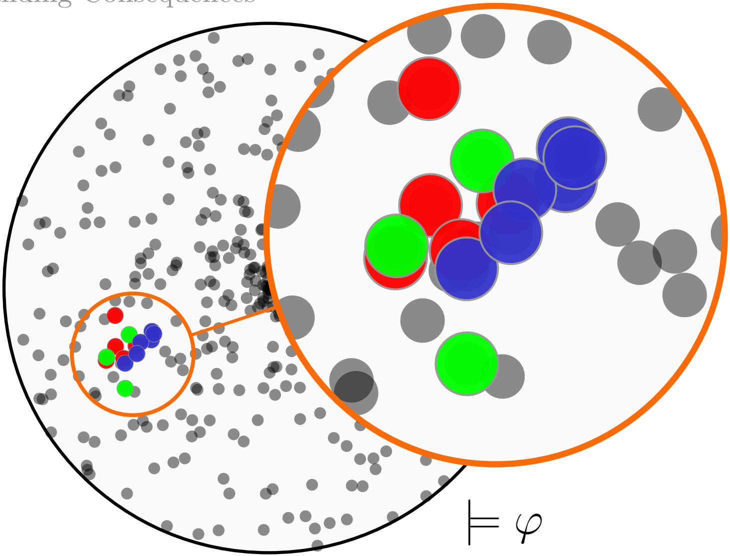
unibz
Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

KRDB

Lean Kernels in Description Logics

R. Peñaloza, C. Mencía, A. Ignatiev, J. Marques-Silva

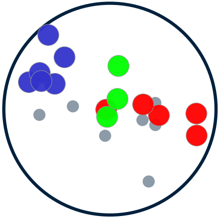
Portoroz, May 30, 2017



$\models \varphi$

Find sub-ontology $\mathcal{K}$ such that:

- all (minimal) causes for φ are in $\mathcal{K}$ (MinA preserving)
- is as small as possible (minimal irrelevance)
- efficiently computable



In propositional logic:

- ontology: set of clauses
- φ: unsatisfiability (empty clause)
- MinA: MUS

lean kernels approximate union of MUSes

Resolution combines two clauses to produce a new one

(entailed)

$$\left.\begin{array}{c} x_1 \vee x_2 \vee \neg x_3 \\ \neg x_2 \vee x_4 \vee \neg x_5 \end{array}\right\} \rightsquigarrow x_1 \vee \neg x_3 \vee x_4 \vee \neg x_5$$

Lean kernel of a (derived) clause c:

set of all clauses appearing in some derivation proof for c

(in particular, for empty clause)

In propositional logic, LKs

- are fast to compute
- (over-)approximate the union of MUSes well

Generalize this idea to description logics?

Consequence-based methods make relevant consequences explicit

Rules: $(\mathcal{B}_0, \mathcal{S}) \to \mathcal{B}_1$

take explicit knowledge ($\mathcal{B}_0$) and axioms ($\mathcal{S}$)

produce more explicit knowledge ($\mathcal{B}_1$)

$\mathcal{ALC}$ Concepts:

$$C, D ::= A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.C \mid \forall r.C$$

Ontology (in normal form): finite set of GCIs

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$$
$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n$$
$$\exists r.A \sqsubseteq B$$
$$A \sqsubseteq \exists r.B$$
$$A \sqsubseteq \forall r.B$$

Consequence-based Rules for $\mathcal{ALC}$

$\mathcal{B}_0$	$\mathcal{S}$	$\mathcal{B}_1$
$(H \sqcap \neg A, N \sqcup A)$	$\emptyset$	(H, N)
$(H, N_1 \sqcup A_1), \ldots, (H, N_n \sqcup A_n)$	$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$	$(H, \bigsqcup_{i=1}^{n} N_i \sqcup B)$
$(H, N \sqcup A)$	$A \sqsubseteq \exists r.B$	(H, N, r, B)
$(H, M, r, K), (K, N \sqcup A)$	$\exists r.A \sqsubseteq B$	$(H, M \sqcup B, r, K \sqcap \neg A)$
$(H, M, r, K), (K, \bot)$	$\emptyset$	(H, M)
$(H, M, r, K), (H, N \sqcup A)$	$A \sqsubseteq \forall r.B$	$(H, M \sqcup N, r, K \sqcap B)$

