Complete Query Answering over Horn Ontologies Using a Triple Store

Yujiao Zhou

joint work with Yavor Nenov, Bernardo Cuenca Grau and Ian Horrocks

Department of Computer Science, University of Oxford

Conjunctive Query Answering

- An *ontology* $\mathcal{O}$ consists of a set of schema-level axioms.
- A *dataset* $\mathcal{D}$ consists of a set of facts (ground atoms).
- A *conjunctive query* is of the form $Q(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$.
- $\vec{a} \in \mathrm{cert}(Q, \mathcal{O}, \mathcal{D})$ iff $\mathcal{O} \cup \mathcal{D} \models Q(\vec{a})$.

Running example

$$\mathcal{O}^{ex} = \{\ \textit{SubClassOf}(\text{Animal } \textit{SomeValuesFrom}(\text{eats Thing})),$$
$$\textit{SubClassOf}(\text{Herbivore } \textit{AllValuesFrom}(\text{eats Plant})),$$
$$\textit{DisjointClasses}(\text{Herbivore Carnivore}),$$
$$\textit{SubClassOf}(\text{Carnivore } \textit{MinCardinality}(2 \text{ hasParent}))\ \}$$

$$\mathcal{D}^{ex} = \{\ \textit{ClassAssertion}(\text{Animal}, \textit{lion}),\ \textit{ClassAssertion}(\text{Animal}, \textit{rabbit}),$$
$$\textit{ClassAssertion}(\text{Herbivore}, \textit{rabbit}),\ \textit{ClassAssertion}(\text{Herbivore}, \textit{sheep}),$$
$$\textit{PropertyAssertion}(\text{eats}, \textit{sheep}, \textit{grass}),\ \textit{ClassAssertion}(\text{Carnivore}, \textit{wolf})\ \}$$

$$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$$

$$\text{cert}(Q^{ex}, \mathcal{O}^{ex}, \mathcal{D}^{ex}) = \{\ \textit{sheep}, \textit{rabbit}\ \}$$

$$\mathcal{O}^{ex} = \{\ \textit{SubClassOf}(\text{Animal } \textit{SomeValuesFrom}(\text{eats Thing})),$$
$$\textit{SubClassOf}(\text{Herbivore } \textit{AllValuesFrom}(\text{eats Plant})),$$
$$\textit{DisjointClasses}(\text{Herbivore Carnivore}),$$
$$\textit{SubClassOf}(\text{Carnivore } \textit{MinCardinality}(2 \text{ hasParent}))\ \}$$

$$\mathcal{D}^{ex} = \{\ \textit{ClassAssertion}(\text{Animal}, \textit{lion}),\ \textit{ClassAssertion}(\text{Animal}, \textit{rabbit}),$$
$$\textit{ClassAssertion}(\text{Herbivore}, \textit{rabbit}),\ \textit{ClassAssertion}(\text{Herbivore}, \textit{sheep}),$$
$$\textit{PropertyAssertion}(\text{eats}, \textit{sheep}, \textit{grass}),\ \textit{ClassAssertion}(\text{Carnivore}, \textit{wolf})\ \}$$

$$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$$

$$\mathcal{O}^{ex} \cup \mathcal{D}^{ex} \models Q^{ex}(\textit{sheep})$$
$$\text{cert}(Q^{ex}, \mathcal{O}^{ex}, \mathcal{D}^{ex}) = \{\ \textit{sheep}, \textit{rabbit}\ \}$$

$$\mathcal{O}^{ex} = \{\ \mathit{SubClassOf}(\text{Animal } \mathit{SomeValuesFrom}(\text{eats Thing})),$$
$$\mathit{SubClassOf}(\text{Herbivore } \mathit{AllValuesFrom}(\text{eats Plant})),$$
$$\mathit{DisjointClasses}(\text{Herbivore Carnivore}),$$
$$\mathit{SubClassOf}(\text{Carnivore } \mathit{MinCardinality}(2 \text{ hasParent}))\ \}$$

