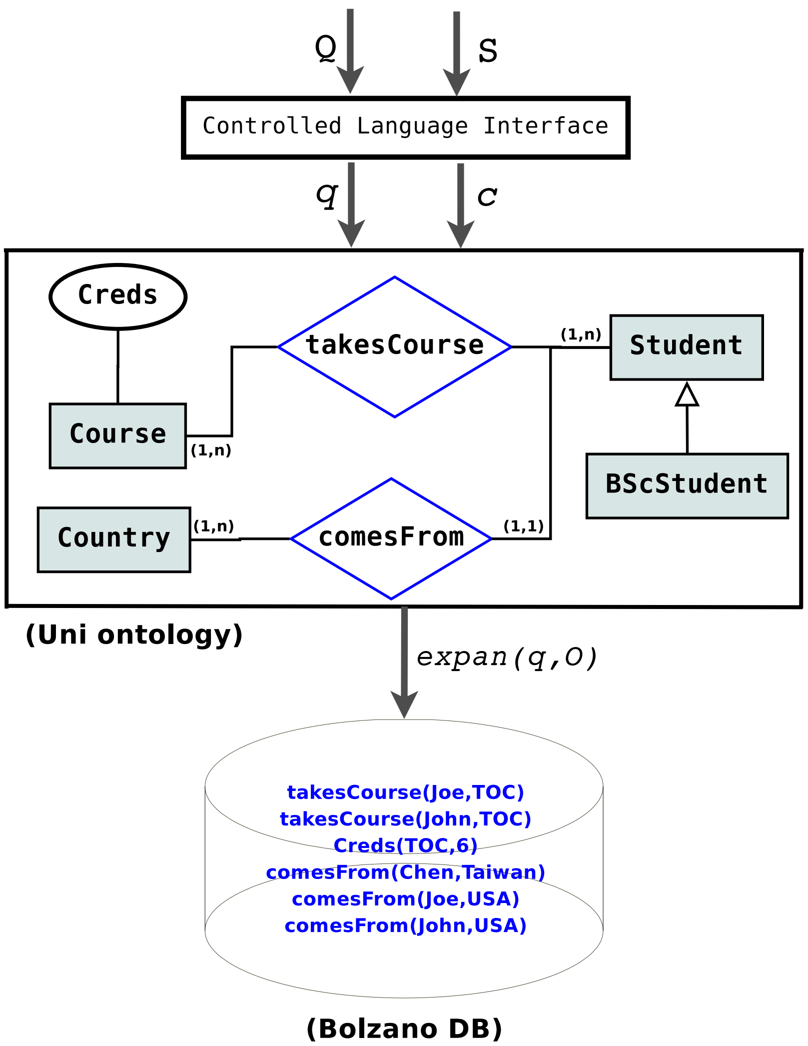
Outline

1. Introduction
   (a) Controlled Languages
   (b) Ontology Data Access Systems and Scalability
   (c) Semantic Complexity
2. Lite-English and GCQ-English
3. Expressing Aggregations
4. The $\{\text{IS-A}_i\}_{i \in [0,7]}$ fragments
5. Complexity of the Fragments of English
6. Conclusions

Controlled Language Interfaces

- Proposed to address the usability problem of information systems [Sowa 2004, Fuchs et al. 2006]
- A controlled language (CL) is an ambiguity-free subset of English which
  - translates into logic meaning representations
  - the translation can be modelled by compositional translations $\tau(\cdot)$
- More recently proposed for interfaces to ontology-based data access systems [Bernstein et al. 2005, Schwitter2006]
- Being ambiguity-free entails good processing, but less is known about other combinatorial properties

Ontology Based Data Access Systems



An ontology based data access system (OBDAS) is a pair $(\mathcal{O}, \mathcal{D})$ where data

– is stored in a databases $\mathcal{D}$

– is queried through an ontology $\mathcal{O}$

Certain answers semantics:

(i) queries are expanded w.r.t. $\mathcal{O}$

(ii) expanded queries are evaluated on $\mathcal{D}$

(iii) answer tuples are returned

$\Rightarrow$ Add a CLI front-end?

Ontology and Query Languages

- Interested in declaratative CLs that translate into fragments of OWL
- $\mathcal{ALCI}$ ontologies structure the domain in terms of concepts and roles built from concept names A and role names r

$$R \longrightarrow r \mid r^{-} \qquad C \longrightarrow A \mid \neg C \mid \exists R{:}C \mid C \sqcap C'$$

- Ontologies $\mathcal{O}$ are sets of concept inclusion assertions $C \sqsubseteq C'$ of semantics $\forall x(C(x) \Rightarrow C'(x))$
- Interested in interrogative CLs that translate into
  - SQL `SELECT-PROJECT-JOIN` queries, a.k.a. conjunctive queries (CQs)
  - SQL aggregate queries
- We assume the standard certain answers semantics for answering CQs over OBDASs

A Survey of CLs

CL (English)	Maps to	Goal
ACE [Fuchs 2005]	FO	KR/User specifications
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]	no	User specifications
English Query (EQ) [Blum 1999]	SQL	DB querying/management
OWL-CNL [Schwitter 2006]	OWL-DL	Ontology authoring
Easy English [Bernth 1998]	no	User specifications
λ-SQL [Winter 2006]	SQL	DB 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	DB querying

Scalability of CLs

- An important measure of performance/scalability of OBDASs is data complexity [Vardi 1982]
- Modulo compositional translations $\tau(\cdot)$ CL constructs have an impact on data complexity $\Rightarrow$ semantic complexity [Pratt&Third 2005]

Scalability of CLs

- An important measure of performance/scalability of OBDASs is data complexity [Vardi 1982]
- Modulo compositional translations $\tau(\cdot)$ CL constructs have an impact on data complexity $\Rightarrow$ semantic complexity [Pratt&Third 2005]
- Study the computational properties of the CLs
  - which combinations of CL constructs give way to <u>tractable</u> data complexity?
  - which combinations of CL constructs give rise to <u>intractable</u> data complexity?