H, K conjunctions of literals $\quad M, N$ disjunctions of literals

$$(H, M) \quad \rightsquigarrow \quad H \sqsubseteq_{\mathcal{T}} M$$

$$(H, N, r, K) \quad \rightsquigarrow \quad H \sqsubseteq_{\mathcal{T}} N \sqcup \exists r.K$$

Given

consequence-based algorithm C

ontology $\mathcal{T}$

consequence φ

Lean kernel of φ w.r.t. $\mathcal{T}$, C:

set of all axioms of $\mathcal{T}$ appearing in some C-proof of φ

Example

$\mathcal{B}_0$	$\mathcal{S}$	$\mathcal{B}_1$
$(H, N_1 \sqcup A_1), \ldots, (H, N_n \sqcup A_n)$	$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$	$(H, \bigsqcup_{i=1}^{n} N_i \sqcup B)$
$(H, N \sqcup A)$	$A \sqsubseteq \exists r.B$	(H, N, r, B)
$(H, M, r, K), (K, N \sqcup A)$	$\exists r.A \sqsubseteq B$	$(H, M \sqcup B, r, K \sqcap \neg A)$
$(H, M, r, K), (K, \bot)$	$\emptyset$	(H, M)

$ax_1 = A \sqsubseteq B$ $\quad ax_2 = A \sqsubseteq \exists r.A$ $\quad ax_3 = \exists r.B \sqsubseteq B$

$ax_4 = B \sqsubseteq C$ $\quad ax_5 = A \sqsubseteq \exists s.C$ $\quad ax_6 = \exists s.A \sqsubseteq B$

R_1	(A, A)	$A \sqsubseteq B$	(A, B)
R_2	(A, A)	$A \sqsubseteq \exists r.A$	$(A, \emptyset, r, A)$
R_3	$(A, \emptyset, r, A), (A, B)$	$\exists r.B \sqsubseteq B$	$(A, B, r, A \sqcap \neg B)$
R_4	$(A, B, r, A \sqcap \neg B), (A \sqcap \neg B, \bot)$	$\emptyset$	(A, B)
R_5	(A, B)	$B \sqsubseteq C$	(A, C)

CB algorithms can be extended to compute LKs:

- attach derived consequences with set of axioms used
- apply rules until saturation
- update set of repeated consequences

Properties:

- produce LKs for all relevant consequences
- only a linear overhead
- MinA preserving

Example

1	(A, A)	$A \sqsubseteq B$	(A, B)	$\{1\}$
2	(A, A)	$A \sqsubseteq \exists r.A$	$(A, \emptyset, r, A)$	$\{2\}$
3	$(A, \emptyset, r, A), (A, B)$	$\exists r.B \sqsubseteq B$	$(A, B, r, A \sqcap \neg B)$	$\{1, 2, 3\}$
4	$(A, B, r, A \sqcap \neg B), (A \sqcap \neg B, \bot)$	$\emptyset$	(A, B)	$\{1, 2, 3\}$
5	(A, B)	$B \sqsubseteq C$	(A, C)	$\{1, \ldots, 4\}$
6	(A, A)	$A \sqsubseteq \exists s.C$	$(A, \emptyset, s, C)$	$\{5\}$

Computed the LKs for all atomic subsumptions from five well-known $\mathcal{EL}^+$ ontologies:

$\mathcal{T}$	# axioms	$\lvert\text{class}(\mathcal{T})\rvert$
GENE	20466	164743
NCI	46800	252519
NOT-GALEN	4379	27980
FULL-GALEN	36544	453674
SNOMED-CT	307704	5333580