$$\mathcal{D}^{ex} = \{\ \mathit{ClassAssertion}(\text{Animal}, \mathit{lion}),\ \mathit{ClassAssertion}(\text{Animal}, \mathit{rabbit}),$$
$$\mathit{ClassAssertion}(\text{Herbivore}, \mathit{rabbit}),\ \mathit{ClassAssertion}(\text{Herbivore}, \mathit{sheep}),$$
$$\mathit{PropertyAssertion}(\text{eats}, \mathit{sheep}, \mathit{grass}),\ \mathit{ClassAssertion}(\text{Carnivore}, \mathit{wolf})\ \}$$

$$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$$

$$\mathcal{O}^{ex} \cup \mathcal{D}^{ex} \models Q^{ex}(\mathit{rabbit})$$
$$\text{cert}(Q^{ex}, \mathcal{O}^{ex}, \mathcal{D}^{ex}) = \{\ \mathit{sheep}, \mathit{rabbit}\ \}$$

Background

- OWL 2
  - High computational complexity
  - Limited scalability of OWL 2 reasoner
- Lightweight profiles, e.g. OWL 2 RL
  - Polynomial data complexity

    Scalable triple stores: OWLim, Oracle's RDF Semantic Graph, RDFox ...
  - Restrictions on expressivity

    Disjunctive and existential axioms

Approach Overview

OWL 2 RL reasoner - most workload

1. Compute lower and upper bounds for query answers;[1]
2. *Extract relavant part to resolve answers in the gap;

OWL 2 reasoner - as little work as possible

3. Check the answers in the gap with the *relevant fragment*.

[1] Yujiao Zhou et al. "Making the Most of your Triple Store: Query Answering in OWL 2 Using an RL Reasoner". In: *WWW*. 2013.

Lower Bounds

$\mathcal{O}_L$: RL fragment of $\mathcal{O}$

$$\mathcal{O}_L \subseteq \mathcal{O}$$

Property:

$$\text{cert}(Q, \mathcal{O}_L, \mathcal{D}) \subseteq \text{cert}(Q, \mathcal{O}, \mathcal{D})$$

Example

$$\mathcal{O}_L^{ex} = \{\ \textit{SubClassOf}(\text{Herbivore } \textit{AllValuesFrom}(\text{eats Plant}))$$
$$\textit{DisjointClasses}(\text{Herbivore Carnivore})\ \}$$

$$\mathcal{D}^{ex} = \{\ \textit{ClassAssertion}(\text{Animal}, \textit{lion}),\ \textit{ClassAssertion}(\text{Animal}, \textit{rabbit}),$$
$$\textit{ClassAssertion}(\text{Herbivore}, \textit{rabbit}),\ \textit{ClassAssertion}(\text{Herbivore}, \textit{sheep}),$$
$$\textit{PropertyAssertion}(\text{eats}, \textit{sheep}, \textit{grass}),\ \textit{ClassAssertion}(\text{Carnivore}, \textit{wolf})\ \}$$

$$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$$

$$\text{cert}(Q^{ex}, \mathcal{O}_L^{ex}, \mathcal{D}^{ex}) = \{\ \textit{sheep}\ \}$$

Upper Bounds

Over-approximate $\mathcal{O}$ to an RL ontology $\mathcal{O}_U$, s.t.

$$\mathcal{O}_U \models \mathcal{O}$$

Property

$$\mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) \subseteq \mathsf{cert}(Q, \mathcal{O}_U, \mathcal{D})$$

- Disjunctive axioms: disjunctions $\rightarrow$ conjunctions.

  SubClassOf(Mammal *UnionOf*(Herbivore Carnivore))

  $\rightsquigarrow$ *SubClassOf*(Mammal *IntersectionOf*(Herbivore Carnivore))

- Existential axioms: skolemisation by fresh individuals.