Research Methodology

We propose 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
   (a) the ontology language $\tau(L_D)$
   (b) the query language $\tau(L_Q)$

$\Rightarrow$ infer the data complexity of the CLs by reasoning the ontology and query languages

Compositional Translations, Grammars and CLs

Syntax Rules	$\tau(\cdot)$
S $\longrightarrow$ NP VP	$\tau(\text{NP})(\tau(\text{VP})) \rhd \tau(\text{S})$
VP $\longrightarrow$ is a N	$\tau(\text{VP}) = \tau(\text{N})$
NP $\longrightarrow$ PN	$\tau(\text{NP}) = \tau(\text{PN})$
NP $\longrightarrow$ Det N	$\tau(\text{Det})(\tau(\text{N})) \rhd \tau(\text{NP})$

Content Lexicon	$\tau(\cdot)$
N $\longrightarrow$ woman	$\tau(\text{N}) = \lambda x^e.Woman(x){:}e{\to}t$
PN $\longrightarrow$ Mary	$\tau(\text{PN}) = \lambda P^{e\to t}.P(Mary){:}(e{\to}t){\to}t$

Function Lexicon	$\tau(\cdot)$
Det $\longrightarrow$ every	$\tau(\text{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\forall x^e(P(x) \Rightarrow Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$
Det $\longrightarrow$ some	$\tau(\text{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\exists x^e(P(x) \wedge Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$

Compositional Translations, Grammars and CLs

Syntax Rules	$\tau(\cdot)$
S $\longrightarrow$ NP VP	$\tau(\text{NP})(\tau(\text{VP})) \rhd \tau(\text{S})$
VP $\longrightarrow$ is a N	$\tau(\text{VP}) = \tau(\text{N})$
NP $\longrightarrow$ PN	$\tau(\text{NP}) = \tau(\text{PN})$
NP $\longrightarrow$ Det N	$\tau(\text{Det})(\tau(\text{N})) \rhd \tau(\text{NP})$

Content Lexicon	$\tau(\cdot)$
N $\longrightarrow$ woman	$\tau(\text{N}) = \lambda x^e.Woman(x){:}e{\to}t$
PN $\longrightarrow$ Mary	$\tau(\text{PN}) = \lambda P^{e\to t}.P(Mary){:}(e{\to}t){\to}t$

Function Lexicon	$\tau(\cdot)$
Det $\longrightarrow$ every	$\tau(\text{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\forall x^e(P(x) \Rightarrow Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$
Det $\longrightarrow$ some	$\tau(\text{Det}) = \lambda P^{e\to t}.\lambda Q^{e\to t}.\exists x^e(P(x) \wedge Q(x)){:}(e{\to}t){\to}((e{\to}t){\to}t)$

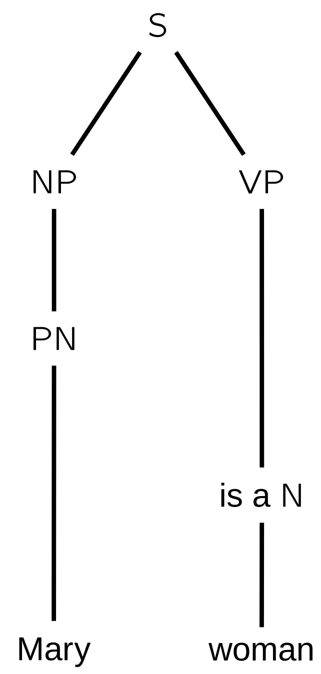
Compositional Translations, Grammars and CLs

Syntax Rules	$\tau(\cdot)$
S $\longrightarrow$ NP VP	$\tau(\text{NP})(\tau(\text{VP})) \triangleright \tau(\text{S})$
VP $\longrightarrow$ is a N	$\tau(\text{VP}) = \tau(\text{N})$
NP $\longrightarrow$ PN	$\tau(\text{NP}) = \tau(\text{PN})$
NP $\longrightarrow$ Det N	$\tau(\text{Det})(\tau(\text{N})) \triangleright \tau(\text{NP})$

Content Lexicon	$\tau(\cdot)$
N $\longrightarrow$ woman	$\tau(\text{N}) = \lambda x^e.Woman(x){:}e{\rightarrow}t$
PN $\longrightarrow$ Mary	$\tau(\text{PN}) = \lambda P^{e\rightarrow t}.P(Mary){:}(e{\rightarrow}t){\rightarrow}t$