Total: more than 6M LKs

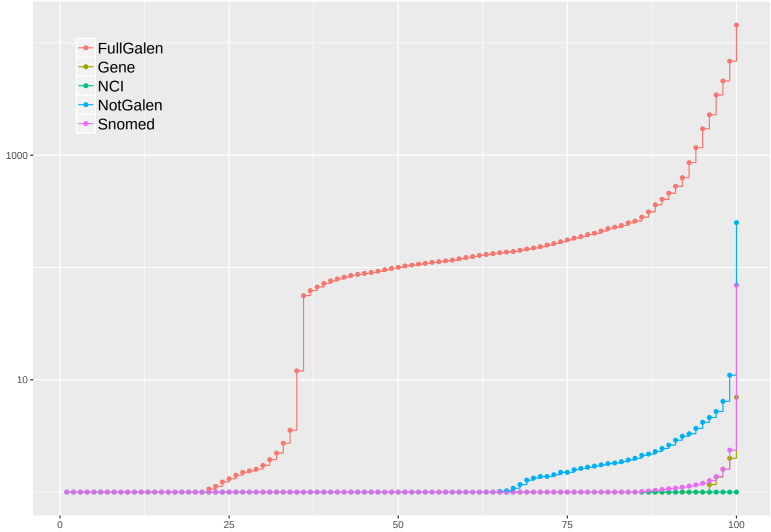
Compare to locality-based modules (MinA preserving)

Formally:

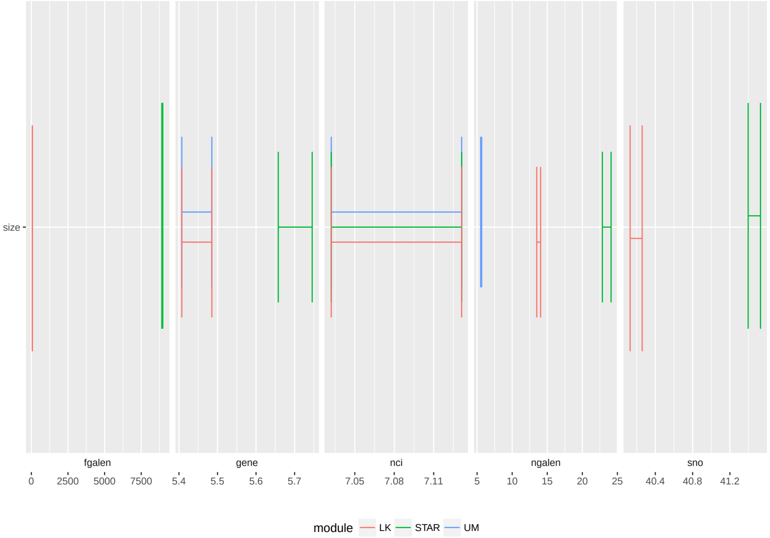
$$\text{UMinAs} \subseteq \text{LK} \subseteq \bot\top^* \subseteq \bot$$

In practice?

Proportional Improvement vs. Star-module



99.9% Confidence Intervals Do Not Overlap



Efficiently Computable?

$\mathcal{T}$	LKs	$\bot$			$\bot\top^*$		
		Min	Avg	Max	Min	Avg	Max
GENE	1.98	0.00	0.01	0.04	0.00	0.01	0.06
NCI	3.79	0.00	0.01	0.10	0.01	0.02	0.13
NGALEN	4.22	0.00	0.01	0.02	0.00	0.01	0.03
FGALEN	461.03	0.00	0.36	1.12	0.02	2.46	8.06
SNOMED	11200.53	0.11	0.65	5.34	0.26	3.91	28.92

For 5.3 M consequences!

Conclusions

Lean Kernels approximate union of MinAs

can be effectively computed for all consequences
(linear overhead)

Want a more goal directed approach?
compute LK from $\bot$-module

Questions?