  SubClassOf(Animal *SomeValuesFrom*(eats Thing))

  $\rightsquigarrow$ *SubClassOf*(Animal *HasValue*(eats *c*))

$$\mathcal{O}^{ex} = \{\ \textit{SubClassOf}(\text{Animal } \textit{SomeValuesFrom}(\text{eats Thing})),$$
$$\textit{SubClassOf}(\text{Herbivore } \textit{AllValuesFrom}(\text{eats Plant})),$$
$$\textit{DisjointClasses}(\text{Herbivore Carnivore}),$$
$$\textit{SubClassOf}(\text{Carnivore } \textit{MinCardinality}(2 \text{ hasParent}))\ \}$$

$$\mathcal{D}^{ex} = \{\ \textit{ClassAssertion}(\text{Animal}, \textit{lion}),\ \ \textit{ClassAssertion}(\text{Animal}, \textit{rabbit}),$$
$$\textit{ClassAssertion}(\text{Herbivore}, \textit{rabbit}),\ \ \textit{ClassAssertion}(\text{Herbivore}, \textit{sheep}),$$
$$\textit{PropertyAssertion}(\text{eats}, \textit{sheep}, \textit{grass}),\ \ \textit{ClassAssertion}(\text{Carnivore}, \textit{wolf})\ \}$$

$$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$$

$$\mathcal{O}_U^{ex} = \{ \ \textit{SubClassOf}(\text{Animal } \textit{HasValue}(\text{eats } c)),$$
$$\textit{SubClassOf}(\text{Herbivore } \textit{AllValuesFrom}(\text{eats Plant})),$$
$$\textit{DisjointClasses}(\text{Herbivore Carnivore}),$$
$$\textit{SubClassOf}(\text{Carnivore } \textit{HasValue}(\text{hasParent } c_1)),$$
$$\textit{SubClassOf}(\text{Carnivore } \textit{HasValue}(\text{hasParent } c_2)),$$
$$\textit{DifferentFrom}(c_1 \ c_2) \ \}$$

$$\mathcal{D}^{ex} = \{ \ \textit{ClassAssertion}(\text{Animal}, \textit{lion}), \ \textit{ClassAssertion}(\text{Animal}, \textit{rabbit}),$$
$$\textit{ClassAssertion}(\text{Herbivore}, \textit{rabbit}), \ \textit{ClassAssertion}(\text{Herbivore}, \textit{sheep}),$$
$$\textit{PropertyAssertion}(\text{eats}, \textit{sheep}, \textit{grass}), \ \textit{ClassAssertion}(\text{Carnivore}, \textit{wolf}) \ \}$$

$$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$$

$$\text{cert}(Q^{ex}, \Sigma_U^{ex}, \mathcal{D}^{ex}) = \{ \ \textit{sheep}, \textit{rabbit}, \textit{lion} \ \}$$

Lower & Upper Bounds

$$\begin{cases} \vec{a} \in \mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) & \vec{a} \in \mathrm{cert}(Q, \mathcal{O}_L, \mathcal{D}) \\ \vec{a} \notin \mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) & \vec{a} \notin \mathrm{cert}(Q, \mathcal{O}_U, \mathcal{D}) \\ & \end{cases}$$

sheep
wolf

Lower & Upper Bounds

$$\begin{cases} \vec{a} \in \mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) & \vec{a} \in \mathrm{cert}(Q, \mathcal{O}_L, \mathcal{D}) & \text{sheep} \\ \vec{a} \notin \mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) & \vec{a} \notin \mathrm{cert}(Q, \mathcal{O}_U, \mathcal{D}) & \text{wolf} \\ ??? & \vec{a} \in \mathrm{cert}(Q, \mathcal{O}_U, \mathcal{D}) \setminus \mathrm{cert}(Q, \mathcal{O}_L, \mathcal{D}) & \text{rabbit, lion} \end{cases}$$

Lower & Upper Bounds

$$\begin{cases} \vec{a} \in \text{cert}(Q, \mathcal{O}, \mathcal{D}) & \vec{a} \in \text{cert}(Q, \mathcal{O}_L, \mathcal{D}) & \text{sheep} \\ \vec{a} \notin \text{cert}(Q, \mathcal{O}, \mathcal{D}) & \vec{a} \notin \text{cert}(Q, \mathcal{O}_U, \mathcal{D}) & \text{wolf} \\ ??? & \vec{a} \in \text{cert}(Q, \mathcal{O}_U, \mathcal{D}) \setminus \text{cert}(Q, \mathcal{O}_L, \mathcal{D}) & \text{rabbit, lion} \end{cases}$$

- Computed by the same OWL 2 RL reasoner efficiently.