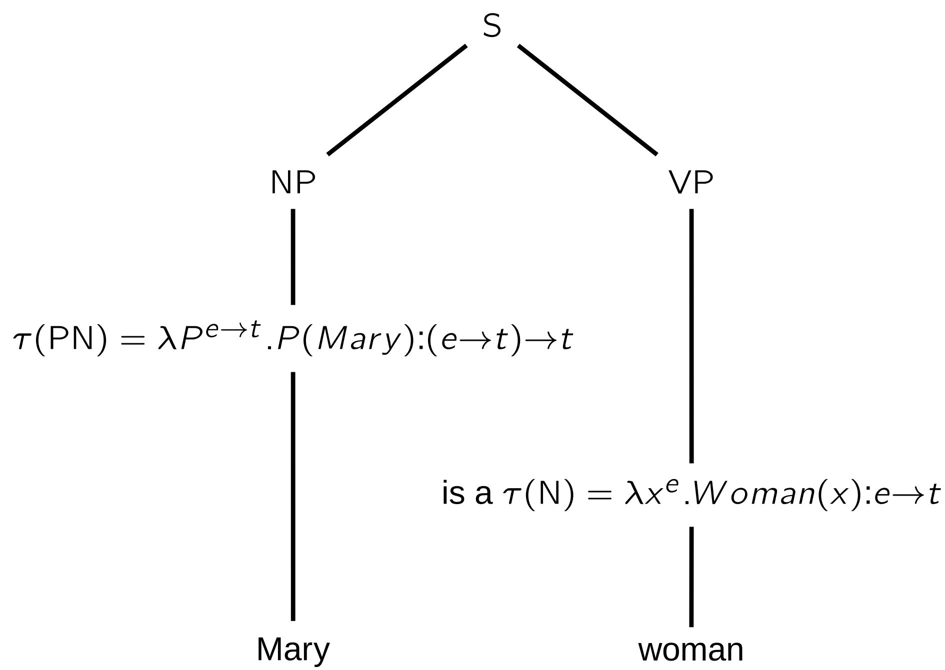
Function Lexicon	$\tau(\cdot)$
Det $\longrightarrow$ every	$\tau(\text{Det}) = \lambda P^{e\rightarrow t}.\lambda Q^{e\rightarrow t}.\forall x^e(P(x) \Rightarrow Q(x)){:}(e{\rightarrow}t){\rightarrow}((e{\rightarrow}t){\rightarrow}t)$
Det $\longrightarrow$ some	$\tau(\text{Det}) = \lambda P^{e\rightarrow t}.\lambda Q^{e\rightarrow t}.\exists x^e(P(x) \wedge Q(x)){:}(e{\rightarrow}t){\rightarrow}((e{\rightarrow}t){\rightarrow}t)$

Compositional Translations, Grammars and CLs



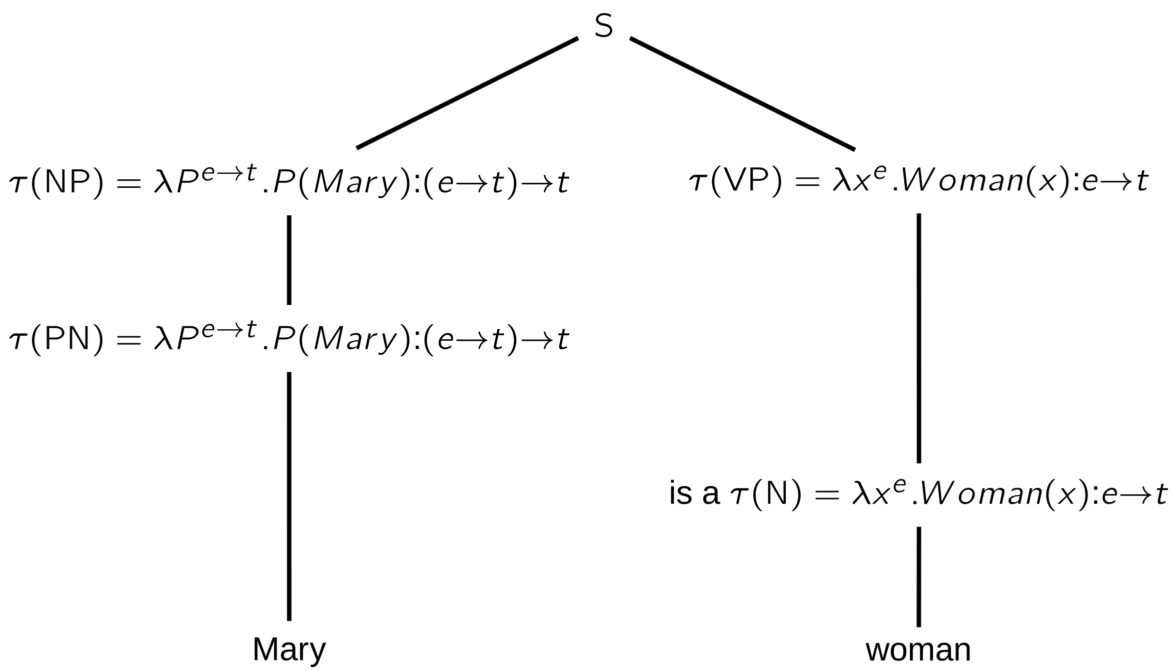
Parsing and interpreting "Mary is a woman"

Compositional Translations, Grammars and CLs



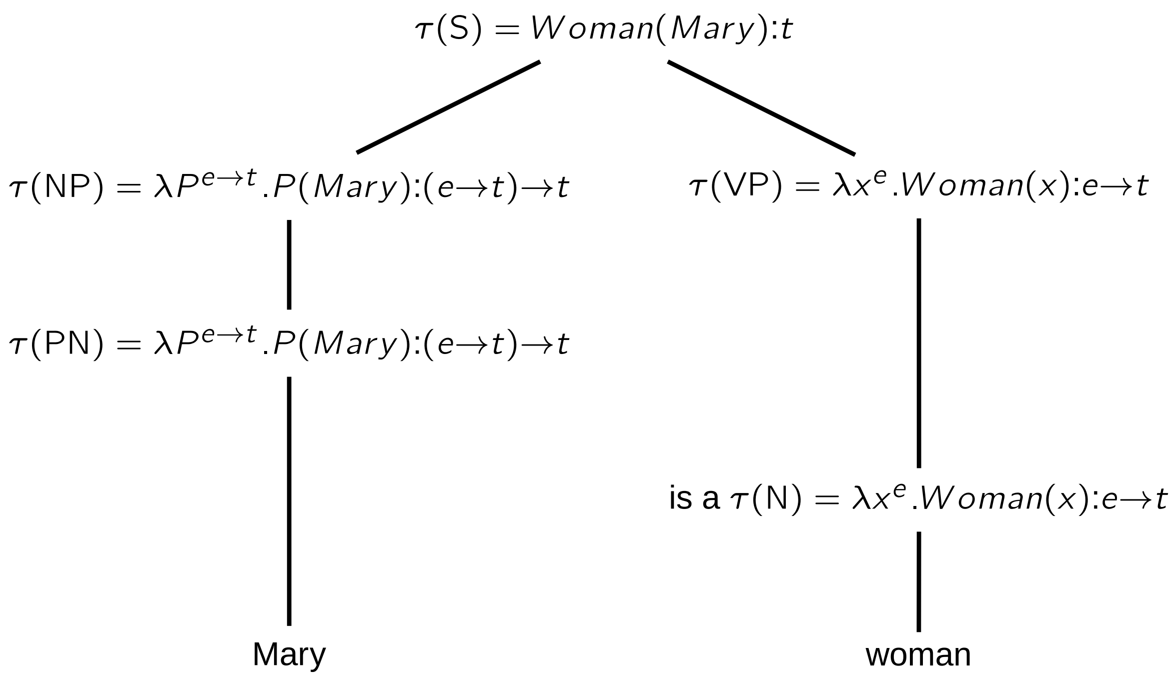
Parsing and interpreting "Mary is a woman"