Main Intuition

Two questions:

- how $Q^{ex}(rabbit), Q^{ex}(lion)$ are obtained from $\mathcal{O}_U^{ex} \cup \mathcal{D}^{ex}$;
- if they can be derived from $\mathcal{O}^{ex} \cup \mathcal{D}^{ex}$ in an analogical way.

A proof of $Q^{ex}(\mathit{rabbit})$ in $\mathcal{O}_U^{ex} \cup \mathcal{D}$

ClassAssertion(Animal *rabbit*)	
SubClassOf(Animal *HasValue*(*eats c*))	eats(*rabbit*, *c*)
ClassAssertion(Herbivore *rabbit*)	
SubClassOf(Herbivore *AllValuesFrom*(eats Plant))	Plant(*c*)

$\{$corresponding axioms in $\mathcal{O} \cup \mathcal{D}\} \models Q^{ex}(\mathit{rabbit})$

ClassAssertion(Animal *rabbit*)
SubClassOf(Animal *SomeValuesFrom*(*eats* Thing))
ClassAssertion(Herbivore *rabbit*)
SubClassOf(Herbivore *AllValuesFrom*(eats Plant))

A proof of $Q^{ex}(lion)$ in $\mathcal{O}_U^{ex} \cup \mathcal{D}$

ClassAssertion(Animal *rabbit*)	
SubClassOf(Animal *HasValue*(*eats c*))	eats(*rabbit*, *c*)
ClassAssertion(Herbivore *rabbit*)	
SubClassOf(Herbivore *AllValuesFrom*(eats Plant))	Plant(*c*)
ClassAssertion(Animal *lion*)	
SubClassOf(Animal *HasValue*(*eats c*))	eats(*lion*, *c*)

$\{$corresponding axioms in $\mathcal{O} \cup \mathcal{D}\} \not\models Q^{ex}(lion)$

ClassAssertion(Animal *rabbit*)
SubClassOf(Animal *SomeValuesFrom*(*eats* Thing))
ClassAssertion(Herbivore *rabbit*)
SubClassOf(Herbivore *AllValuesFrom*(eats Plant))
ClassAssertion(Animal *lion*)

Main Intuition

Given a tuple $\vec{a}$ in the gap,

- find *ALL* the proofs of $Q(\vec{a})$ in $\mathcal{O}_U \cup \mathcal{D}$;
- "trace back" the proofs into fragments $\mathcal{O}_f \subseteq \mathcal{O}$ and $\mathcal{D}_f \subseteq \mathcal{D}$.

$$\vec{a} \in \mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) \text{ iff } \vec{a} \in \mathrm{cert}(Q, \mathcal{O}_f, \mathcal{D}_f) \text{ for each } \vec{a} \text{ in the gap}$$

$G^{ex} = \{ \; rabbit, \; lion \; \}$

$\mathcal{O}^{ex} = \{ \; \mathit{SubClassOf}(\text{Animal } \mathit{SomeValuesFrom}(\text{eats Thing})),$

$\mathit{SubClassOf}(\text{Herbivore } \mathit{AllValuesFrom}(\text{eats Plant})),$

$\mathit{DisjointClasses}(\text{Herbivore Carnivore}),$

$\mathit{SubClassOf}(\text{Carnivore } \mathit{MinCardinality}(2 \text{ hasParent})) \; \}$

$\mathcal{D}^{ex} = \{ \; \mathit{ClassAssertion}(\text{Animal } \mathit{lion}), \; \mathit{ClassAssertion}(\text{Animal } \mathit{rabbit}),$

$\mathit{ClassAssertion}(\text{Herbivore } \mathit{rabbit}), \; \mathit{ClassAssertion}(\text{Herbivore } \mathit{sheep}),$

$\mathit{PropertyAssertion}(\text{eats } \mathit{sheep} \; \mathit{grass}), \; \mathit{ClassAssertion}(\text{Carnivore } \mathit{wolf}) \; \}$

$Q^{ex}(x) = \exists y(\text{eats}(x, y) \wedge \text{Plant}(y))$

$$\mathcal{O}_f \cup \mathcal{D}_f \models Q^{ex}(\mathit{rabbit}) \text{ but } \mathcal{O}_f \cup \mathcal{D}_f \not\models Q^{ex}(\mathit{lion})$$