Compositional Translations, Grammars and CLs



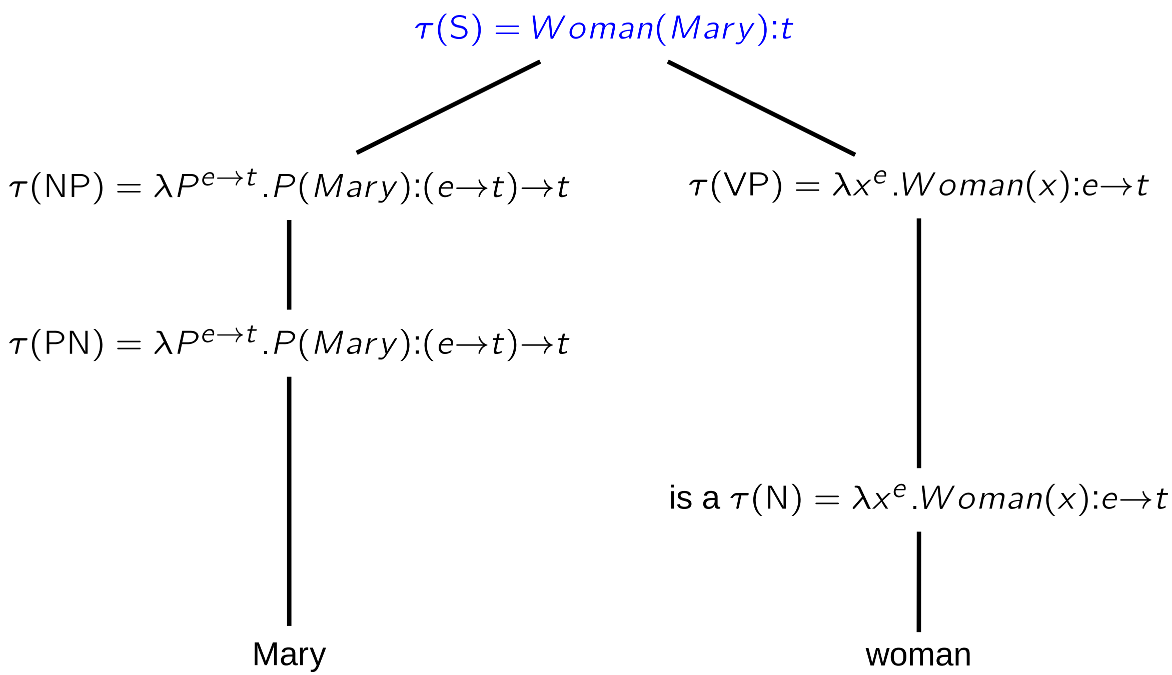
Parsing and interpreting "Mary is a woman"

Compositional Translations, Grammars and CLs



Parsing and interpreting "Mary is a woman"

Compositional Translations, Grammars and CLs



Generates a logic fragment!

Outline

1. Introduction
   (a) Controlled Languages
   (b) Ontology Data Access Systems and Scalability
   (c) Semantic Complexity
2. Lite-English and GCQ-English
3. Expressing Aggregations
4. The $\{\text{IS-A}_i\}_{i \in [0,7]}$ fragments
5. Complexity of the Fragments of English
6. Conclusions

Lite-English and GCQ-English

- GCQ-English capture tree-shaped CQs (TCQs)
- Lite-English sentences capture *DL-Lite*$_\sqcap$ assertions

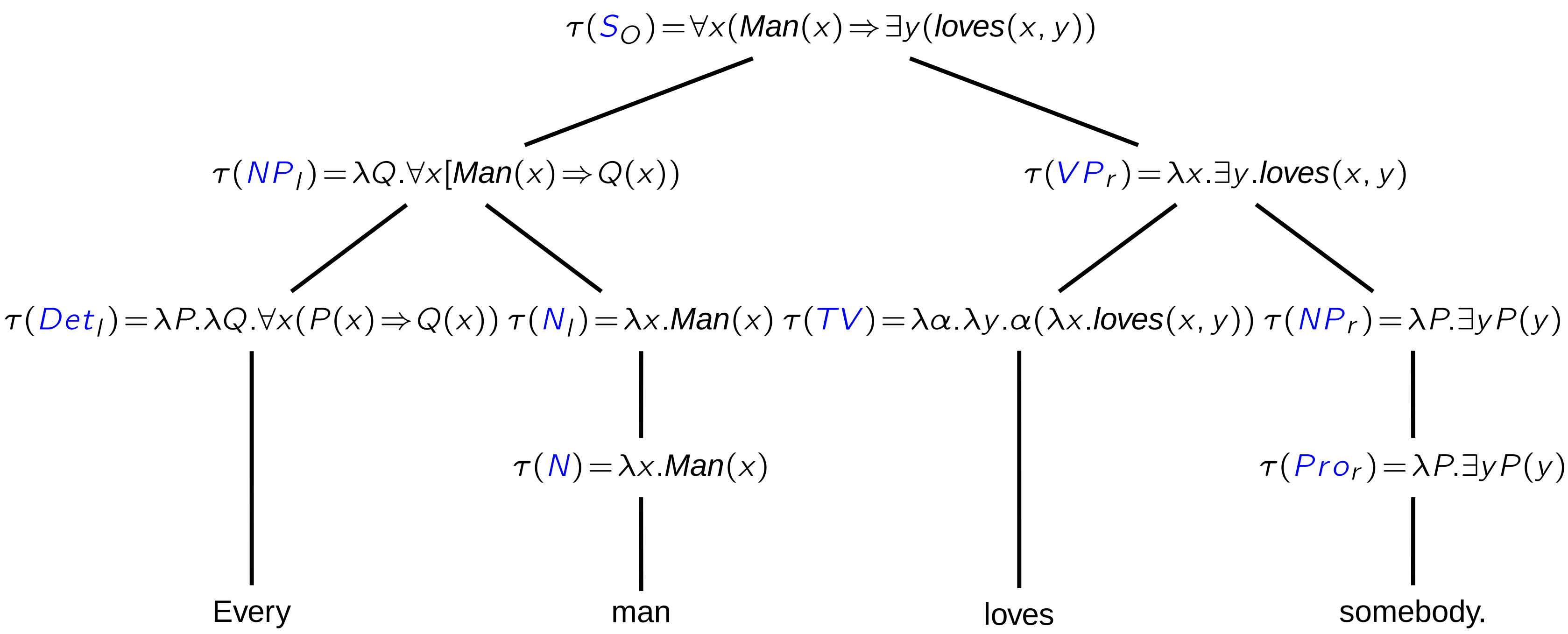
$$\underbrace{\forall x}_{Det_l} \quad (\underbrace{C_l(x)}_{N_l} \quad \Rightarrow \quad \underbrace{C_r(x)}_{VP_r})$$

- We subcategorize syntactic constituents using features!
- In *DL-Lite*$_\sqcap$ concept to the left and to the right of $\sqsubseteq$ have a separate syntax

$$C_l \longrightarrow A \mid \exists R \mid C_l \sqcap C_l' \qquad C_r \longrightarrow \neg A \mid \neg\exists R \mid C_l$$

- Parsing succeedds if features unify
- OBDAS query evaluation is in L w.r.t. data complexity

Lite-English and GCQ-English



Capturing *DL-Lite*$_\sqcap$ assertion *Man* $\sqsubseteq \exists$*loves*

Lite-English and The fragments of English [Pratt&Third 2005]

- The Fragments of English are built by combining the primitives:

  copula (COP), transitive verbs (TV), ditransitive verbs (DTV), relatives (Rel), restricted (RA) and unrestricted (GA) anaphora

- We showed that COP can be simulated by Lite-English
- We showed that Lite-English is closed under the unions of chains, but not COP+TV
- The abolute expressive power of Lite-English cannot be characterized

Outline

1. Introduction
   (a) Controlled Languages
   (b) Ontology Data Access Systems and Scalability
   (c) Semantic Complexity
2. Lite-English and GCQ-English
3. Expressing Aggregations
4. The $\{\text{IS-A}_i\}_{i \in [0,7]}$ fragments
5. Complexity of the Fragments of English
6. Conclusions

Aggregations

- Corpora analysis shows that in English determiners like "the number of" occur frequently in questions about numerical data
- We want to model their semantics with SQL aggregate functions: `SUM`, `MIN` `MAX`
- These functions are applied to bags called groups

Aggregations

- Corpora analysis shows that in English determiners like "the number of" occur frequently in questions about numerical data
- We want to model their semantics with SQL aggregate functions: `SUM`, `MIN` `MAX`
- These functions are applied to bags called groups

- An aggregate tree-shaped query (ATCQ) is a query of the form

$$\varphi := \{\underbrace{(x, \alpha(y))}_{\text{head}} \mid \underbrace{\psi_1(x) \land r(x, y) \land \psi_2(y)}_{\text{body } \psi \text{ is "tree-shaped"}}\}$$

whose core $\tilde{\varphi}$ is the quantifier-free matrix of ψ: a CQ!

Aggregates and Certain Answers

- Computing $\alpha(\cdot)$ is easy over a single model or database, but not immediate over OBDAS

- Problem: an OBDAS $(\mathcal{O}, \mathcal{D})$ has many models

- Different models contain different data $\Rightarrow \alpha(\cdot)$ will not yield a unique value!

Aggregates and Certain Answers

- Computing $\alpha(\cdot)$ is easy over a single model or database, but not immediate over OBDAS

- Problem: an OBDAS $(\mathcal{O}, \mathcal{D})$ has many models

- Different models contain different data $\Rightarrow \alpha(\cdot)$ will not yield a unique value!

- Solution: given an ATCQ $\varphi := \{(x, \alpha(y)) \mid \psi\}$

  1. return the certain answers of its core CQ $\tilde{\varphi}$ over $(\mathcal{O}, \mathcal{D})$
  2. define aggregates (SQL groups) over the certain answers
  3. return the value of $\alpha(\cdot)$

  i.e., return their aggregate certain answers

ATCQ-English and Bag Semantics

- The CL ATCQ-English expresses ATCQs
- Bags are functions from a set S into $\mathbb{N}$
- We need bag-typed expressions of type $e \rightarrow \mathbb{N}$ to capture groups:

$$\lambda x^{e}.\mathit{Woman}(x){:}e \rightarrow \mathbb{N}$$

- This generalizes the standard typing for English formal semantics: set-typed expressions are special cases of bag-typed expressions

ATCQ-English and Bag Semantics

- The CL ATCQ-English expresses ATCQs
- Bags are functions from a set S into $\mathbb{N}$
- We need bag-typed expressions of type $e{\rightarrow}\mathbb{N}$ to capture groups:

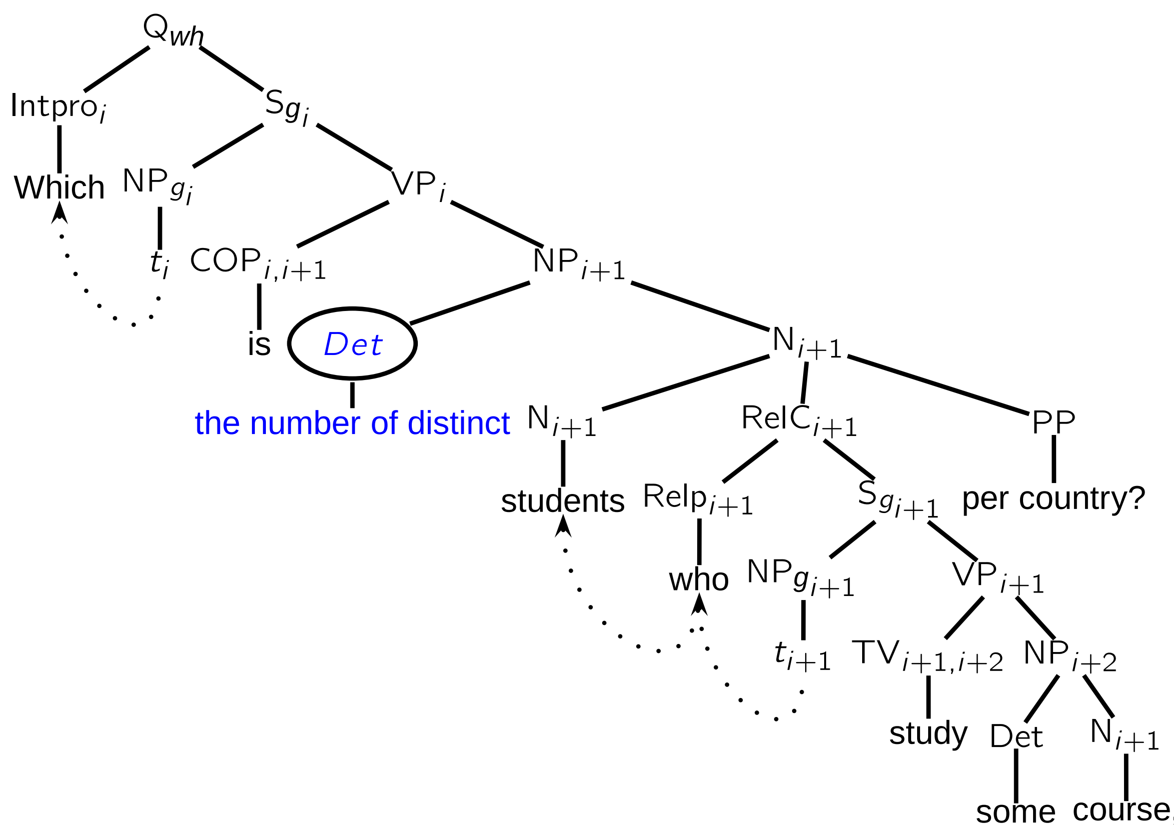
$$\lambda x^e.\mathit{Woman}(x){:}e{\rightarrow}\mathbb{N}$$

- This generalizes the standard typing for English formal semantics: set-typed expressions are special cases of bag-typed expressions
- Given $\{e, t, \mathbb{N}, \mathbb{Q}\}$ as basic types we define aggregate determiners like

$$\tau(\text{the distinct number of}) := \lambda P^{e\rightarrow\mathbb{N}}.\textsc{cntd}(P) : (e{\rightarrow}\mathbb{N}){\rightarrow}\mathbb{Q}$$
$$\tau(\text{the number of}) := \lambda P^{e\rightarrow\mathbb{N}}.\textsc{count}(P) : (e{\rightarrow}\mathbb{N}){\rightarrow}\mathbb{Q}$$

ATCQ-English and Bag Semantics

$$\lambda z^e.\lambda m^{\mathbb{Q}}.m \approx \text{CNTD}(\lambda x^e.\mathit{Student}(x) \wedge \exists y^e(\mathit{attends}(x,y)) \wedge \mathit{Course}(y) \wedge \mathit{comesFrom}(x,z) \wedge \mathit{Country}(z)) : e \to (\mathbb{Q} \to t).$$



ATCQs and Beyond

- Computing aggregates is not more costly than answering of its body
- Nor do body disjunctions $\psi(x) \vee \psi'(x)$ increase data complexity

ATCQs and Beyond

- Computing aggregates is not more costly than answering of its body
- Nor do body disjunctions $\psi(x) \vee \psi'(x)$ increase data complexity

- Complexity jumps to coNP-hard with
  - negations of bodies, $\neg\psi(x)$
  - universally restricted bodies, $\forall x(R(x,y) \rightarrow \psi(y))$
  - comparisons in bodies, $t\,\theta\,t'$, $\theta \in \{=,>,<,\geq,\leq\}$

- Idea: Reduction from 2+2-SAT (satisfiability of 2+2 propositional formulas)

ATCQs and Beyond: Data Complexity

	$\vee$-(A)TCQs	$\leq$-(A)TCQs	$\forall$-(A)TCQs	$\neg$-(A)TCQs
Lite-English	in L	coNP-hard	coNP-complete	coNP-complete
COP	in L	coNP-hard	coNP-complete	coNP-complete
COP+TV	in PTime	coNP-complete	coNP-complete	coNP-complete

N.B. (A)TCQ = (aggregate) tree-shaped CQ

ATCQs and Beyond: Data Complexity

	$\vee$-(A)TCQs	$\leq$-(A)TCQs	$\forall$-(A)TCQs	$\neg$-(A)TCQs
Lite-English	in L	coNP-hard	coNP-complete	coNP-complete
COP	in L	coNP-hard	coNP-complete	coNP-complete
COP+TV	in PTime	coNP-complete	coNP-complete	coNP-complete

N.B. (A)TCQ = (aggregate) tree-shaped CQ

- What happens when we go beyond Lite-English?