OWL 2 Reasoner

To verify if the extracted fragments validate $Q(\vec{a})$ for each $\vec{a}$ in the gap:

$$\mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) = \mathrm{cert}(Q, \mathcal{O}_L, \mathcal{D}) \cup \{\vec{a} \text{ in the gap} \mid \mathcal{O}_f \cup \mathcal{D}_f \models Q(\vec{a})\}.$$

Example

$$\mathcal{O}_f \cup \mathcal{D}_f \models Q^{ex}(\mathit{rabbit}) \text{ but } \mathcal{O}_f \cup \mathcal{D}_f \not\models Q^{ex}(\mathit{lion})$$

$$\mathrm{cert}(Q^{ex}, \mathcal{O}^{ex}, \mathcal{D}^{ex}) = \mathrm{cert}(Q^{ex}, \mathcal{O}^{ex}_L, \mathcal{D}^{ex}) \cup \{\mathit{rabbit}\} = \{\mathit{sheep}, \mathit{rabbit}\}$$

Datasets

- Lehigh University Benchmark (LUBM)
  Generated queries
- Fly Anatomy (FLY)
  A realistic and complex ontology describing the anatomy of flies.

Data	DL	Horn	Existential	Classes	Properties	Axioms	Individuals	Dataset
LUBM(n)	$\mathcal{SHI}$	Yes	8	43	32	93	$1.7 \times 10^4 n$	$10^5 n$
FLY	$\mathcal{SRI}$	Yes	8,396	7,533	24	144,407	1,606	6,308

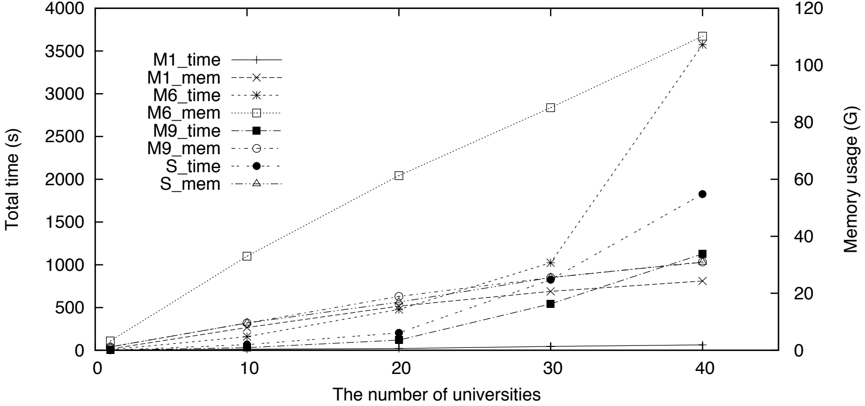
Results

on LUBM40 with $|\mathcal{O}| = 93$ and $|\mathcal{D}| = 4,000,000$

Query	$\lvert G \rvert$	$\lvert \mathcal{O}_f \rvert$	$\lvert \mathcal{D}_f \rvert$	time(s)
M_1	39	6	29,041	60.7
M_2	1	6	29,004	42.1
M_3	16	6	29,054	47.6
M_4	30	6	29,032	60.2
M_5	4	6	29,010	64.3
M_6	29	10	87,209	3,339.4
M_7	15	6	29,033	60.7
M_8	14	6	29,038	52.2
M_9	10	12	86,785	886.7
S	39	12	29,041	2,126.5

Scalability

on LUBM1 – LUBM40



Results of the other queries are very similar to that of M_1.

Results

on FLY with $|\mathcal{O}| = 144,407$ and $|\mathcal{D}| = 6,308$

Query	$\lvert G\rvert$	$\lvert\mathcal{O}_f\rvert$	$\lvert\mathcal{D}_f\rvert$	time(s)	HermiT(s)
Q_1	803	224	4,515	155.2	3,465.9
Q_2	342	224	4,054	114.0	3,179.0
Q_3	28	217	3,712	92.3	5,863.3
Q_4	25	233	3,762	99.2	2,944.3
Q_5	518	222	3,712	124.6	3,243.7