Outline

1. Introduction
   (a) Controlled Languages
   (b) Ontology Data Access Systems and Scalability
   (c) Semantic Complexity
2. Lite-English and GCQ-English
3. Expressing Aggregations
4. The $\{\text{IS-A}_i\}_{i \in [0,7]}$ fragments
5. Complexity of the Fragments of English
6. Conclusions

The IS-A$_{i \in [0,7]}$ CLs

- They are subsumed by

  DL-English $\Rightarrow$ DL $\mathcal{ALCI}$

- Data complexity w.r.t. to tree-shaped CQs:

IS-A_0	L	EL-English	PTIME-complete
IS-A_1	NL-complete	IS-A_5	coNP-complete
IS-A_2	PTIME-complete	IS-A_6	coNP-complete
IS-A_3	PTIME-complete	IS-A_7	coNP-complete
IS-A_4	PTIME-complete	DL-English	coNP-complete

- coNP-hard CLs are Boolean closed

IS-A_7	$\neg A \sqsubseteq A_1 \sqcap \cdots \sqcap A_n$	coNP-complete	Anybody who is not an area manager is an employee who eis a project manager

Outline

1. Introduction
   (a) Controlled Languages
   (b) Ontology Data Access Systems and Scalability
   (c) Semantic Complexity
2. Lite-English and GCQ-English
3. Expressing Aggregations
4. The $\{\text{IS-A}_i\}_{i \in [0,7]}$ fragments
5. Complexity of the Fragments of English
6. Conclusions

Complexity of the Fragments of English

- We have also studied the data complexity of Pratt & Third's fragments of English w.r.t.
  - ODBASs consistency
  - query evaluation
- We used resolution to study data complexity
- Resolution can be turned into a decision procedure by adopting refinements [Joyner 1976]
- They can be used to derive NP and coNP data complexity upper bounds
- For PTime bounds we use other techniques (reduction to DATALOG, separation property)

Complexity of the Fragments of English

	Data	
	CQs	Consistency
COP	in L	in L
COP+TV	in PTIME	in L
COP+TV+DTV	N.A.	in L

⇒ non-Boolean closed!

Data complexity in L ⇒ resolution + separation

Data complexity in PTIME ⇒ reduction to DATALOG

Complexity of the Fragments of English

	Data	
	(G)TCQs	Consistency
COP+Rel	coNP-complete [Pratt 2009]	in NP
COP+Rel+TV	coNP-complete [Pratt 2009]	NP-complete
COP+Rel+DTV	coNP-complete	NP-complete
COP+Rel+DTV+TV	coNP-complete	NP-complete

N.B. (G)TCQs = (generalized) tree-shaped CQs

$\Rightarrow$ Boolean closed!

Data complexity in coNP or NP $\Rightarrow$ resolution

Data complexity in coNP or NP $\Rightarrow$ 2+2 SAT

Outline

1. Introduction
   (a) Controlled Languages
   (b) Ontology Data Access Systems and Scalability
   (c) Semantic Complexity
2. Lite-English and GCQ-English
3. Expressing Aggregations
4. The $\{\text{IS-A}_i\}_{i \in [0,7]}$ fragments
5. Complexity of the fragments of English
6. Conclusions

Conclusions

- We have proposed to study the scalability to data of CLs by
  - proposing CLs: Lite-English, the IS-A_is, DL-English, ATCQ-English, GCQ-English
- We have proposed a compositional analysis of aggregations and aggregate questions in English, relating them to SQL aggregate queries
- We have pinpointed English constructs/function words whose combination
  - gives rise to tractable data complexity
  - gives rise to intractable data complexity
  - gives rise to undecidability
- We have also studied the data complexity of known CLs (viz. the fragments of English)

List of Peer-Reviewed Publications

- Publications 2009:
  - Camilo Thorne, Diego Calvanese, "Tree Shaped Aggregate Questions over Ontologies", Proceedings of the FQAS 2009 conference
  - Camilo Thorne, Diego Calvanese, "Controlled English Ontology-Based Data Access", Pre-Proceedings of the CNL 2009 workshop
- Publications 2008:
  - Diego Calvanese, Evgeny Kharlamov, Werner Nutt, Camilo Thorne "Aggregate Queries over Ontologies", Proceedings of the ONISW 2008 workshop
  - Camilo Thorne, "Expressing Conjunctive and Aggregate Queries over Ontologies with Controlled English", Proceedings of the ESSLLI 2008 summer school student workshop
- Publications 2007:
  - Raffaella Bernardi, Francesca Bonin, Diego Calvanese, Domenico Carbotta, Camilo Thorne, "English Querying over Ontologies: E-QuOnto", Proceedings of the AI*IA 2007 conference,
  - Camilo Thorne, "Managing Structured Data with Controlled English - An Approach based on Description Logics", Proceedings of the ESSLLI 2007 summer school student workshop
  - Camilo Thorne, Raffaella Bernardi, Diego Calvanese "Expressing DL-Lite Ontologies with Controlled English", Proceedings of the DL 2007 workshop,
  - Raffaella Bernardi, Diego Calvanese, Camilo Thorne, "Lite Natural Language" Proceedings of the IWCS-7 workshop

Presentations

- Presentations 2009:
  - "The Data Complexity of the Syllogistic Fragments", AC 2009, Amsterdam, Holland
  - "Tree Shaped Aggregate Questions over Ontologies", FQAS 2009, Roskilde, Denmark
  - "Exploring Controlled English Ontology-Based Data Access", CNL 2009, Marettimo, Italy
  - "Aggregate Queries over Ontologies", KRDB lunch talk, Bolzano, Italy
- Presentations 2008:
  - "Aggregate Queries over Ontologies", ONISW 2008, Napa, US
  - "Expressing Conjunctive and Aggregate Queries over Ontologies with Controlled English" (Poster presentation), ESSLLI 2008, Hamburg, Germany
  - "Expressing Formal Queries over DL-Lite Ontologies with Controlled English", Bolzano LCT student colloquium, Bolzano, Italy
  - "Expressing *DL-Lite* QA with Controlled English", Bolzano CS Faculty winter seminar, Dobbiaco, Italy
- Presentations 2007:
  - "Expressing DL-Lite Ontologies with Controlled English", DL 2007, Bressanone, Italy
  - "Expressing DL-Lite Ontologies with Controlled English", KRDB PhD students workshop, Bolzano, Italy

Annex: CL Complexity

	Declarations	Questions
Constructs that scale (PTIME or less)	– Negation in predicate VPs, relatives in predicate VPs, conjunction in predicate VPs – Relatives and conjunction in subject NPs and predicate VPs, but no negation	– Existential quantifiers, conjunction, relatives, aggregations, disjunctions
Constructs that do not (CONP-hard)	– Negation in subject NPs – Relatives and negation in subject NPs and predicate VPs	– Full negation – Comparisons – Universal restrictions
Undecidable Constructs	– TVs, relatives, negation, existential and universal quantifiers, restricted anaphoric pronouns and indeterminate pronouns in subject NPs and predicate VPs, copula	– TVs, existential indeterminate pronouns, relatives and restricted anaphoric pronouns

Annex: Aggregate Certain Answers

Given ATCQ

$$\varphi := \{(x, \alpha(y)) \mid \psi\}$$

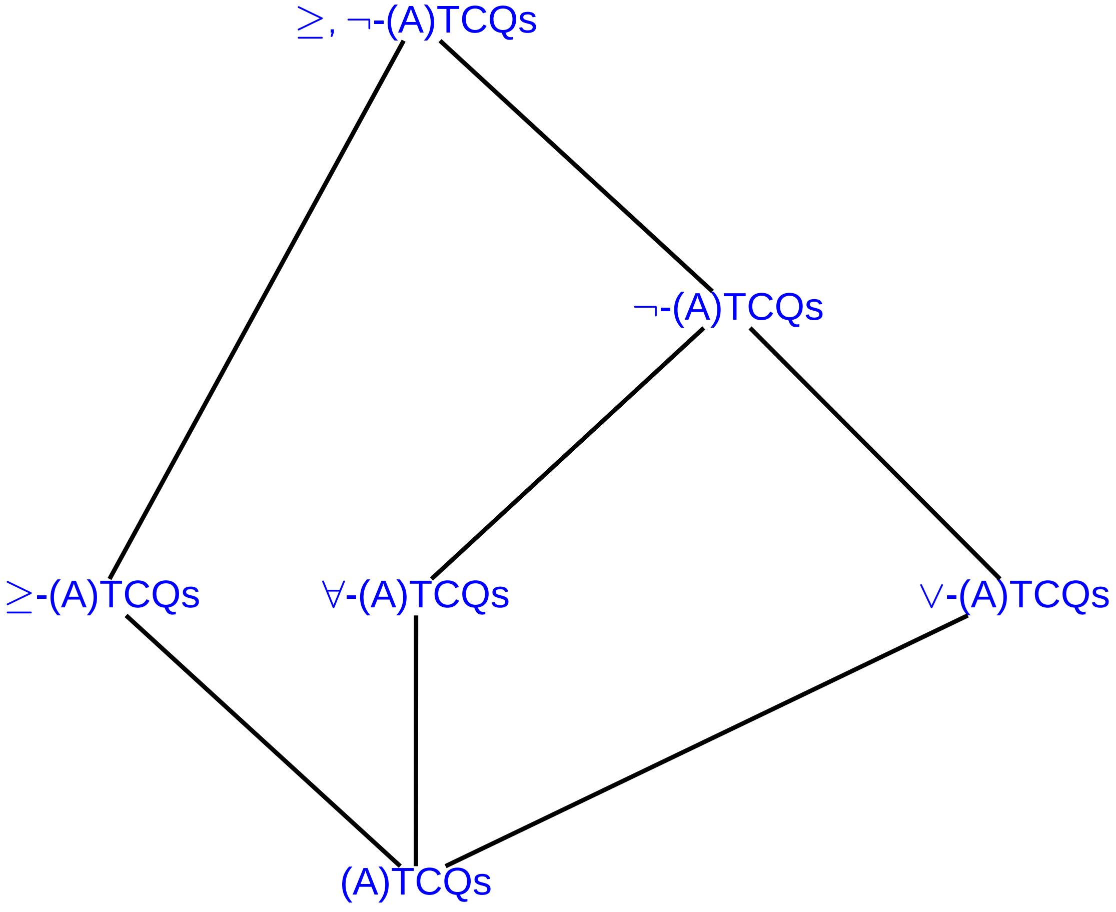
– the certain group of tuple c is the bag

$$H_c := \{\!|\sigma(y) \mid \sigma(x) = c, (\mathcal{O}, \mathcal{D}) \models \tilde{\varphi}\sigma|\!\}$$

– the set of aggregate certain answers of φ of aggregation variable x and term $\alpha(y)$ is

$$\textit{cert}_a(\varphi, \mathcal{O}, \mathcal{D}) := \{(\sigma(x), \alpha(H_{\sigma(x)})) \mid (\mathcal{O}, \mathcal{D}) \models \tilde{\varphi}\sigma\}$$

Annex: ATCQs and Beyond



Annex: The IS-A$_{i\in[0,7]}$ CLs

$$\overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{every}}\ \overbrace{C_l}^{Nom_r}\ \overbrace{C_r}^{\text{VP}_l} \qquad \overbrace{\lambda C_l.\lambda C_r.C_l \sqsubseteq C_r}^{\text{everybody who}}\ \overbrace{C_l}^{\text{VP}_l}\ \overbrace{C_r}^{\text{VP}_r}$$

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