Conclusion

$$\langle Q, \mathcal{O}, \mathcal{D} \rangle \overset{G}{\rightsquigarrow} \langle \mathcal{O}_f, \mathcal{D}_f \rangle$$

$$\vec{a} \in \mathrm{cert}(Q, \mathcal{O}, \mathcal{D}) \text{ iff } \vec{a} \in \mathrm{cert}(Q, \mathcal{O}_f, \mathcal{D}_f) \text{ for each } \vec{a} \in G$$

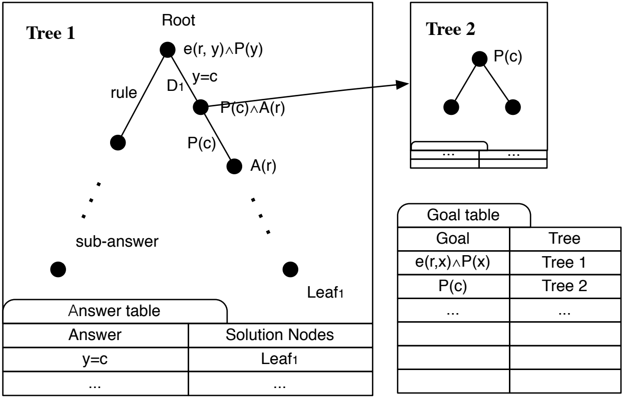
Future Work

- To extend the work to OWL 2 ontologies
- To improve the scalability by datalog encoding
- Other optimisations, e.g. summarisation, improvement of lower & upper bounds.

Backward Chaining Algorithm

Tabling technique

Objective: to ensure the termination of finding out *all* proofs;



Answer table

Answer	Solution Nodes
y=c	Leaf1
...	...

Goal table

Goal	Tree
e(r,x)∧P(x)	Tree 1
P(c)	Tree 2
...	...

Backward Chaining Algorithm

Pruning with lower and upper bounds

Before backward chaining on a node labeled with a goal P, we check P both in the lower and upper bounds.

- Upper bounds: $\text{cert}(P, \mathcal{O}_U, \mathcal{D}) = \emptyset$;
- Lower bounds: P is ground and $\text{cert}(P, \mathcal{O}_L, \mathcal{D}) \neq \emptyset$.

Results

on LUBM40 with $|\mathcal{O}| = 93$ and $|\mathcal{D}| = 4,000,000$

Query	$\|V\|$	n	$\|G\|$	t_f	$\|\mathcal{O}_f\|$	$\|\mathcal{D}_f\|$	t_{check}	t_{total}
M_1	2	3	39	36.4	6	29,041	H: 23.3	H: 60.7
M_2	3	4	1	37.1	6	29,004	H: 4.0	H: 42.1
M_3	4	6	16	38.2	6	29,054	H: 8.4	H: 47.6
M_4	2	3	30	36.0	6	29,032	H: 23.3	H: 60.2
M_5	3	4	4	39.4	6	29,010	H: 24.0	H: 64.3
M_6	4	6	29	2,845.8	10	87,209	H: 483.0	H: 3,339.4
M_7	3	5	15	38.0	6	29,033	H: 10.3	H: 49.3
M_8	3	5	14	39.3	6	29,038	H: 11.9	H: 52.2
M_9	3	4	10	328.9	12	86,785	H: 556.2	H: 886.7
S	1	2	39	310.0	12	86,802	H: 1,780.0 P: 16,592.1	H: 2,126.5 P: 16,870.0

Results

on FLY with $|\mathcal{O}| = 144,407$ and $|\mathcal{D}| = 6,308$

Query	$\lvert V \rvert$	n	$\lvert G \rvert$	t_f	$\lvert \mathcal{O}_f \rvert$	$\lvert \mathcal{D}_f \rvert$	t_{check} (H)	t_{total}	t_{HermiT}
Q_1	2	3	803	108.9	224	4515	45.9	155.2	3,465.9
Q_2	3	5	342	97.7	224	4054	16.0	114.0	3,179.0
Q_3	1	1	28	91.0	217	3712	0.9	92.3	5,863.3
Q_4	2	3	25	94.3	233	3762	4.7	99.2	2,944.3
Q_5	2	2	518	100.3	222	3712	24.0	124.6	3,243.7